Appears in *Proceedings of the Eleventh World Conference on Medical Informatics* (Medinfo 2004), San Francisco, CA, September 7-11, 2004.

Remote Display Performance for Wireless Healthcare Computing

Albert Max Lai[a], Jason Nieh[b], Andrew Laine[c,d], Justin Starren[a,d]

[a] *Department of Biomedical Informatics, Columbia University, NY, USA*
[b] *Department of Computer Science, Columbia University, NY, USA*
[c] *Department of Biomedical Engineering, Columbia University, NY, USA*
[d] *Department of Radiology, Columbia University, NY, USA*

Abstract

Organizations are beginning to recognize that health care providers are highly mobile and optimal care requires providing access to a large and dynamic body of information wherever the provider and patient are. Remote display protocols (RDP) are one way that organizations are using to deliver healthcare applications to mobile users. While many organizations have begun to use RDPs to deliver real-time access to health care information to clinicians, little formal work has been done to evaluate the performance or the effectiveness of thin-client computing with health care applications. This study examines the performance of wireless thin-client tablets with two web-based clinical applications, a text-centric, graphics-poor EMR and a graphic-rich image analysis program. The study compares the performance of two popular RDP implementations, Citrix and Microsoft Remote Desktop, with the performance of a traditional web browser in a wireless environment. For both applications, the RDPs demonstrated both higher speed and reduced bandwidth requirements than the web browser.

Keywords:

Computer Communication Networks, Internet, Medical Records Systems, Teleradiology, Software, Remote Display Protocols, Tablet, Wireless

Introduction

Organizations are beginning to recognize that health care providers are highly mobile and optimal care requires providing access to a large and dynamic body of information wherever the provider and patient are. Thin-client computing is one way to support delivery of information in this complex environment. There are a variety of types of thin-client systems. Among these are traditional client-server applications, web applications, and those running over remote display protocols (RDP). The thin-client approach is an effective way to deliver access to this information due to the real-time access nature of the approach.

Figures 1 and 2 contrast the traditional web client model with the remote display protocol (RDP) model. A RDP allows for graphical displays to be virtualized across a network to a client device while the application logic is executed on the server. Using this display protocol, the client transmits the user input to the server, and the server returns screen updates of the user interface of the applications from the server to the client. The remote server typically runs a standard server operating system and is used for executing all application logic. Because all application processing is done on the server, the client only needs to display and manipulate the user interface. In addition, because the server runs a standard operating system, most applications users are familiar with on the desktop can be used. This allows for a much more interactive suite of applications than can be used with the web paradigm. The client can be either a specialized hardware device or simply an application that runs on a low-end personal computer. Examples of popular thin-client platforms using remote display protocols include Citrix MetaFrame [4, 16], Microsoft Terminal Services[1, 7], AT&T Virtual Network Computing (VNC) [19], and Tarantella [3, 20].

Figure 1: Traditional web client model

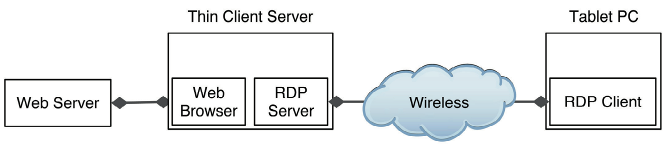

Figure 2: RDP client model

The major benefits of these platforms are the reduced cost of maintaining systems due to the maintenance and management centralization. This allows for centrally pre-configured applications as well as the selection of supported applications. This provides for increased uniformity, reducing the difficulty of supporting applications. This allows organizations to maintain smaller numbers of computers, ensuring a higher probability of proper security maintenance. This is one of the reasons why some hospitals have begun to use these types of servers as part of their HIPAA security compliance solution. In addition, this approach allows for a wider variety of client devices. Because the application processing is done on the server, thin-client computing provides additional benefits for mobile devices. Mobile devices are often plagued with under-powered processors and short battery life. With the application logic pushed to an application server, where there battery life is not an issue, the processor utilization can be decreased to allow for improved performance and longer battery life[9].

