在一實驗系統中高畫質影像傳輸的封包遺失問題
Pack-Lost Problems of High-Quality Video Transmission in An Experimental System*

葉春超 [1]　莊志洋 [1]　Dirk Steinberg[2]
台灣大學資訊系 [1]　德國國家資訊研究中心 [2]
Chun-Chao Yeh[1], Jie-Young Juang[1], and Dirk Steinberg[2]
CSIE NTU Taiwan[1]　IMK GMD Germany[2]

摘要

本文探討高畫質影像在傳輸中封包遺失問題。主要應用如在分散式環境電影電視等專業數位影像後製作流程，或在分散式虛擬工作室中的影像傳輸。其要求高畫質且近乎無失誤的影像傳送。目前大多數影像傳輸的網路協定採UDP Protocol，然而從實驗中我們發現UDP Protocol並非十分適合如此類高畫質的影像傳輸。

關鍵字：影像傳輸，數位影像製作，ATM網路，RTP及時傳輸協定，封包遺失

Abstract

In this paper, we investigate the issues of packet loss in high-quality video transmission. A important applications of high-quality video transmission is in digital video production areas. Examples include TV program/film production and distributed virtual studio. Transmission of high-quality video imposes more stringent quality requirements than that of conventional video streams does. More network bandwidth is required and less frame loss is allowed. Currently, most of video transmission systems are based on UDP protocol due to its simplicity and low overhead in packet handling. However we found that, under UDP connections, even with sufficient network bandwidth and powerful processing hosts, systems still cannot guarantee acceptable video quality. To investigate the packet transmission problems, we made a trace of lost packets and analyzed their behavior in an experimental system. issues on applying conventional TCP or UDP protocols to high-quality video transmission are discussed.

Keywords: *Video Transmission, Digital Video Production, ATM Network, RTP Real-time Transport Protocol, Packet Loss*

*This work is supported by NSC Taiwan, GMD and DAAD Germany, under NSC/DAAD academic cooperating programs.

1 Introduction

Distributed multimedia applications such as distributed virtual studio, tele-presence applications, distributed VR-simulation and distributed video archiving, are major applications of distributed video production based on emergent new computer and communication technologies, as depicted in GMD DVP(Distributed Video Production) project[1]. One of main focus of DVP project is on professional digital video production for digital video/audio broadcasting industry, which aims to enable professional producers and editors from the video production and post-production industry to use audiovisual material in LAN/WAN networks.

One of key issues in distributed video production applications is how to deliver the audiovisual material, especially video data, through networks. First, the required network bandwidth would be much larger. This is because that high-quality images are required for professional production. For example, digitization of PAL video signal in the standard 4:2:2 representation with 720x576 image format, each pixel made up of 16 bits(8 bits for luminance and 4 bits each for the two color-difference signal), would generate a data stream with data-rate of 166 Mbits/sec(720x576x25x16). Although the raw image-data can be compressed before transmission, extensive compression is not likely to be done since compression would cause quantization error and consequently would result in bad video quality, which would not be suitable for professional production. A proper estimation of bandwidth requirement is about a few Mb/s up to 50 Mbits/sec or more[6]. Second, transmission quality in terms of frame loss and delay is more stringent. Only few frames lost is allowed. Meanwhile, problems due to non-negligible delay could be serious if multiple streams have to be synchronized with each other. For example, in a virtual studio where the performers are distributed in differ locations, large delay of a source stream would come out apparently inconsistent.

Causal frame loss might not be a problem for most of video transmission applications such as video-conferences[5]. It could be harmful to professional video production. Moreover, since frame-loss is usually bursty instead of well amortized, transmission faults are more sensible in high quality video transmission. Most of existing video transmission systems, for example Mbone[3] and *vic*[4], apply UDP (User Datagram Protocol) which provides a connectionless connection. Packets might be lost on the way of transmission. However, for UDP connections, there is no guarantee of error-free packet transportation. No packet error detection and correction schemes will be done. One of main causes of packet loss is shortage of system resources including network bandwidth, processing power, and memory buffer.

