A REMOTE THIN CLIENT SYSTEM FOR REAL TIME MULTIMEDIA STREAMING APPLICATIONS

Kheng-Joo Tan[1], Guo-An Jian[1], Chia-Hsiang Chen[2], Dou-Cheng Chang[1], Chun-Kun Wang[1], Maw-Ren Lee[1], Peng-Sheng Chen[1], and Yuan-Sun Chu[3]

[1]Department of Computer Science, National Chung Cheng University, Taiwan, R.O.C.
[2,3]Department of Electrical Engineering, National Chung Cheng University, Taiwan, R.O.C.
[1]{ccj96m, wck98m, ctc97m, lmr98m, chienka, pschen, shiwulo, jiguo}@cs.ccu.edu.tw, [2]m9779@cn.ee.ccu.edu.tw, [3]chu@ee.ccu.edu.tw

ABSTRACT

This demo proposes a remote thin client system for real time multimedia streaming over VNC under extremely low bandwidth situations. A remote frame can be split into two parts, i.e. high motion part and low motion part, and transmitted through the Internet from servers to clients according to the proposed hybrid RTP protocol. A Dynamic Image Detection Scheme (DIDS) is proposed to automatically detecting the high motion part of a frame with only 1% of extra CPU loading. By using the proposed thin client system, we can achieve about 22 fps of real-time SIF video streaming with good video quality under 32 KByte/s of bandwidth limitation, which speeds up about 172 times in remote frame display when compared to pure VNC.

Keywords— Thin client, VNC, Image detection, Video streaming.

1. INTRODUCTION

In thin client systems, clients are mostly light weight portable devices with low computing power. They are mainly responsible for displaying graphical output and handling user events such as keyboard or mouse events. In contrast to clients, servers are powerful to handle more complicated operations such as image capturing, image encoding, and so on. Remote desktop is a kind of thin client systems using Remote Frame Buffer Protocol (RFB) or Remote Desktop Protocol (RDP) to remotely control servers by clients like personal computers, notebooks, PDAs, and even smart phones. This mechanism allows thin clients to keep light weight while sharing the computing power from servers. But in most applications especially in displaying multimedia contents, real time transmission is the main design challenge caused by extremely high data bandwidth. Thus, the current thin client systems adopting VNC cannot achieve a smooth display on multimedia contents remotely through thin clients under the limited data bandwidth. In order to solve this problem, there have been some solutions proposed in the literature [1-3].

In [1], high/low motion scenarios are tested on remote thin client systems. The experimental results show that in high motion scenario such as video streaming and 3D gaming, extra hardware is a must on both the server and client sides to resolve the high bandwidth problem.

In [2] and [3], similar hybrid remote thin client systems with different motion detection algorithms are used to solve the real time streaming problem mentioned in [1]. In the thin client systems [2, 3], remote frames are classified into high and low motion modes. If the remote frame is belonging to high motion mode, the H.264 encoder is used to encode the frame before sending it to clients. On the contrary, if the remote frame is belonging to low motion mode, the traditional RFB protocol is used to update frame buffer pixels of the client devices. In other words, either high or low motion mode will be chosen before a remote frame is sent to clients. The hybrid remote thin client structure from [2-3] can effectively reduce data transfer bandwidth through video compression, but it also leads to some critical problems such as degradation of text quality and high computational complexity caused by motion detection algorithm.

In order to solve the above mentioned real time streaming problems and ensure good video quality in remote thin client systems, we propose a real-time, low-complexity Dynamic Image Detection Scheme (DIDS) to split each remote frame to be parts of low motion (like still image and text) and high motion like video. For low motion part, traditional RFB protocol is used to transmit data to clients to ensure good quality. As for high motion part, we firstly encode it using an optimized H.264 encoder, and then send it to clients by using UDP or TCP protocol.

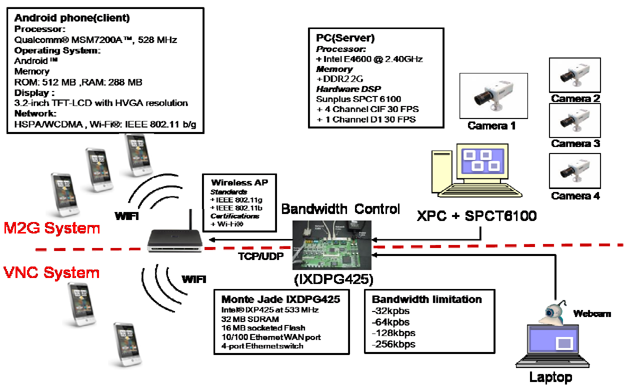

Fig. 1. Prototype of the proposed remote thin client system

2. IMPLEMENTATION RESULTS

The proposed thin client system has been implemented and fully tested under several transmission bandwidth situations. As shown in Fig.1, a laptop and a mini computer with a PCI card consisting of the H.264 encoder realized in a DSP processor, i.e. SPCT6100, are used as the servers for the traditional VNC system and the proposed system, respectively. Besides, bandwidth limitation program is porting on IXP425 development board. As for clients, the cell phones with Android operating system are chosen in the demonstration systems. If without any bandwidth limitation, both pure VNC and the proposed thin client system achieve over 30 fps of SIF video in remote frame display. But as shown in Table 1, if we limit the bandwidth to be 32 KByte/s, the remote frame display rate with pure VNC system is degraded to be about 0.13 fps of SIF video. It is caused by the fact that the video frame is viewed as individual image frames and transferred from the server to client pixel by pixel without any compression if pure VNC system is used. On the contrary, under the same bandwidth, the proposed thin client system performs much better than the pure VNC system. It achieves 22.46fps of SIF video in remote frame display which is about 172 times faster than the result of using pure VNC system. As for the rest of bandwidth limitation settings, the proposed system achieves about 28 fps of SIF video display. But the pure VNC system achieves only 0.25fps, 0.58fps, 1.00fps of SIF video display, respectively. The results show that the proposed system achieves the real-time transmission under strict bandwidth situations.

Table 1. Remote frame rate comparison of the proposed system with pure VNC system under different bandwidth restriction

Bandwidth limitation	Pure VNC system	Proposed system
32Kbytes/s	0.13 fps	22.46 fps
64Kbytes/s	0.25 fps	28.61 fps
128Kbytes/s	0.58 fps	28.31 fps
256Kbytes/s	1.00 fps	28.43 fps

3. REFERENCES

[1] L. Deboosere, J. De Wachter, P. Simoens, F. De Turck, B. Dhoedt, and P. Demeester, "Thin Client Computing Solutions in Low- and High-Motion Scenarios," Proc. IEEE Third International Conference on Networking and Service, pp. 38-43, June. 2007.

[2] P. Simoens, P. Praet, B. Vankeirbilck, and J. De Wachter, Design and Implementation of a Hybrid remote display protocol to optimize multimedia experience on thin client devices," Proc. IEEE Telecommunication Networks and Applications Conference, pp. 22-25, Dec. 2008.

[3] D. De Winter, P. Simeons, L. Deboosere, F. De Turck, J. Moreau, and B. Dhoedt, P. Demeester, "A Hybrid Thin-Client Protocol for Multimedia Streaming and Interactive Gaming Application" Proc. Int. Workshop on Network and Operating System Support for Digital Audio and Video, Aug. 2006.

[4] Joint Video Team (JVT) reference software JM 14.2.

[5] "Real VNC," available via http://www.realvnc.com/

[6] "RTP Payload format for H.264," available via http://tools.ietf.org/html/rfc3984

[7] "RFB Protocol," http://www.realvnc.com/docs/rfbproto.pdf