Design and Implementation of Real-Time Digital Video Streaming System over IPv6 Network using Feedback Control

UEDA Kazunori † OHSAKI Hiroyuki †† SHIMOJO Shinji ‡ MIYAHARA Hideo ††

† Osaka University,
Osaka School of International Public Policy
1–31 Machikaneyama, Toyonaka, Osaka, JAPAN
ueda@osipp.osaka-u.ac.jp

‡ Osaka University, Cybermedia Center
5–1 Mihogaoka, Ibaraki, Osaka, JAPAN
shimojo@cmc.osaka-u.ac.jp

†† Osaka University,
Graduate School of Information Science and Technology
1–3 Machikaneyama, Toyonaka, Osaka, JAPAN
{oosaki,miyahara}@ist.osaka-u.ac.jp

Abstract

In this paper, we discuss a design of a real-time DV (Digital Video) streaming system, which dynamically adjusts packet transmission rate from the source host according to feedback information from the network. In our DV streaming system, the destination host continuously notifies the source host of network status (e.g., the end-to-end packet transmission delay and the packet loss probability in the network). The source host dynamically adjusts its packet transmission rate by lowering the quality of the video stream using a feedback-based control mechanism. Our DV streaming system achieves an efficient utilization of network resources, and prevents packet losses in the network. Thus, our DV streaming system realizes high-quality and real-time video streaming services on the Internet. By modifying the existing DVTS (Digital Video Transmission System), we implement a prototype of our real-time DV streaming system. Through several experimental results, we demonstrate the effectiveness of our DV streaming system.

1. Introduction

In recent years, technology related to the Internet has been rapidly advancing. Moreover, these technological advances have made sending and receiving a huge amount of data, such as high-quality video streams over the Internet, realistic. The quality of video clips provided by several VoD (Video on Demand) content providers has become better. VoD service will most likely become a powerful application of the next-generation Internet. The realization of such VoD service has been the focus of active research in the last few years [1, 2].

Several techniques used for transmitting a real-time video stream over the Internet exist. One of these techniques is called, “off-line transmission”, where a video clip is played on the user’s side when all video data have been transferred from the server to the user’s computer. With this technique, however, the user has to wait for the completion of the video data transmission. Thus, if the length of the video clip is not short, the user must wait for a long time and he or she often feels frustrated. Moreover, the user’s computer must be equipped with a huge storage capacity to store the entire video clip. To solve these problems, another technique called “real-time streaming” has been widely deployed in many video applications. With real-time streaming, the user’s computer receives video data while it plays the video clip. Using the real-time streaming technique, the user can start playing the video clip without waiting for the entire video clip to be transferred to the user’s side. This technique results in less waiting time for users than with the off-line transmission technique. Moreover, since several seconds of video clip data are stored on the user’s computer, the capacity of the computer’s storage can be small.

Although the real-time streaming technique is superior to the off-line transmission technique in various aspects, it is difficult to realize real-time video streaming over the Internet. For example, a large network bandwidth is required to transmit video clips in real time. In addition, the quality of the video stream may be severely degraded because of limited network bandwidth or fluctuating end-to-end packet transmission delay. If the network bandwidth is not sufficient for real-time video streaming, the user’s computer sometimes fails to play the video clip. Such a problem is caused by shortage of video data in the computer’s buffer when either the network bandwidth is insufficient or the packet transmission delay becomes temporarily large.

To solve these problems, a great amount of research has been carried out. For example, research has been proposed, which involves solutions that intentionally lower the quality of the video clip when the network bandwidth is limited [3-6]. In contrast, another approach has been proposed, which includes solutions that require a large buffer on the user's computer for absorbing fluctuations of packet transmission delays [7, 8]. However, even with these solutions, other problems still remain. With the former solution that is the technique of lowering quality of video clip intentionally, most of the existing codecs do not allow video quality to be changed at the server for utilizing the available network bandwidth. With the latter solution that is the technique of using a large buffer, prior to playing a video clip, a fair amount of video clip must be stored in the computer's buffer, so that the lag between transmitting video data from the server and playing a video clip on the user's computer is not negligible.