To investigate the packet transmission problems in high-quality video transmission, we made a trace of lost packets and analyzed their behavior in an experimental system. In this study, the ATM network bandwidth and the host processing power are well enough to handle the data rate generated by the compressed-video streams. Nonetheless, problems are still no-negligible. Improper choices of packet size may make the problem worse.

In the following section, we describe transmission of compressed-video streams. Experiment setup is depicted in Section 2. Results and analysis are presented in Section 3. Finally, concluding remarks are given in Section 4.

2 Experiment setup

Experiments of high-quality video transmission are set up in GMD as shown in Fig.1. Both SUN UltrSparc and SUN Sparc10 are as source and sink hosts. Each of them has an OC-3 ATM link (155 Mbits/sec) connection connected to Fore ATM switches. A JPEG-compression coding device, SCSIVideo, attached to each host with fast-wide SCSI interfaces. Detailed descriptions of the hosts are depicted in Table 1.

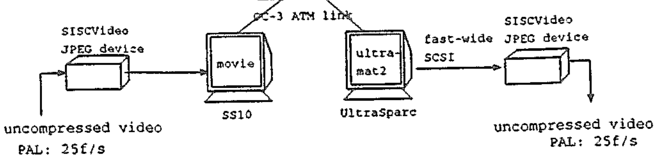

Figure 1: Experimental system for high-quality video transmission with JPEG-compression

Host Name	movie	ultramat2
System Model	SUN SS 10SX	SUN UltraSparc 1
Main memory	96 MB	124 MB
CPU #	2	1
OS	SunOS 5.5	SunOS 5.5
ATM link	OC3 155 Mbps	OC3 155 Mbps
M-JPEG codec	SCSIVideo	SCSIVideo

Table 1: Host description

Compressed-video streams are packetized into RTP(Real-time Transport Protocol)[2] packets. Measurements of system-load with different compression ratio and network configurations are conducted. Conventional UDP/IP and TCP/IP network protocols are considered under the RTP protocol. Other system parameters involved in the experiments are described as follows.

- *compression ratio*: Different quality of video compression can be made by the *q* factor provided by the JPEG coding device. The q factor ranges from 0 to 100. Higher value results in better-quality/larger-size compressed image from the same source image. Fig. 2 shows effects of the compression ratio on the video quality of the tested video clips used in the experiments.

- *network protocols* : Both UDP and TCP are chosen as network protocols to encapsulate RTP packets. To improve connection quality, network buffers are enlarged from 8 KBytes to 64 KBytes. Other parameters are shown in Table 2. In which, value in parentheses is default value set by the SUN operating systems.

- *LAN architecture*: Both ATM and Ethernet connections are available in the host workstations. Table 3 shows effective network bandwidth measured by TCP throughput via different network interfaces between the two hosts. The experiments are based on *tcpblast* network benchmarker[7]. While both ATM classic IP and LAN Emulation (Ethernet) are available in the ATM network, only classic IP interface is measured since it provides higher bandwidth than the Ethernet LAN Emulation interface does.

- *compressed-video streams*: The compressed-video

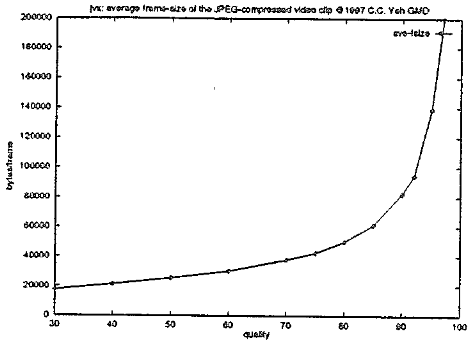

Figure 2: Average frame size for different compression ratio(q= 30 to 97)

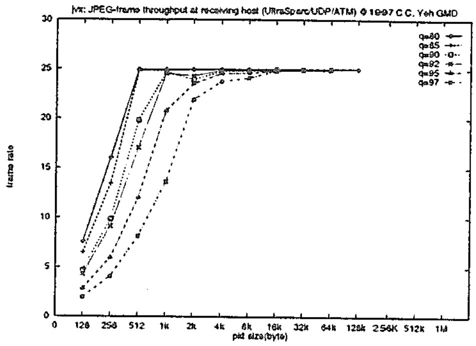

Figure 3: Frame throughput at the receiving host(q=80 to 97, UDP/ATM)

