Marina Gavrilova et al. (Eds.)

LNCS 3981

Computational Science and Its Applications – ICCSA 2006

International Conference
Glasgow, UK, May 2006
Proceedings, Part II

2 Part II

Springer

Workshop on Mobile Communications (MC 2006)

Thin-Client Computing for Supporting the QoS of Streaming Media in Mobile Devices*

Joahyoung Lee, Dongmahn Seo, Yoon Kim,
Changyeol Choi, Hwangkyu Choi, and Inbum Jung

Department of Computer Science and Engineering, Kangwon National University,
Chuncheon, 200-701, Korea
{jinnie4u, sarum, yooni, cychoi, hkchoi, ibjung}@kangwon.ac.kr

Abstract. Due to the limited resources in thin-clients on mobile devices and the large amount of computation for decoding streaming media, it is not easy to support the QoS of streaming media for thin-client on mobile environment. To solve the problems, the terminal servers would be charged for decoding the streaming media and thin-clients have a role to update only the changed areas in their screen. In this paper, an intelligent media player is proposed to provide the improved QoS for streaming media in thin-client computing. Since the proposed method reflects the intrinsic property of streaming media, it provides both the enhanced video quality and the audio streaming fully synchronized with image frames.

1 Introduction

The thin-client computing has been used in computing areas until the desktop computing model has emerged in the mid 1980's. However, as the desktop computing has suffered from their own problems like separated administration, security, maintenance and unaffordable many large computational tasks, the thin-client computing has taken a growing interest again as a recent computing model [1, 2].

In the thin-client computing, a server and clients are communicates with each other on a remote display protocol. The protocol allows graphical displays to be virtual and served across a network to a client device. Using the remote display protocol, the client transmits user input to the server and the server returns screen updates of the user interface of the applications to the client. Clients can access the resources of servers and make their virtual working space with their keyboard and monitor. From this approach, the thin-client can not only sustain the centralized computing environment but also support the desktop working environment for clients. The server centralizes the system management jobs like

* This work was partially supported by the Kangwon Institute of Telecommunications and Information, grant No. B1220-0401-0203 from the University Fundamental Research Program of the Ministry of Information & Communication in Republic of Korea, and Basic Research Program of the Korea Science Engineering Foundation (KOSEF).

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3981, pp. 562–571, 2006.

the software modification and update so that the thin-client solution relieves the cost and the time exhausted by maintaining the installed software. Under the convergence of administration, the effective working environment is provided for general clients.

During the pass several years, most of information could be obtained from the Web pages at the level of text and graphic data. However, most recently, the computing fields are move to the multimedia applications, instead of the existing text-based and graphic applications. The necessary information is being provided from the various multimedia contents. In particular, the streaming media service has been spotlighted in many multimedia applications. To meet the contemporary requirement, the desktop computing model provides the clients with the working platform for handling the multimedia contents. However, the thin-client computing still has remained in the graphic based circumstance.

The thin-client system has supplied the large scale clients in the office space with the advantage of effectively managing the system. However, if the requirement to multimedia applications can not be satisfied, the thin-client computing would disappear from the prominent computing platforms. Therefore, it is natural requirement to support the multimedia applications in thin-client environment.

The streaming media service should satisfy the real time requirement to support the QoS for clients. The reason is that the ceasing and jittering streaming media are unmeaning on the side of clients. In particular, to provide QoS for clients, it is inevitable to support the high quality of audio sound together with smooth image streams. To arrive at the reasonable QoS level, the audio stream should synchronize with the corresponding image stream. For no jittering images and high quality audio, the intrinsic property of streaming media should be reflected in the design of thin-client system.