In this paper, we discuss the design for a real-time DV (Digital Video) streaming system, which sends DV streams with dynamically changing video quality according to feedback information from the network. In our DV streaming system, the destination host sends the feedback information back to the source host. The feedback information consists of the end-to-end packet transmission delay and the packet loss probability in the network. The source host then adjusts the quality of the DV stream to regulate the transmission rate of the video data based on the received feedback information. Because of such control, the number of packet losses in the network can be minimized, and the network bandwidth can be effectively utilized. Thus, high-quality and real-time video streaming over the Internet can be realized. We have implemented a prototype of our DV streaming system by modifying a part of the existing DV transmission system called the "DVTS" (Digital Video Transmission System) [9].

The rest of this paper is organized as follows. In Section 2, several conventional video streaming systems and their drawbacks are discussed. In Section 3, the overview of our DV streaming system, which consists of four building blocks, is presented. In Section 4, the prototype of our DV streaming system is explained, followed by several experimental results in Section 5. Section 6 summarizes this paper and discusses future work.

2. Related Work

On the Internet, various video streaming systems have been widely adopted. Most popular video streaming systems include RealNetworks's RealSystem [10], Microsoft Windows Media Technology [11], and Apple's QuickTime [12]. These video streaming systems and their proprietary video encoding formats have gained popularity in the market, in particular, from many content providers as well as from content creators. Other video streaming systems, which can transmit better quality videos such as MPEG [13] and DV (Digital Video) [9] also exist. For instance, DVTS (Digital Video Transmission System) allows IEEE 1394 devices to be connected via the Internet rather than via an IEEE 1394 interface. DVTS is publicly available as open-source software.

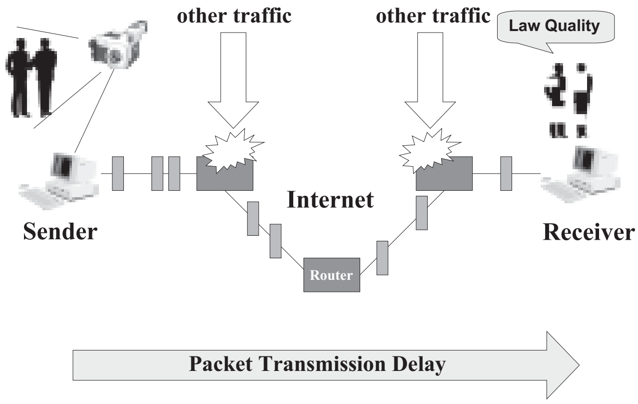

Figure 1. Quality Degradation in the Internet

DVTS has a mechanism that changes the DV frame rate to reduce the required network bandwidth for transmitting DV data. However, the DV frame rate of DVTS cannot be changed dynamically. On the Internet, the network bandwidth is shared by many users so that the available bandwidth for one user is heavily affected by the behaviors of other users. Hence, by changing the DV frame rate dynamically according to the current network status, a fair amount of DV data will be lost in the network because of network congestion, so that the quality of DV stream received by the user is considerably degraded by, for example, block noises caused by decoding errors (Fig. 1).

One possible solution for such a problem is to rely on the QoS (Quality of Services) control mechanisms in the network layer [14]. Although the QoS control mechanisms have been a hot topic in the IETF, it is unrealistic for current Internet users to receive benefits from QoS control mechanisms because of incremental deployment of those QoS control mechanisms. In other words, all or, at least, most routers between the source and the destination hosts must be replaced by a *QoS-ready* router in order for users to enjoy the support of the QoS control mechanisms. Since it takes quite a while for all routers on the Internet to become QoS-ready, another solution for realizing high-quality and real-time video streaming is urgently needed.

3. Designing a DV Streaming System

In this Section, we describe our real-time DV transmission system, which dynamically adjusts its transmission rate using a feedback-based control mechanism. This system utilizes network resources efficiently and achieves a high-quality and real-time video transmission over the Internet by dynamically adjusting its transmission rate according to the network congestion status.

3.1. System Overview

The system consists of the following four functional blocks (Fig. 2).

