Designing the ON-OFF CBR Transmission Schedule for Jitter-Free VBR Media Playback in Real-Time Networks

Ray-I Chang, Meng Chang Chen, Jan-Ming Ho, Ming-Tat Ko
Institute of Information Science, Academia Sinica, Taiwan
{william,mcc,hoho,mtko}@iis.sinica.edu.tw

Abstract

This paper addresses the problem of designing the CBR transmission schedule (CTS) for jitter-free media playback. Comparing to the VBR transmission schedule (VTS), CTS is known to have advantages in low overhead and complexity for managing real-time communications. However, media data are usually compressed as VBR streams with noticeable bit-rate variabilities. Conventional CTS approaches usually require large buffer size or work-ahead to resolve this problem. In this paper, a linear-time CTS algorithm with the minimum system resources is proposed to transmit VBR media for jitter-free playback. Different from the conventional approaches, the obtained CTS is with the ON-OFF switch behavior. As the ATM transport mechanism used in current high-speed networks, it transmits fixed size cells in the ON period and zero cells in the OFF period. In 1993, Sohraby has shown that the general ON-OFF resource has a simple multiplexer queue length distribution function. It is very easy to do the call admission control (CAC) for real-time networks. In this paper, empirical studies to different VBR media streams are explored. The obtained results demonstrate that the proposed approach is effective and efficient.

1 Introduction

A media stream is usually composed of a set of data frames with a time-critical display order. Thus, the capability of handling media streams such that the playback is in real-time without jitter is an essential requirement to a multimedia system. Fig. 1 shows a basic multimedia system model and its operation flow. The server retrieves VBR video data from a real-time storage subsystem [2] and buffer the data in the server memory. According to a transmission schedule, the buffered video data is packetized and transmitted via the real-time network subsystem. Once the data arrives at the client site, it is buffered in the client memory. The client fetches media data from buffer, decodes and displays the media periodically. If the real-time constraint is not met and data in the client buffer is less than the playback requirement, client has to stop displaying the frame. The discontinuity phenomena is called *jitter*. It is an important QoS (quality-of-service) measurement of the system performance. In order to guarantee QoS, sufficient resources must be reserved to each request and require high cost. There is a tradeoff between the QoS and the system cost.

VBR media streams are generally with noticeable bit-rate variability [7]. Although a VBR transmission schedule (VTS) may fit the VBR stream well, high overhead and complexity of network management are the drawbacks. Besides, as the network bandwidth is limited and shared, real-time QoS may not be guaranteed for VTS [8]. To resolve these drawbacks, the CBR transmission schedule (CTS) is

---

*The research described in this paper was partially supported by NSC under grant NSC84-2221-E-001-005.

0-8186-8073-3/97 $10.00 © 1997 IEEE

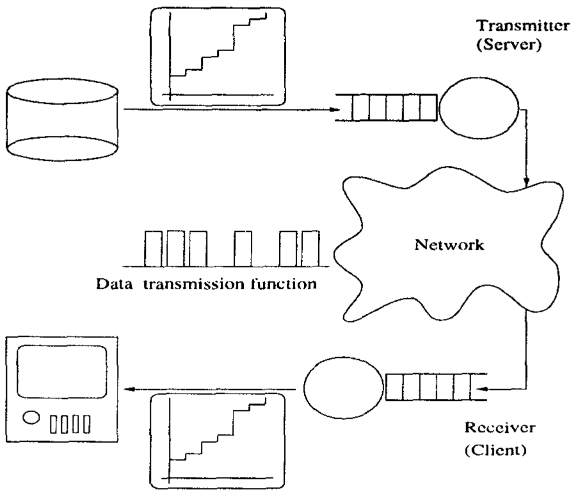

Figure 1. The basic multimedia system with real-time storage and network subsystems.

designed. Conventional CTS methods continuously transport fixed size cells by a fixed rate [6]. They are shown to be good for network management to guarantee real-time QoS. However, the required buffer size is usually large with a long work-ahead time [13]. The over-reservation of resources usually causes system's under-utilization and rejects the new requests needlessly. These VBR media streams lead a big challenge to design a good CBR transmission schedule [9]. In this paper, we design a new CTS with the ON-OFF behavior to support jitter-free VBR stream playback.

An ON-OFF CTS (O^2CTS), as a CTS, is a CBR service model which transmits the media stream by a fixed rate. However, the transmission is turned to OFF if the sufficient data for jitter-free playback is already buffered in the client. As the buffered data size is minimized (just sufficient for jitter-free playback in the next time period), the required buffer size is minimized. Besides, the work-ahead is also minimized. In this paper, an algorithm with the theoretic proof is presented to determine the minimum buffer size and work-ahead to transmit a VBR media stream. Notably, the available system parameters are varied from machine to machine. It is necessary to give client and server the flexibility to negotiate the allocated resources. However, the relationships between these parameters are not trivial. Although a large buffer can conceal jitter, the side effect is a long work-ahead. We derive the chart of the required buffer size and network utilization verse the given transmission rate. The chart can be generated off-line such that client and server can use it to negotiate an agreed transmission context (such as network bandwidth, buffer size and work-ahead).