The decoding for streaming media is a computation-intensive job. Since the thin-client on mobile device has the limited hardware resources, it is difficult to decode the streaming media to the level of guaranteeing the QoS. To address this problem, the existing solutions suggest that the servers would be charged for decoding the streaming media and thin-clients have a role to update only the changed areas in their screen. However, these approaches supply severely low image quality for thin-clients and do not support the audio streaming. Even if the audio stream is provided, the synchronizing with the corresponding image stream would not be supported completely. The mismatched order between the image stream and the audio stream causes clients a great deal of dissonance. In addition, servers perform all procedures to decode streaming media so that they are easily saturated even in a small number of clients.

In this paper, The iMedier(intelligent Media Player) system is proposed to provide the improved QoS for streaming media in thin-client computing on mobile clients. Since the proposed method reflects the intrinsic property of streaming media, it provides both the enhanced video quality and the audio streaming fully synchronizing with image frames. The iMedier is based on the distributing computing concept for streaming media decoding. To arrive at this goal, first, the iMedier system proposes the RMDP(Remote Multimedia Display Protocol) to support the data transportation between the server and thin-clients. Second, since the YUV format instead of the RGB format is used as the data of display update, the image quality is improved due to the completed transportation of image data. Third, to provide the high quality audio streaming, the synchronization data are generated based on the decoded image frames and audio frames. Since the synchronization points are sent to clients along with streaming media, the high quality audio streaming can be provided.

The rest of this paper is organized as follows. Section 2 describes related work for our research. In section 3, the iMedier system is proposed to achieve the QoS streams in the thin-client computing environment. Section 4 explains our actual experimental environment and performance metrics for QoS. In section 5, the performance of the iMedier system is evaluated and compared to other thin-client system. Section 6 concludes the paper.

2 Related work

As representative thin-client solutions in these areas, there are Citrix MetaFrame [7], Microsoft Terminal Services [8], AT&T VNC(Virtual Network Computing) [3] and Tarantella [9]. These products have their originality technologies to support the thin-client computing environment. They are distributed as the free software or the commercial product. In some specific solutions, the parts of software function were merged into the hardware so that they have been sold as total merchandise.

The VNC(Virtual Network Computing) system had been suggested to exploit the high performance computing power in the side of servers. This system may allow clients to exploit the servers' resources throughout the user graphic interface and widely distributed to commercial products [3, 4]. In addition, to execute the Web browsing on the thin-client computing environment, some researches were undertaken for the performance improvement and the effective resource utilization in the limited wide band network. In particular, in the low hardware specification, the sources of performance bottleneck were analyzed and the major factors dominating the total performance were found [5]. In addition, recently the advancement in wireless network technologies has enabled the streaming media service on the mobile devices such as PDAs and cell phones. According to the changing working environment, the researches for the thin-client system are moved into the mobile thin-client computing environment [1, 2, 6, 12].

The previous thin-client solutions are only focused on the text-based working environment with user graphic interfaces. However, it is natural phenomenon to support the multimedia applications in thin-client environment. In particular, there is a strong preference for the streaming media service in many multimedia applications. However, since the streaming media have larger and more complex data than the traditional text and picture image data, the large amount of network traffics and the high performance computing ability are inevitable to support the QoS streams [11,13]. On the other hand, the wireless network has low bandwidth channels and many mobile devices compose of limited hardware specifications. Thus, it is necessary to research a new thin-client solution that supports the QoS of the streaming media in the mobile working environment.

3 Intelligent Media Player System

To support the QoS in thin-client computing environment, the intrinsic characteristic of streaming media should be reflected on both the server side and the client side. In this paper we proposed the iMedier (intelligent Media Player) system to provide the improved QoS for thin-clients. The iMedier system is composed of the iMedier server and the iMedier client part. The Figure 1 shows the total architecture of iMedier system. When a client requires the streaming media service, the corresponding media is decoded in the server and streamed to the client side. The iMedier client plays the role of terminal emulator. In addition, it performs the conversion from the YUV data to the RGB data and displays the updated image data into the screen.