- Network Measurement Block
- Feedback Control Block
- Filtering Block
- Protocol Conversion Block

The basic operational algorithm of the system is as follows.

1. At the network measurement block in the source host, the end-to-end packet transmission delay and the packet loss probability in the network are measured from feedback packets sent from the destination hosts. Notification of these values are sent to the feedback control block, which will be explained below.

2. At the feedback control block in the source host, quantization scale or DV frame rate is updated by estimating how the current network status is different from the desired status. Macro blocks of each DV frame are re-encoded based on the quantization scale at the filtering block.

3. At the filtering block in the source host, DV frames are filtered to adjust the transmission rate to that specified by the feedback control block. The filtering block has two operational modes: (1) decoding and re-encoding of a DV frame with a smaller quantization scale, or (2) discarding some DV frames to adjust to a smaller DV frame rate.

4. At the protocol conversion block in the source host, a DV frame is encapsulated into several UDP packets. Each UDP packets is then encapsulated as an RTP (Realtime Transmission Protocol) packet.

MPEG (Motion Picture Expert Group) [15] format has been widely used as a video encoding format, which realizes a very high compression ratio. However, in our DV streaming system, DV is adopted instead of MPEG as the video encoding format because of the following reasons:

- DV video can be easily captured from a DV camera using the IEEE1394 interface.
- Even some laptop computers have IEEE1394 interfaces.
- A costly hardware-based encoder is required for real-time encoding of a high quality live stream with MPEG2.
- We can implement our DV streaming system by modifying a part of the DVTS, which is open-source software.

Although the DV format is adopted as the video-encoding format in our DV streaming system, the system can be easily extended to other video formats, such as MPEG2, by simply replacing the filtering block. In the following Sections, each functional block is explained in detail.

3.2. The Network Measurement Block

In the network measurement block, the end-to-end packet transmission delay and the packet loss probability between a source and destination host are measured from the feedback information, which is notified by the destination host. The detailed algorithm is described as follows. First, when the source host encapsulates DV frames into UDP packets, a sequence number n and packet transmission time T_s are recorded in each packet header. At the destination host, an ACK (Acknowledgement) packet is sent to the source host when the destination host receives a packet. In the ACK packet, not only a sequence number n and a packet transmitting time T_s are recorded, but also, at the destination host, a packet reception time T_d is recorded.

At the source host, the average end-to-end packet transmission delay $\overline{\tau}$ and the instantaneous packet loss probability p during T are calculated from the sequence number n, the packet transmission time T_s, and the packet reception time T_d. In this paper, T is called a *sampling interval*. Since we cannot assume that the internal clocks of the source and destination hosts are accurately synchronized, $\overline{\tau}$ is not the exact end-to-end packet transmission delay. However, since the feedback control block controls a DV frame rate or a DV quantization scale based only on the variation in $\overline{\tau}$, synchronization of those internal clocks is not necessary.

To remove temporal variation in the end-to-end packet transmission delay or the packet loss probability, $\overline{\tau}$ and p are filtered to remove high-frequency components using the following low pass filters:

$$\begin{aligned} \overline{\tau}^* &\leftarrow \gamma_1 \overline{\tau}^* + (1-\gamma_1)\,\overline{\tau} \\ p^* &\leftarrow \gamma_2\, p^* + (1-\gamma_2)\, p \end{aligned}$$

Here, γ_1 and γ_2 are the cutoff frequencies ($0 < \gamma_1, \gamma_2 < 1$).

3.3. The Feedback Control Block

In the feedback control block, a quantization scale or a frame rate is re-calculated and notification is sent to the fil-