Host Name	movie	ultramat2
udp_do_checksum	on	on
udp_xmit_hiwat	65535 (8192)	65535 (8192)
udp_recv_hiwat	65535 (8192)	65535 (8192)
tcp_recv_hiwat	65535 (8192)	65535 (8192)
tcp_recv_hiwat	65535 (8192)	65535 (8192)

Table 2: Network parameter setting

Network Interface	MTU	TCP throughput
Ethernet(10 BaseT)	1500	9 Mbps
ATM		
Classic IP	9180	71 Mbps
LAN Emulation(Ethernet)	1500	23 Mbps

Table 3: TCP throughput between the hosts (measured by *tcpblast*)

streams are taken from the same sequences of image frames with different compression ratio setting by q factors ranging from 30 to 97. The image frames are from a video clip of 1994 World Cup Football Championship final game, Italy vs. Brazil. About sixty frames are sampled from the source video clip. Figure 2 shows the average frame-size of the tested video clip after compression using different compression ratio setting. Compressed-video streams are pre-sampled and are stored in disks at the sending host. Thus, same compressed-streams can be taken for different experiments.

- *transmission software*: The RTP code is compliant to both the RTP-JPEG payload format and the RTP protocol except RTCP protocols. Multithread and multiple frame-buffer(four buffer) schemes are employed in the receiving processes. On the other hand, the sending process is basically implemented by a single thread/ single frame-buffer scheme.

3 Results and analysis

Figure 3 shows the result of the compressed-video transmission in term of frame throughput. In general, the system would prefer large packet-size to achieve higher throughput. The highest throughput is achieved at 25 frames per second, which is bounded by the frame rate of PAL video according to the PAL standard. Via UDP connection, packet handling at receiving hosts is more sensitive to network status. Bursty packet traffics could cause receiving buffer overflow. Therefore, the results of system resource-utilization observed at receiving sides could be some-

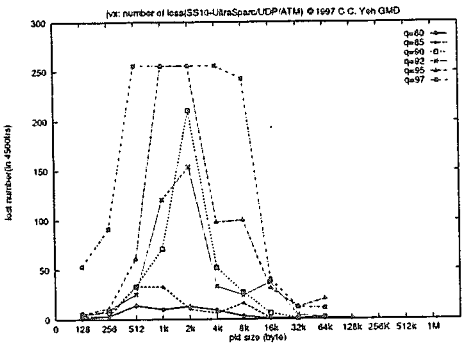

Figure 4: Number of packet-loss faults during transmission (total frames = 4500)

what different over the time, while those observed at sending sides has less variance. Choices of packet sizes make significant difference to the end results. Our experiments show that host CPU utilization can be as low as 20% in the handling of the JPEG-compression streams, which could generate a data rate of about 40M bit/s as indicated in Fig. 2. This indicates that the host computing power is sufficient to handle the streams, if packet-size is properly set. Small packet-size results in higher packet rate and interrupt rate, and may bring large burden to the hosts. More detailed descriptions about frame throughput and system load can be found in [8].

Packet loss makes corresponding frames incomplete and results in frame loss. Figure 4 shows the number of packet-loss faults [1] for each compressed-streams in the experimental system. Most of incomplete frames are caused by a small number of packet-loss faults. In addition, we found that, in each packet-loss faults, only a few packets are missing. This indicates that only a few retransmissions are needed to rescue most of the incomplete frames, if there were some retransmission mechanisms enhancement in the UDP protocol. Meanwhile, for most of the cases, packet-loss rate is below 1%. Figures 5, 6, and 7 show the frame loss and packet loss from different points of view at the same UDP/ATM frame-transmission experiments. In general, the plotting curves shown in Fig. 4 looks like a *mountain*. Exact shapes of the curves depend on the results of packet receiving. The top of the *mountain* is around at the point where the chosen packet size makes the throughput largest at the sending side. That is at 512 Byte for q=80 and 85, 1 KByte for q=90 and 92, and 2 KByte for q=95 and 97, as shown in Fig. 3. Before that point, data rate of the packet streams is not fully loaded, and consequently less likely cause packet loss. In general, the number of lost packets declines as the packet-size getting larger. For the cases of packet-size larger than 4 KByte, packets lost become quite insignificant.