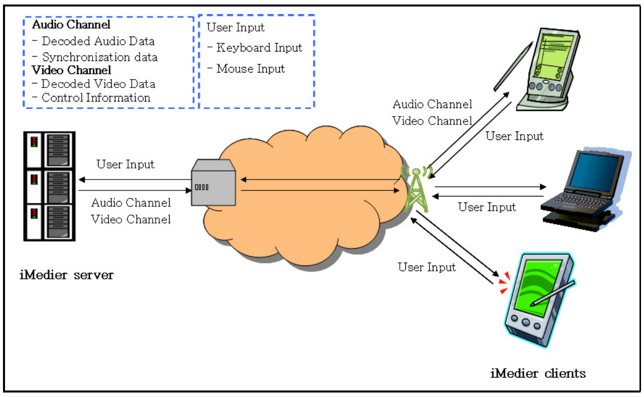

Fig. 1. Architecture of iMedier System

3.1 Screen Update Policy

In the previous thin-client computing environment, the updated data for the screen are sent to a client as the RGB format. However, since the RGB format represents the color difference space without the compression, the high data transfer rate is required. Therefore, the bottleneck phenomenon in the network path is inevitable. To address this problem, the iMedier uses the YUV format of 4:2:0 form for representing the color difference space, instead of the RGB format. When using the YUV format, the total transfer data are reduced to roughly the half of original amount. The client just performs the translating step from YUV data to RGB data. In contrast to the existing thin-client solutions, since the conversion procedure to the RGB data is eliminated in the server, the burden of the server can be relieved in the iMedier system.

3.2 Audio Synchronization

Since the existing thin-client solutions were focused on the text-based working environment, the audio function is overlooked. In part, although the audio streaming is provided, the synchronization problem appears in between the audio streaming and image streaming. The mismatching between the image stream and the audio stream causes clients a great deal of dissonance. To provide the QoS in the streaming media service, it is inevitable to supply the audio stream synchronizing with the corresponding image frames.

The Figure 2 shows the synchronization procedures in the level of the sequence diagrams. As shown in this Figure, the iMedier system has the objects for server, sender, receiver and client. After initialing the audio and image channel, the server begin to read the media data from its disk and to execute the de-multiplex process. After that, firstly the decoding for audio data is performed and is sent to the client. As shown in this Figure, before the decoded image data are transported to the client, the synchronization data should be sent to the client. The decoding for image data begins to execute when every synchronization time reaches. The client also sets up the initial stage for both the audio and image channels. Firstly, the decoded audio data are received from the server and they are played in the audio device. After the received YUV data are changed to the RGB data, the image streaming is played at every synchronization point.

Since the audio data should be played within the more narrow time interval than the image data, the time line of synchronization is depended on the audio sampling rate. Therefore, the audio data should be decoded and played firstly. After that, if the image data are decoded and played on the corresponding time

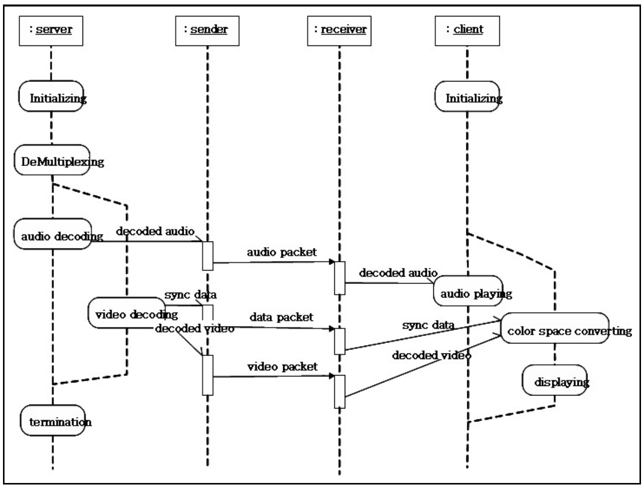

Fig. 2. Sequence Diagram of the iMedier System

line, the synchronization between the audio and the image is done. That is the key idea to support the audio in our iMedier system.