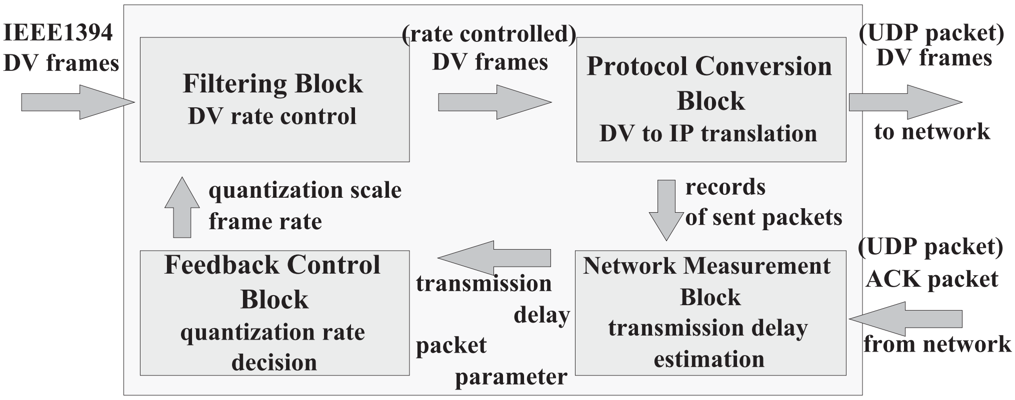

Figure 2. Block Diagram of Our DV Streaming System

tering block. As feedback information the values are calculated using the end-to-end packet transmission delay $\overline{\tau}^*$ and the packet loss probability p^*, which are calculated in the network measurement block.

More specifically, a quantization scale or the frame rate, x, is determined at every fixed sampling interval T, as the following formula:

$$x \quad \leftarrow \quad x + \alpha(p_0 - p^*) + \beta(\overline{\tau}_0 - \overline{\tau}^*) \qquad (1)$$

Here, p_0 is an target value for the packet loss probability and $\overline{\tau}_0$ is an target value of the end-to-end packet transmission delay. That is, the feedback control block controls a frame rate or a quantization scale to keep x constant. However, determing p_0 and $\overline{\tau}_0$ carefully is necessary. Although the appropriate value p_0 of packet loss probability depends on the packet loss probability in a vacant network, controlling the value to several percents to utilize a network effectively is necessary [16]. On the other hand, although the appropriate value $\overline{\tau}^*$ of the end-to-end packet transmission delay depends on the link capacity of a bottleneck link, controlling $\overline{\tau}^*$ to a value of the minimum end-to-end packet transmission delay plus the queuing delay of several packets is also necessary [17, 18]. However, if the values of p_0 and $\overline{\tau}_0$ are unsuitable, the control of the feedback control block becomes unstable. Therefore, those values must be determined carefully.

Moreover, in Eq(1), α is a feedback gain for packet loss probability and β is a feedback gain for the end-to-end packet transmission delay. If α and β are large, the transient behavior of system is improved, but the stability of the system is degraded. In contrast, if α and β are small, the system-wide transient behavior is degraded, but the system-wide stability improves. The method shown by Eq (1), is a sort of PID controls. Therefore, we can consider the Ziegler-Nichols technique applicable for the determination of a feedback gain [19].

3.4. The Filtering Block

In the filtering block, the packet transmission rate is adjusted by lowering the quality of the DV frames according to the quantization scale or the frame rate, x, which is determined by the feedback control block. By using one of the following techniques, the quality of DV streams can be adjusted.

- The DV data is discarded per block; i.e., the technique of simply discarding some DV blocks

  The advantage of this technique is that this technique is easy to implement and a transmission rate of DV stream can be determined flexibly. However, the quality of the DV stream becomes quite low, because block noises occur irregularly when the DV stream is played.

- The DV data is discarded per frame; i.e., the technique of discarding some DV frames

  The advantage of this technique is that discarding DV frames is easy to implement and the quality of the DV stream is better than that with the technique of discarding DV blocks. However, a disadvantage also exists; namely, a transmission rate of DV stream cannot be determined flexibly when compared with the tecnique of discarding DV blocks.

- The re-quantization of DV frames, i.e., the technique of re-quantizing DV frames

  The disadvantages of this technique are that re-quantization needs a very complicated implementation, and the load of a source host's CPU becomes very high. However, when compared with the technique of discarding DV frames, the quality of the DV stream can be kept high.