While many organizations have begun to use thin clients to deliver real-time access to health care information to clinicians, little formal work has been done to evaluate the performance or the effectiveness of thin-client computing with health care applications. This study examines the performance of wireless thin-client tablets with two web based clinical applications.

Materials and Methods

Because thin-client systems are designed and used very differently from traditional desktop systems, quantifying and measuring their performance effectively can be difficult. In traditional desktop systems, an application typically executes and displays its output on the same machine. In thin-client systems, an application executes on a server machine and sends its output over a network to be displayed on a client machine via a remote display protocol. The output display on the client may be completely decoupled from the application processing on the server such that an application runs as fast as possible on the server without regard to whether or not application output has been displayed on the client. Furthermore, display updates may be merged or even discarded in some systems to conserve network bandwidth. Since the server processes all application logic in thin-client systems, standard application benchmarks effectively measure only server performance and do not accurately reflect user perceived performance at the client. The problem is exacerbated by the fact that many thin-client systems, including those from Citrix and Microsoft are proprietary and closed-source, making it difficult to instrument them to obtain accurate, repeatable performance results.

To address these problems, we employed slow-motion benchmarking to evaluate thin client performance. This method employs two techniques to obtain accurate measurements: monitoring client-side network activity and using slow-motion versions of application benchmarks. We give a brief overview of this technique below. For a more in depth discussion, see [18].

We monitored network activity to obtain a measure of user-perceived performance based on latency. Since we could not directly peer into the black-box thin-client systems, our primary measurement technique was to use a packet monitor to capture resulting network traffic on the client-side. For example, to measure the latency of an operation from user input to client output, we could use the packet monitor to determine when the user input is first sent from client to server and when the screen update finished sending from server to client. The difference between these times could be used as a measure of latency. To accurately measure user-perceived thin-client performance, measurements must be performed at the client-side; server-side measurements of application performance are insufficient. For instance, a video application might deliver smooth playback on the server-side only to deliver poor video quality on the client-side due to network congestion.

We employed slow-motion versions of application benchmarks to provide a measure of user-perceived performance based on the visual quality of display updates. While monitoring network activity provides a measure of the latency of display updates, it does not provide a sufficient measure of the overall quality of the performance.

To address this problem, we created web benchmark applications with delays between the pages of the benchmark, so that the display update for each web page is fully completed on the client before the server begins processing the next display update. We monitored network traffic to make sure the delays were long enough to provide a clearly demarcated period between display updates where client-server communication drops to the idle level for that platform. We then process the results on a per-page basis to obtain the latency and data transferred for each web page, and obtain overall results by taking the sum of these results.

Table 1: Testbed machine configurations

Role/Model	Hardware	OS	Software
Thin Server IBM Netfinity 4500R	Dual 933MHz Intel PIII 512 MB RAM 9GB disk 10/100Base-T NIC	Microsoft Windows 2000 Pro.	Microsoft Windows Terminal Services Citrix Meta-Frame XP Release 1 Mozilla 1.4
Web Server (as above)	(as above)	Debian Linux	Apache 1.3.27
Packet Monitor (as above)	(as above)	Debian Linux	Ethereal 0.9.13
Thin Client ViewSonic ViewPad 1000	800 MHz Mobile Celeron	Microsoft Windows 2000 Pro.	Citrix ICA Win32 Client MS RDP5 Client Mozilla 1.4

Our combined measurement techniques provide two key benefits. First, the techniques ensure that display events reliably complete on the client so that packet captures from network monitoring provide an accurate measure of system performance. Ensuring that all clients display all visual components in the same sequence provides a common foundation for making comparisons among thin-client systems. Second, the techniques do not require any invasive modification of thin-client systems. As a result, we are able obtain our results without imposing any additional performance overhead on the systems measured. More importantly, the techniques make it possible for us to measure popular but proprietary thin-client systems, such as those from Citrix and Microsoft.