Figure 7 indicates that only a few packets are lost in each packet-loss faults. About one frame is lost per packet-loss fault. This means that, for most cases, only one retransmission is needed. That is, under such circumstances, with only few overhead for packet retransmission, quality of video playback can be improved substantially. Also, we found that the system load had never been heavy. It would be possible to utilize available system resources to do the retransmission. This conjecture is verified by another tests which sending the same RTP-streams through TCP connections. Results of the TCP connection tests are shown in Fig.8. Detailed measurements of system load can be found in [8]. Additionally, from the measurements, we found that system load in the case of TCP did not increase significantly, comparing with that in the case of UDP. Comparing with TCP, UDP traffics was considered to be more bursty , although it is not necessarily true, since usually packets delivered through UDP connections are sent out as soon as they could. No traffics feedback is sent back as that with TCP flow control mechanisms. Bursty traffics would be more likely to cause packet loss due to resource contention.

4 Concluding remarks

In this paper, we conduct an empirical study on high-quality video transmission in an experimental system, and discuss transmission problems. Detailed system behavior for high-quality video transmission is presented based on empirical measurement data. From our study, we argue that instead of UDP or TCP, a more flexible transport protocol is needed for high-quality video transmission applications such as the one we investigated in this study.

In this study, we found that even under enough computing power and network bandwidth, still there is non-negligible frames lost. Most of the frames loss results from a small number of lost packets. Moreover, only a limited number of packet-lost faults happen during the transmission. This provides hints that most of lost frames can be recovered without too much

[1] Here the packet-loss fault means the fault that the receiving host cannot get the right packet with expected sequent number. Since the packetized video streams are transmitted in order, each packet-loss faults means a group of consequent packets lost.

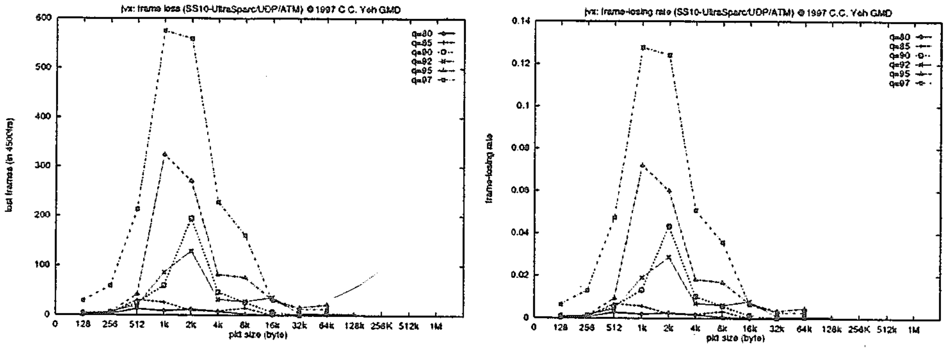

Figure 5: Number of lost frames and frame-loss rate(total frames = 4500)

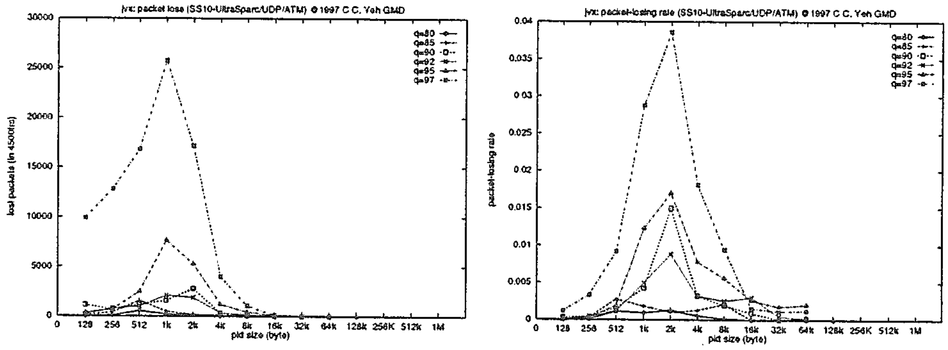

Figure 6: Number of lost packets and packet-loss rate(total frames = 4500)