3.3 Remote Multimedia Display Protocol

The RMDP(Remote Multimedia Display Protocol) has a role for transmitting the video and audio data decoded on the server and also providing with the synchronization data. The Figure 3 shows the basic structure of channels in the RMDP. To transmit the updated screen data, the RMDP uses both an image channel and an audio channel. The audio channel transmits the audio and synchronization data. The image channel is used for handling the image data and the control information.

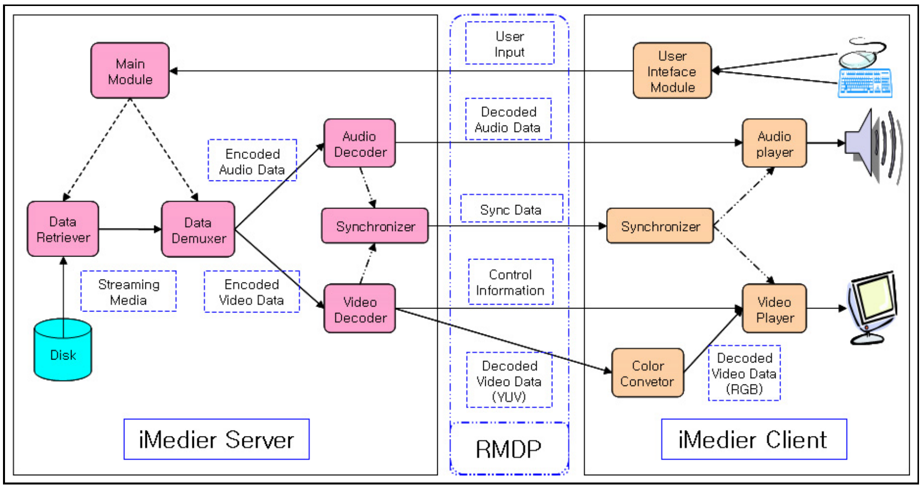

Fig. 3. Channels in RMDP

4 Performances Evaluation

4.1 Tested Thin-Client Systems and Streaming Media

The iMedier system is implemented on the Linux system. The compressed format of audio media is MPEG Layer-III and the OSS library is used as an output

Table 1. Specification of Thin-Client Systems

	Server	Client
CPU	AMD Atholon MP2000	206MHz StrongARM
Memory	1GB SDRAM	128MB SDRAM
Disk	Segate 36G(SCSI)	32MB Flash ROM
Network	100Mbps Ethernet	11Mbps 802.11b
OS	Linux	ARM Linux(Linupy)

Table 2. Specification of Streaming Media

Image	video Size (H x V)	240 x 180
	Color (bits)	16
	Frame rate(number/sec)	8
	Running Time (sec)	90
Audio	Sampling rate (Hz)	22050
	Channel	2
	Quantization (bits)	16

device. The Table 1 shows the configuration of a server and clients used in our experiments. The Table 2 show the specification of tested streaming media.

4.2 Performance Metrics for Image Quality

To evaluate the image quality, we use the slow-motion benchmarking technique [10]. In this technique, the packet traffics are measured under both the 1 fps(frames per second) rate and the 30 fps playing rate. Actually, even if any clients do not see the movie on the 1 fps rate, it is needed to measure the completed amount of media data transferred from the server to the client. On the other hand, the 30 fps rate is the normal play mode. As comparing the amounts of transferring data in two frame rates, the amount of data loss on the 30 fps rate can be measured. As mentioned above, the data loss has a negative impact on the image quality.

$$IQ = \frac{\left\{ \frac{\left\{ \frac{DataTransfer(30fps)}{PlaybackTime(30fps)} \right\}}{IdealFPS(30fps)} \right\}}{\left\{ \frac{\left\{ \frac{DataTransfer(1fps)}{PlaybackTime(1fps)} \right\}}{IdealFPS(1fps)} \right\}} \qquad (1)$$

The above equation (1) represents the numerical formula about the image quality. It is based on the ratio of the data transferred on two frame rates. In addition, to reflect the real time requirement, the total playback time is added in this equation.