Experimental Testbed

The testbed consists of a thin client server, a packet monitor, a web server used for testing web applications, and an 802.11b wireless access point. The features of each system are summarized in Table 1. The packet monitor machine was dedicated to running Ethereal [2], a software packet monitor that time-stamps and records all packet traffic visible by the machine. The thin client server, packet monitor, and web server are IBM Netfinity 4500R's, each with dual 933 MHz Pentium III CPUs, 512 MB RAM, and 7200 RPM Ultra 160 9GB SCSI disks. The tablet computer is a ViewSonic ViewPad 1000, with an 800 MHz Mobile Celeron CPU 256 MB of RAM. All of the IBM Netfinity machines are connected via Intel EtherExpress Pro 100 10/100BaseT NICs. The wireless access point we used is a Proxim ORiNOCO AP-2000 Wireless Access Point, equipped with a single 11 Mbps PC card. The thin client server is running Windows 2000 Server. The packet monitor and web server are running Debian Linux with a 2.4 series kernel. The tablet computer was placed within a few feet of the access point during the experiments.

To minimize application environment differences, we used common configuration options and common applications across all thin client platforms whenever possible. Where it was not possible to configure all the platforms in the same way, we generally used default settings for the platforms in question. The video resolution of the client was set to the maximum resolution supported by the tablet, 800x600. Compression and memory caching, and disk caching were left in their default settings.

We tested the applications with the traditional web browser (Mozilla), Citrix MetaFrame XP Feature Release 1, and Micro-soft Windows Terminal Services. In this paper, we refer to Citrix and Terminal Services by their remote display protocols, which are Citrix ICA (Independent Computing Architecture) and Microsoft RDP (Remote Desktop Protocol). We selected these platforms because they are the two most widely used remote display protocols in the healthcare industry. We performed the tests with settings up to the 16 bits per pixel (bpp) maximum allowed by the tablet device: ICA in 8 bpp, ICA in 16 bpp, and RDP in 8 bpp. Unfortunately, Windows 2000 Terminal Services does not support bit-depths higher than 8 bpp. ICA was configured to use 128-bit encryption of the display to reflect how it would likely be configured for use in a clinical setting.

Application Benchmarks

To determine the performance of thin clients, we used two web benchmarks: a clinical information system (CIS) and a mammogram enhancement application. The CIS benchmark application is a sequence of 18 web pages from New York Presbyterian Hospital's Web-based clinical information system (WebCIS) [6, 11]. WebCIS displays data from a variety of clinical data sources including results from laboratory, radiology, and cardiology departments. These pages contain primarily textual data in free text and in tabular form. Navigation and forms submission is JavaScript driven. The pages selected for the benchmark were based upon common page sequences determined through CIS log file analysis. CIS log analysis is a technique based on data mining and Web usage mining used to discover patterns of CIS usage. From one year's worth of WebCIS logs, one of the techniques used in this method, sequential pattern discovery was applied to identify archetypal representative sequences of data access in WebCIS. We used these typical sequences in the CIS benchmark in order to reflect the usage patterns of actual clinical users of WebCIS. For a more in depth discussion of CIS log analysis, see [5].

The mammogram enhancement web application is a sequence of 20 primarily graphical web pages that demonstrates real time wavelet enhancement of mammograms. In this application, the image enhancement calculations are performed by a common gateway interface (CGI) on the server. The CGI computes the image enhancement and displays the results in a web page. These images are representative of the kinds of image processing activities we anticipate will become commonplace in the clinical practice of radiological diagnosis in the near future.

Each of these web applications were configured to be used as slow motion benchmark applications. As each page downloads, a small script contained in each page starts off the subsequent download after introducing delays of several seconds between pages, sufficient in each case to ensure that the thin client received and displayed each page completely and there was no temporal overlap in transferring the data belonging to two consecutive pages. We used the packet monitor to record the packet traffic for each page, and then used the timestamps of the first and last packet associated with each page to determine the download time for each page. The browser's cache was enabled but cleared before each test run.