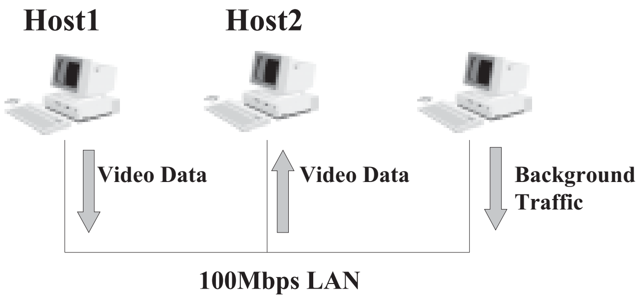

Figure 3. Network Model

Considering these advantages and disadvantages, we adopt both the technique of DV frame re-quantization and the technique of discarding DV frames as the techniques for lowering a transmission rate of DV stream. The combination of these methods enables the source host to adjust the transmission rate of DV stream flexibly. In addition, these methods do not spoil real-time application practicality.

3.5. The Protocol Conversion Block

In the protocol conversion block, DV frames that are received via the IEEE1394 interface, are encapsulated into UDP packets, and the UDP packets are sent to the destination host. In this paper, we use the DV→IP conversion function implemented in DVTS [9]. However, to record the packet transmitting time T_s, and the packet receiving time T_d in higher accuracy (μ second unit), a part of the RTP header is extended.

4. Implementation of the DV Transmission System

By modifying a part of DVTS version 0.9a15, the prototype of the real-time DV transmission system was implemented. In the implementation of this prototype, FreeBSD version 4.2-RELEASE was used as operating system, and ThinkPad s30 from IBM Corp. as a computer that sends or receives DV stream. ThinkPad s30 is equipped with the IEEE1394 interface. In the filtering block, although the techniques of discarding DV frames and re-quantization DV frames are adopted, only the function of discarding DV frames is implemented in the filtering block, so our this prototype is simple. In the followings, we show a result of out experiment in LAN environment obtained with our prototype. In the LAN environment, each host is connected via 100M Ethernet. The DV video was transmitted from the source host (Host1) to the destination host (Host2) (Fig. 3). The values of control parameters used in this experiment are shown in Table1. In Table 1, we set $\beta = 0$, that is,

Sampling interval	T	1,000,000 [μ sec]
The target of the rate of loss	$p0$	0.01
The feedback gain for the loss probability	α	100,000
The feedback gain for the end-to-end packet transmission delay	β	0

Table 1. Control Parameters

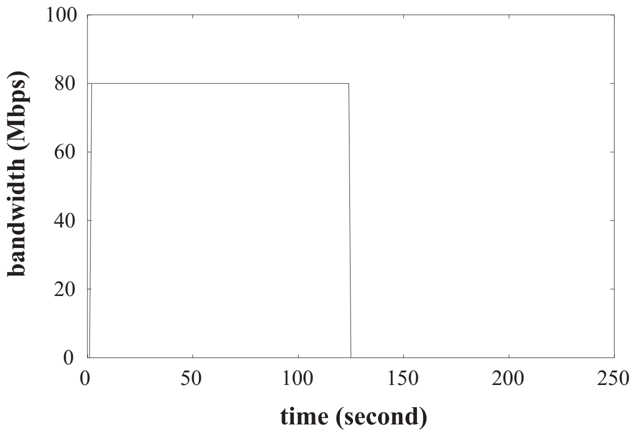

Figure 4. Amount of Background Traffic

we regarded only packet loss probability as the control target. Moreover, the UDP traffic of 80Mbps was generated as dummy traffic on the path, to verify how the dummy traffic influences the DV stream. The initial value of the frame rate was set to 30 [frame/s].

The results are shown in Figs. 4– 6. These figures show the amount of background traffic that was generated on the network, the transition of packet loss probability p measured in the network measurement block, and the transition of the frame rate x calculated in the feedback control block, respectively. We can find the followings: the packet loss probability is changed according to the transition of the background traffic, the frame rate is therefore changed. In this case, since the feedback gain α is set to a small value, $\alpha = 100,000$, the transition of the frame rate was slow.