5 Performance Evaluation

In our implemented thin-client system, we evaluate the performance of the iMedier system and the RealVNC system. The RealVNC system is a thin-client solution developed in the AT&T and involves in the Open Source Projects [3].

The Figure 4 show the total data transferred on the 1 fps rate and the 8 fps rate. In mobile devices, the 8 fps is regarded as the normal play mode. As shown in this Figure, the VNC transferred the data of 32 MBytes and the iMedier takes 53 Mbytes on the 1 fps rate. The VNC server compresses the screen update

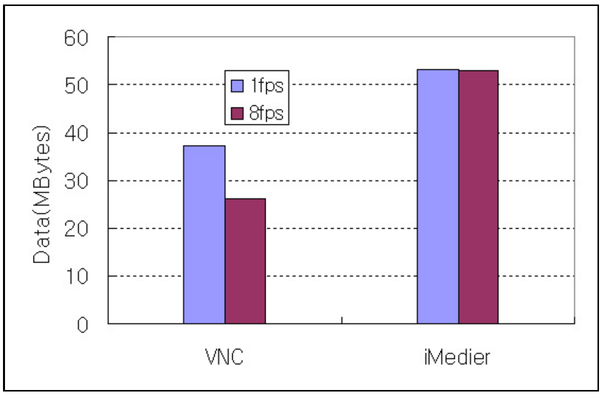

Fig. 4. Total Transferred Data

data with high compression rate to support the low network bandwidth of the mobile environment. After that, the server sends the highly compressed data to the client. However, this approach in the VNC system aggravates the quality of media play because mobile devices are usually operated with the limited resources like the low CPU computing ability. For the streaming media service, this real time requirement should be satisfied. From this reason, the ceasing and jittering phenomenon in the client screen occurred in our experiment. Thus, the QoS could not be expected in the VNC system.

In addition to the low image quality, the VNC system does not provide the audio streaming for the client. However, the iMedier system supports the audio streaming synchronizing with the image data. The transferred data in the iMedier system involves the image data as well as the audio data. The amount of audio data is calculated as multiply the bit rate and the playing time. The bit rate is obtained as the multiplication of the sampling rate and the quantization rate and the number of channels. Therefore, the amount of transferred audio data is 22050Hz 16bit 2bit 90 second=7.6Mbytes. In the Figure 4, the total transferred data in the iMedier system includes these audio data.

In the 1 fps rate, the VNC transports the data of 37 Mbytes. However, only the 26 Mbytes data are transferred in the 8 fps rate. The reason for this data loss is similar to the results in the desktop environment. The VNC does not consider the intrinsic property of streaming media in the screen update policy. If the client does not treat the screen update data within the limited time, the server discards large amount of transferring data. In addition, the VNC system is designed to send the update data in only the case where the update event occurs, instead of updating the screen as much as the designated frame rate per second. From these reasons, the transferred data in the 8 fps rate is less than those of the 1 fps rate. As a result, the VNC solution can not provide the streams guaranteed the QoS for mobile clients.

On the other hand, since the iMedier is designed to reflect the intrinsic property of the streaming media, the amount of transferred data in the normal play mode