Results

The figures refer to the thin-client platforms based on their remote display protocols. Figure 3 shows the per-page latency results of running the mammogram benchmark on the tablet computer. Although there is variation between pages, there is a general trend that the traditional web browser provides poorer performance than the thin clients. Figure 4 shows the average latency of a page in the respective benchmark. Latency is measured as the time difference between the first and last packet associated with a web page. As expected, as demonstrated in Figure 4, the textual (graphics poor) application performs much more quickly than the graphics intensive applications regardless of protocol. All three thin clients give better performance than using the traditional web browser approach for both the WebCIS and for the graphical mammogram applications.

Figure 5 shows the average data transferred per page displayed. In all cases, the data transferred by the thin clients was significantly less than that transferred by the traditional web browser approach.

Discussion

For both graphics rich and graphics poor clinical web pages, the performance of RDP clients in a wireless environment is faster than the performance of a native web browser. In addition, the data transferred when using thin clients in both applications is significantly less. Lastly, this study demonstrates the effectiveness of utilization log analysis in the creation of representative application benchmarks.

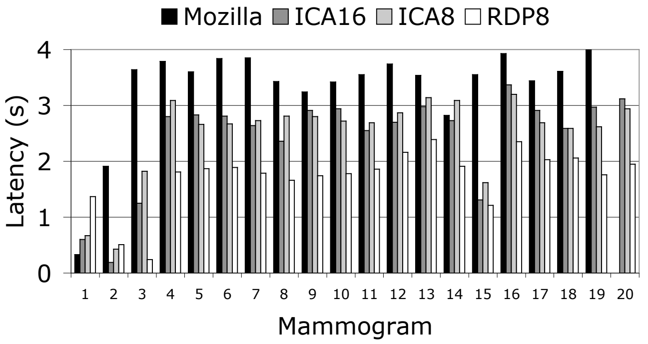

Figure 3: Per-page load latency comparing various RDPs with traditional web browser. Mammogram represents performance of graphics rich applications. Mozilla represents a web browser running on the wireless tablet. ICA16 and ICA8 represent eight and sixteen bit options for Citrix. RDP8 represents Microsoft Remote Desktop Protocol.

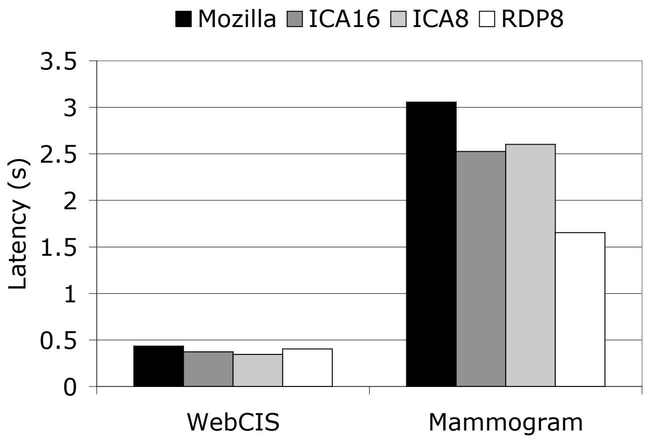

Figure 4: Average web page load latency comparing various RDPs with traditional web browser. WebCIS represents performance of graphics poor applications. Mammogram represents performance of graphics rich applications. Thin clients are the same as for Figure 3.

Several studies have examined the performance of RDPs in other application environments [8, 13, 18]. These studies have focused on measuring the performance of RDPs in different network bandwidths and latencies, but have not directly considered the impact of conditions found specifically in wireless networks such as 802.11b. These studies also have not investigated the performance of clinical applications over RDPs. One study investigated the performance of viewing 3D medical imaging datasets over a variety of network bandwidths and latencies [14].