5. Experiment

We have to verify effectiveness of our real-time video transmission system, which has the function of the feedback control, on a large-scale network where the end-to-end packet transmission delay is large. Thus, we performed an experiment on the network between UCSD (University of California San Diego) and Osaka University. Thanks to the JGN (Japan Gigabit Network), the APAN (Asia Pacific

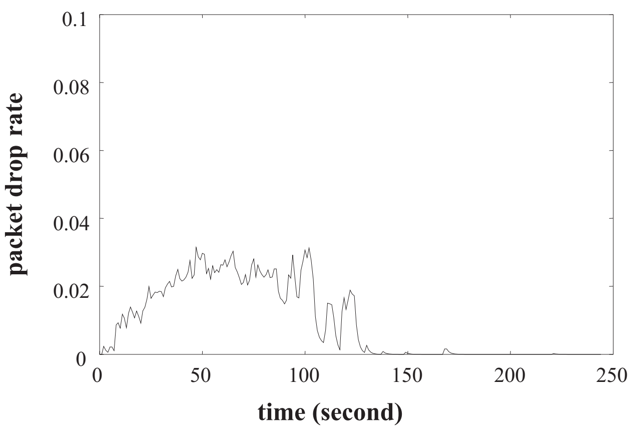

Figure 5. Packet Drop Rate calcurated by Network Measurement Block

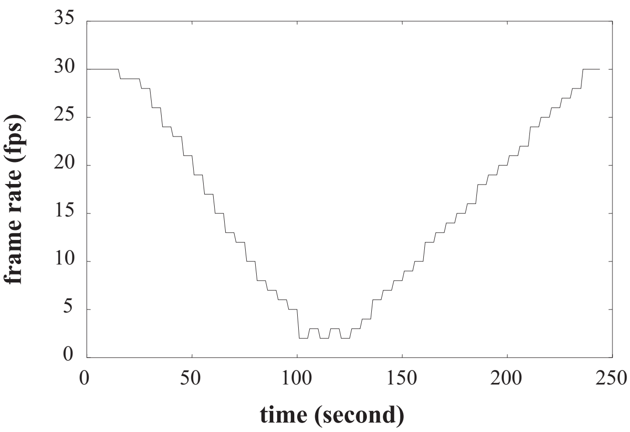

Figure 6. Frame Rate calcurated by Feedback Control Block

Advanced Network), and the vBNS (very high speed Backbone Network Service), we can perform the experiment on this network (Fig. 7). In this testbed, we perform the experiment with both IPv4 and IPv6.

It is possible to perform a distance conference between remote places with our real-time DV streaming system. Then, this system is used as a system for the researcher of SDSC (San Diego Supercomputer Center) who operates the ultra high voltage electric microscope, which is in Osaka University, by remote control (Fig. 8). The researcher uses the real-time DV streaming system to control the ultra high voltage electron microscope from the SDSC. Note that in this case, the detailed images of the high voltage electron microscope are transmitted with TCP using another channel.