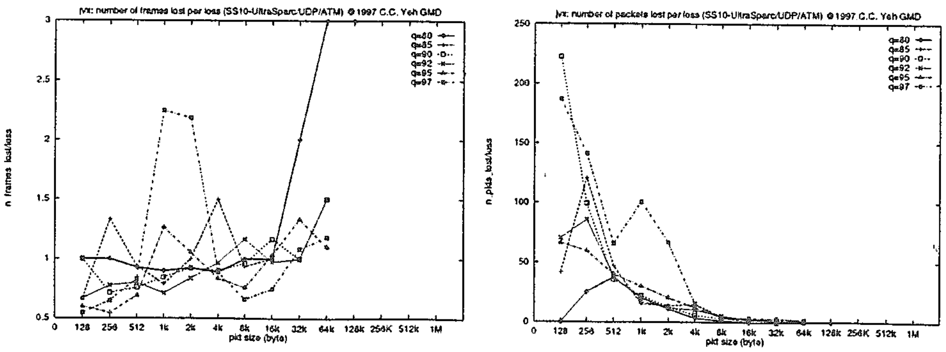

Figure 7: Ratio of lost frames and lost packets per packet-loss fault(total frames = 4500)

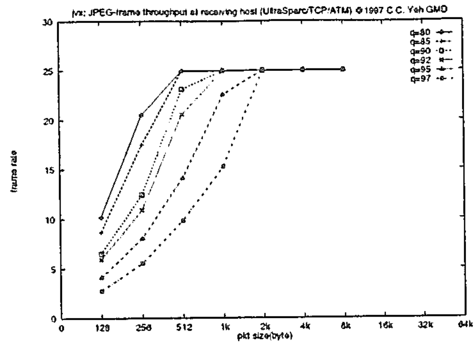

Figure 8: Frame throughput at receiving hosts(q=80 to 97, TCP/ATM)

packet-recovery overhead. Results from TCP experiments justify our conjecture.

Although we figure out the benefits to retransmit lost packets with TCP examples, we do not intend to endorse TCP as a proper transport protocol for high-quality video transmission. Since TCP protocol would recover any lost packets all the time, it might spend too much time to recover lost packets and thus makes the following frame packets miss the time to be handled. A better choice is a more flexible protocol to handle lost packets at proper time, not all the time. One possible solution is to enhance conventional UDP protocol with a mechanism to handle packet retransmission under some circumstances. Also, proper flow information would help more UDP packets survive from lost when packet traffics are congestive. Although RTP protocol provides flow information with RTCP packets, time granularity of the RTCP messages is too large. It is suggested to be greater than a minimum of five seconds according to the RTCP protocol. Apparently, it is not feasible for applications of high-quality video transmission.

References

[1] Manfred Kaul, The DVP Project. http://viswiz.gmd.de/DVP.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Internet Engineering Task Force, January 1996.

[3] Michael R. Macedonia and Donald P. Brutzman, "MBone Provides Audio and Video Across the Internet", IEEE Computer, April 1994, pp. 30-36.

[4] Steven McCanne and Van Jacobson, "vic: A Flexible Framework for Packet Video", ACM Multimedia'95, San Francisco, 1995, pp 511-522.

[5] Thierry Turletti and Christian Huitema, "Videoconferencing on the Internet", IEEE/ACM Transation on Networking, vol. 4, no. 3, June 1996, pp 340-351.

[6] "DVCPRO and Digital Video Recording", Panasonic Broadcast and Digital Systems Company 1996.

[7] Daniel Karrenberg, tcpblast package. software available in http://www.freebsd.org.

[8] Chun-Chao Yeh and Dirk Steinberg, "Hight-Quality Video Stream Transmission: A Case Study on JPEG-compressed Video Streams Over ATM Networks", GMD Technical Report, No. 1078, 1997.