In studies performed by Yang, Nieh, et al. [21], they investigated the performance of general web browsing using RDPs with wireless networks. They showed that under degraded network conditions, browsing the web through a thin client ensures more reliable web transactions and faster response time. Our results are consistent with their claim of reduced response times with RDPs in the wireless environment.

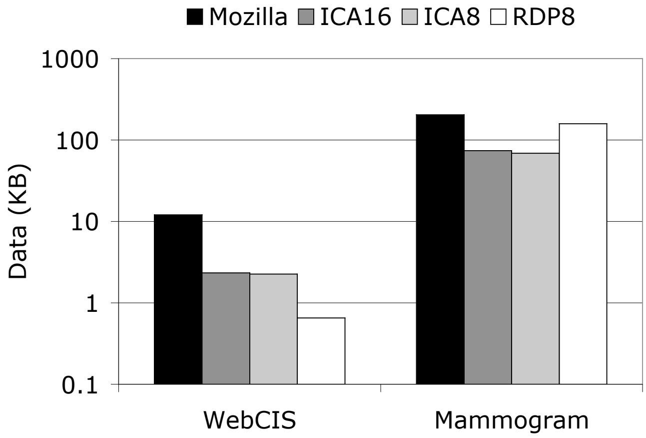

Figure 5: Average web page data transfer comparing various RDPs with traditional web browser presented in log scale. WebCIS represents performance of graphics poor applications. Mammogram represents performance of graphics rich applications. Thin clients are the same as for Figure 3.

However, their study used results from a simulated network. We extend this work through the use of real 802.11b hardware and have determined that accessing clinical applications with RDPs provide lower response times than a traditional web browser in a controlled wireless environment.

Our results by necessity have been collected in a controlled network environment as opposed to real clinical users in the field. However, because our page sequences have been selected using a year's worth of actual interaction with WebCIS, we believe our measurements constitute a valid predictor of performance to be expected in real world clinical use. Our data does not provide measures statistical significance due to the nature of slow-motion benchmarking. Because of the highly controlled environment of an isolated testbed network and deterministic behavior of computers, the measurement are highly reproducible run-to-run. Any significant run-to-run variability would indicate problems with the testbed and the entire data set would be discarded. The current study also does not investigate the network intermittency issues that may arise in a mobile environment or how wireless wide area network (WWAN) performance may be affected.

To our knowledge, this is the first study reporting quantitative RDP performance measures for healthcare applications, and the first to study report RDP performance in a real-world wireless environment. For the CIS benchmark, a text rich application, contrary to commonly held beliefs, the web browser required a higher data transfer than the RDPs transferring the encoded display in a graphical representation. Although initially surprising, this finding is quite plausible. Like many modern web applications, WebCIS makes extensive use of JavaScript and other code executed in the browser. This code may account for the surprisingly large amount of data transferred to the web browser. This may provide an explanation for why so much data is transferred to the web browser.

In many ways, the graphics-rich radiology benchmark can be viewed as a worst-case scenario for RDPs. The use of RDPs is traditionally recommended when there is a need to off-load large computations to a remote server because the client device does not have enough resources. Simple display multimedia has traditionally been considered an area of difficulty for RDPs. However, our data show that RDPs can perform better than the traditional web browser in both page load latency and data transfer in the wireless environment despite the fact that, for this application, all of the image processing occurred server-side for both the web and RDP clients. We expect that, in cases where the image processing occurs on the client device, the performance advantage of RDPs will only be improved due to the likelihood of greater computational resources to be available on the RDP server than on a mobile tablet device.

In spite of the paucity of published evaluations, many healthcare institutions have begun to roll out the use of RDPs, including the VA Hospital System [12, 15]. However few studies have been done to evaluate the scale of deployment or the effectiveness of thin-client computing in the healthcare environment. A number of CIS manufacturers have certified their products for use with RDPs [17]. Concentra Health Services is using Citrix to provide access to CIS applications to physicians using Windows CE devices because it provided better performance and provided more efficient use of bandwidth [10].