In addition, we demonstrated our prototype at the SC2001

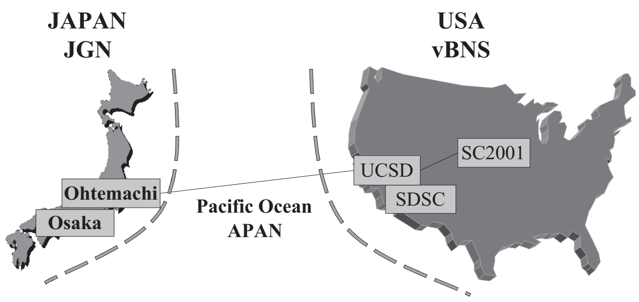

Figure 7. Experiment Network Environment

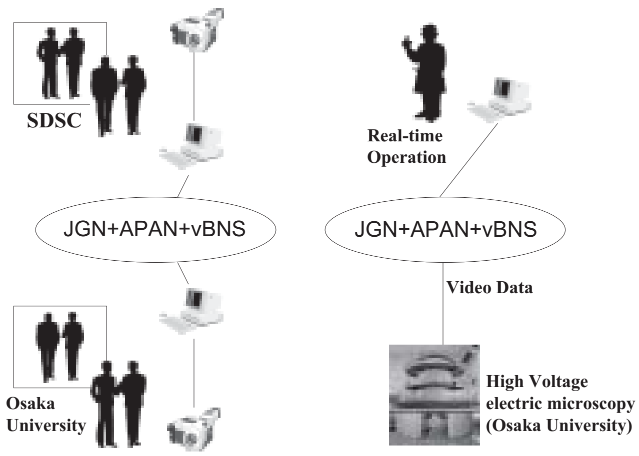

Figure 8. Remote Control of Microscopy

(Super Computing 2001), held in Denver, U.S. in November 2001 (Fig. 9). In this demonstration, we performed the DV transmission on the network between Denver and Osaka.

6. Summary

In this paper, we have discussed the development of a real-time DV streaming system, which dynamically adjusts the quality of DV frames using a feedback control mechanism according to the status of network congestion. In this system, the destination host notifies the source host of the feedback information. By using such information, the source host can dynamically adjust the quality of DV frames and can send the DV frames. Furthermore, the prototype for this system was implemented by modifying a part of the existing DVTS. Experimental results using our prototype were shown, and the experiment on a testbed between SDSC and Osaka University was also performed.

For future works, we are considering to implement a re-quantization function of DV frames. In addition, we are planning to use a mathematical technique for tuning control parameters in the feedback control block.

Figure 9. Demo in SC2001

Acknowledgement

This work was supported in part by the research project (JGN-P122561) of Telecommunications Advancement Organization of Japan. It is thanks to Dr. Mark Ellisman, Mr. Tom Hutton and other member of SDSC (San Diego Supercomputer Center) that we could advance our research. Mr. Keita Natsuyama (Nara Institute of Science and Technology) entirely supported our work, and Mr. Takuya Nomura (Superstation, Inc.) gave valuable advices to us.

References

[1] "Intertainer." available at `http://www.intertainer.com/`.

[2] "Real.com - Guide." available at `http://realguide.real.com/`.

[3] J.-C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the Internet," *ACM SIGCOMM Computer Communication Review*, pp. 4–15, Jan. 1998.

[4] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proceedings of IEEE INFOCOM '99*, Mar. 1999.

[5] M. Miyabayashi, N. Wakamiya, M. Murata, and H. Miyahara, "MPEG-TFRCP: video transfer with TCP-friendly rate control protocol," in *Proceedings of IEEE International Conference on Communications 2001*, vol. 1, pp. 137–141, June 2001.

[6] H. Radha and Y. Chen, "Fine-granular-scalable video for packet networks," in *Proceedings of Packet Video '99*, Apr. 1999.

[7] R. Ramjee, J.Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide area networks," in *Proceedings of IEEE INFOCOM '94*, pp. 680–688, Apr. 1994.

[8] S. B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms," *ACM/Springer Multimedia Systems*, vol. 5, pp. 17–28, Jan. 1998.

[9] K. Sugiura, A. Ogawa, O. Nakamura, and J. Murai, "Design and implementation of DV based video over RTP," in *Packet Video Workshop 2000* , May 2000.

[10] "RealSystem RealServer Best Practices." available at `http://service.real.com/help/library/guides/server8/deploy/htmfiles/notice.htm`.

[11] "Windows Media Format." available at `http://www.microsoft.com/windows/windowsmedia/wm7/format.asp`.

[12] "Inside QuickTime." available at `http://developer.apple.com/techpubs/quicktime/qtdevdocs/PDF/insideqt_intmov.pdf`.

[13] "MediaBase." available at `http://www.sgi.co.jp/software/mediabase/`.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *Request for Comments (RFC)* 2475, December 1998.

[15] "ISO/IEC 13818-1 international standard MPEG-2 systems." available at `http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=31537`.

[16] R. Jain, "Congestion control in computer networks: issues and trends," *IEEE Network Magazine*, pp. 24–30, May 1990.

[17] R. Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," *ACM Computer Communication Review*, vol. 19, pp. 56–71, Oct. 1989.

[18] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of ACM SIGCOMM '94*, pp. 24–35, Oct. 1994.

[19] J. Aweya, M. Ouellette, and D. Y. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, pp. 203–235, 2001.