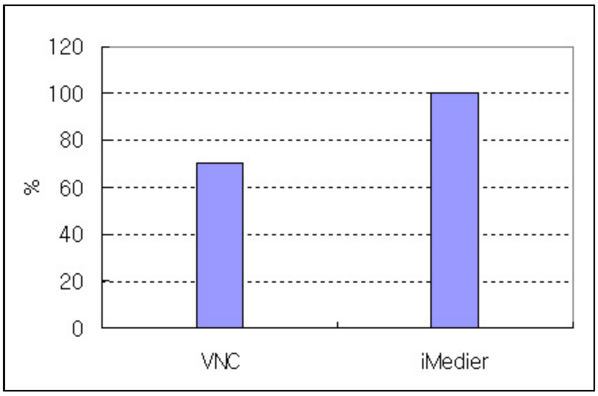

Fig. 5. Comparison of Image Quality

is equal to that in the 1fps rate. Thus, the iMedier updates the screen according to the designated frame rate so that the completed data can be received in the client without the data loss. Base on these experimental results, we confirm that the iMedier can provide the streams guaranteed the QoS to clients. From the results of the Figure 4, the image quality of the streaming media in the mobile device is calculated by using the equation 1. The Figure 5 shows the image quality of both thin-client systems working in the mobile environment. Regardless of the low data transfer rate, the VNC system records the image quality of 70% in the normal play mode. However, the iMedier shows the image quality of 100%.

6 Conclusions

In this paper, the streaming media service is studied in the thin-client system. To support the QoS in thin-client computing environment, we proposed the iMedier system based on the RDMP protocol and the distributed computing to the MPEG decoding. Since the iMedier used the YUV format instead of the RGB format, the image quality was improved by the reduced transportation data. In addition, by synchronizing the image frames and the audio sound during the decoding time, the high quality of audio streaming could be supported.

The iMedier system provided not only the screen image of high quality without jittering and ceasing phenomenon but also the audio stream completely synchronizing with the corresponding images. Since the iMedier system distributed the decoding operation between the server and clients, the server can relieve its own computing load. As a result, the more number of streams can be serviced. In experiment results, when compared to the VNC system, the iMedier showed the enhanced image quality of 30% in the mobile device environment.

In our future work, we plan to improve the total system performance by devising the optimized compression technique for the YUV data.

References

1. Krikke, J., "Thin clients get second chance in emerging markets," IEEE Pervasive Computing, Vol. 3, Issue 4, pp. 6-10, 2004
2. Volchkov, A., "Server-based computing opportunities," IEEE IT Professional, Vol. 4, Issue 2, pp. 18 23, 2002
3. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, Andy Hopper, "Virtual Network Computing," IEEE Internet Computing, Vol. 2, Issue 1, pp. 33-38. 1998
4. Brian K. Schmidt, Monica S. Lam, J. Duane Northcutt, "The interactive performance of SLIM: a stateless, thin-client architecture," ACM SOSP'99, pp. 31-47, 1999.
5. S. Jae Yang, Jason Nieh, Matt Selsky, and Nikhil Tiwari, "The Performance of Remote Display Mechanisms for Thin-Client Computing," USENIX, June 2002
6. S. Jae Yang, Jason Nieh, Shipa Krishnappa, "Web Browsing Performance of Wireless Thin-client Computing," International WWW Conference, May 2003
7. T. W. Mathers and S. P. Genoway, "Windows NT Thin Client Solutions: Implementing Terminal Server and Citrix MetaFrame," Macmillan Technical Publishing, Indianapolis, IN, Nov. 1998
8. B. C. Cumberland, G. Carius, and A. Muir, "Microsoft Wndows NT Server 4.0, Terminal Server Edition: Technical Reference," Microsoft Press, Redmond, WA, Aug. 1999.
9. A. Tirumala and J. Ferguson, http://dast.nlanr.net/Projects/Iperf
10. S. J. Yang, J. Nieh, and N. Novik, "Measuring Thin-Client Performance Using Slow-Motion Benchmaking," USENIX, June 2001
11. R. W. Scheifler and J. Gettys, "The X Window System," ACM Transactions on Graphics, Vol. 5, Issue 2, pp. 79-109, April 1986
12. Albert Lai, jason Nieh, "Limits of Wide-Area Thin-Client Computing," Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 228-239, 2002
13. J. Danskin, P. Hanrahan, "Higher Bandwidth X," Proceedings of the Second ACM International Conference on Multimedia, pp.89-96, 1994