Conclusion

For both text-centric and graphics-centric web-based healthcare applications in a wireless environment, RDPs can not only deliver a better web browsing experience than a native web browser. In addition, RDPs can reduce data transfer requirements, even for graphics-poor applications.

Acknowledgments

This work has been supported by National Library of Medicine Training Grant NO1-LM07079. The authors thank Elizabeth Chen for the WebCIS sequential patterns data.

References

[1] Microsoft Windows NT Server 4.0, Terminal Server Edition: An Architectural Overview. 1998.

[2] *Ethereal* [computer program]. Version 0.9.13; 2003.

[3] Tarantella Web-Enabling Software: The Adaptive Internet Protocol. Dec. 1998.

[4] Citrix MetaFrame 1.8 Backgrounder. June 1998.

[5] Chen ES, Cimino JJ. Automated Discovery of Patient-Specific Clinician Information Needs Using Clinical Information System Log Files. *Proc AMIA Symp.* Nov. 8-12, 2003:(In Press).

[6] Cimino JJ, Socratous SA, Clayton PD. Internet as clinical information system: application development using the World Wide Web. *J Am Med Inform Assoc.* Sep-Oct 1995;2(5):273-284.

[7] Cumberland BC, Carius G, Muir A. Microsoft Windows NT Server 4.0, Terminal Server Edition: Technical Reference. Aug. 1999.

[8] Danskin JM, Hanrahan P. Profiling the X Protocol. *Proceedings of the SIGMETRICS Conference on Measurement and Modeling of Computer Systems*. Nashville, TN; May 1994.

[9] Gurun S, Krintz C. *Addressing the Energy Crisis in Mobile Computing With Developing Power Aware Software*: University of California, Santa Barbara; June 2003. 2003-15.

[10] Higgins K. Concentra Health Finds a Cure for Wireless Growth. Paper presented at: Network Computing, July22 2002.

[11] Hripcsak G, Cimino JJ, Sengupta S. WebCIS: large scale deployment of a Web-based clinical information system. *Proc AMIA Symp.* 1999:804-808.

[12] Jackson W. VA's thin clients cover a lot of territory. *Government Computer News.* July2 2001;20(17).

[13] Lai A, Nieh J. Limits of Wide-Area Thin-Client Computing. *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. Marina del Rey, CA, USA; June15-19 2002:228-239.

[14] Lai A, Nieh J, Laine A, Starren J. Thin Client Performance for Remote 3-D Image Display. *Proc AMIA Symp.* Nov. 8-12, 2003 2003:(In Press).

[15] Mangus M, Thomas S. Veteran's Health Administration Adopts Server-based Computing to Access Patient Information. Paper presented at: Power Solutions, 2001.

[16] Mathers TW, Genoway SP. Windows NT Thin Client Solutions: Implementing Terminal Server and Citrix MetaFrame. Nov. 1998.

[17] Meditech. *Thin Client Technology: The Citrix and MEDITECH Value* 2002.

[18] Nieh J, Yang SJ, Novik N. Measuring Thin-Client Performance Using Slow-Motion Benchmarking. *IEEE/ACM Trans. Computer Systems.* Feb. 2003;21(1):87-115.

[19] Richardson T, Stafford-Fraser Q, Wood KR, Hopper A. Virtual Network Computing. *IEEE Internet Computing.* Jan./Feb. 1998;2(1).

[20] Shaw A, Burgess KR, Pullan JM, Cartwright PC. Method of Displaying an Application on a Variety of Client Devices in a Client/Server Network. Aug. 2000.

[21] Yang SJ, Nieh J, Krishnappa S, Mohla A, Sajjadpour M. Web browsing performance of wireless thin-client computing. Paper presented at: Proceedings of the twelfth international conference on World Wide Web, 2003.

Address for correspondence

Albert Lai, albert.lai@dbmi.columbia.edu. Department of Biomedical Informatics, Vanderbilt Clinic, 5th Floor, 622 West 168th St., New York, NY 10032.