The remainder of this paper is organized as follows. Some previous works are reviewed in Section 2. Then, the proposed approach is presented in Section 3. We introduce the general concepts and proposes the detail algorithm for data transmission. As shown by Sohraby [18], the related multiplexer queue length distribution is a simple function of the source peak rate and the first two moments of their ON-OFF periods. The call admission control (CAC) becomes very easy. Negotiations for different transmission contexts are given in Section 3. Empirical studies of the proposed approach to different video clips are shown in Section 4. In the final section, the conclusion is discussed.

2 Related Work

The problem to reduce the rate variability of a VBR media stream contains several dimensions [17]. For a live media, [14] presents a per-stream buffering approach to reduce the peak transmission rate. When network congestion is detected, [16] delay the video encoder to reduce the congestion. To smooth the traffic, a heuristic method is proposed [12] to predict future frame sizes. Comparing with live media, stored media can work on a broader time scale to achieve better improvement. *Tenet* group has done an extensive work in this problem [5, 19, 20, 10]. However, as [4, 17, 8], these approaches renegotiate different rates to accommodate the multiple-time-scale variability of VBR video streams. They have the same drawbacks of VTS.

Currently, CBR transmission services are widely used in high-speed networks such as ATM, cable networks, ISDN and telephone, to simplify of network management. In [13], a simple CTS is designed to optimize the memory buffer size related to the given work-ahead. As the data cells are continuously transmitted by a fixed rate, the required buffer size is very large. In this paper, a new CTS with the ON-OFF switch behavior [18] is proposed. Different from the conventional approaches, the transmission state is OFF if the sufficient data for jitter-free playback is already buffered. Thus, the required buffer size is reduced to a minimum. The relations between the proposed O^2CTS and the obtained performance parameters (including buffer sizes, network utilization rate, and work-ahead) are also studied in this paper.

3 ON-OFF CBR Transmission Model

In this section, we introduce the general concepts of O^2CTS. A lazy-receiving scheme is then proposed to construct an O^2CTS for the given VBR media stream. Some important properties of the obtained work-ahead and buffer size are exploited. We have shown that the proposed scheme can minimize both the obtained work-head and buffer size in linear time. In this paper, we assume that the end-to-end transmission behavior is continuous. In another words, the transmission schedule T is a continuous function ($T(t^-) = T(t) = T(t^+)$). The network delay is assumed to be zero or be controlling within a very small range [20]. Given the maximum packet size p and the minimum packet time s, it can be easily extended to a discrete transmission schedule by some modifications ($T(t+s) \leq T(t) + p$).

3.1 General Concepts

A media stream can be viewed as a *finite* sequence of frames, $f = f_1 f_2 \dots f_n$, where n is the number of frames and f_i is the ith frame. For the simplification, in this paper, f_i also denotes the size of frame and f denotes the size of video. Assume that the server playbacks a frame per unit time. The cumulative frame size at time i is defined as $F(i) = F(i-1) + f_i$, where $F(0) = 0$ and $F(i) = f$ if $i \geq n$. Given a VBR media stream f as shown in Fig. 2(a), F is a nondecreasing step function (see Fig. 2(b)). Define the data consuming function $FR(t)$ as the cumulative data size that has been played up to time t. Under jitter-free playback, we have $FR(t) = F(t)$ for all t.

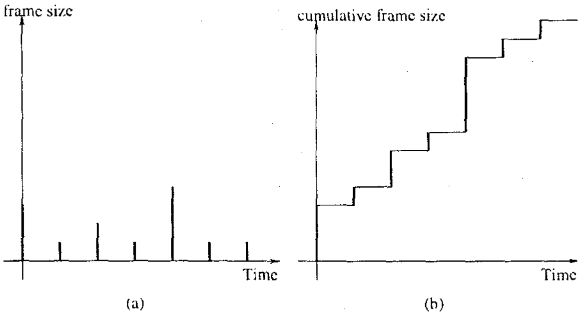

Figure 2. (a) The frame size of a VBR media. (b) The cumulative frame size.

Define $T(t)$ as the cumulative data size that transmitted by a transmission schedule up to time t. It can be used to characterize a transmission schedule. A simple example of O^2CTS is illustrated in Fig. 3. It is a piecewise linear nondecreasing function. For the given rate r, each linear piece is either of slope 0 or r. Note that, if $T(t^-) \geq F(t)$ for all t, the transmission schedule T is said to be jitter-free for playing f . The difference $(T(t) - F(t^-)) \leq 0$ is the client buffer size required to store the transmitted data that has not be played at time t. The client buffer size required by using the transmission schedule $T(t)$ is the maximum value among these differences. As shown in Fig. 2 and Fig. 3, we have $F(t^-) = F(t-1)$ and $T(t^-) = T(t)$. The required client buffer size is $\max\{T(t) - F(t-1)$ — for all $t\}$. Assume that w is the starting time of the transmission schedule where $T(w) > 0$ and $T(x) = 0$ for all $x < w$. The playback work-ahead of $T(t)$ is $1 - w$.

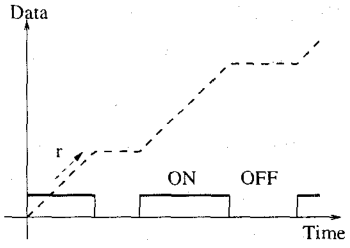

Figure 3. The ON-OFF behavior of an ON-OFF CBR transmission schedule.

Let the data serving function $FX(t)$ be the the cumulative data size that has been retrieved from the disk to the server buffer up to time t. We should also guarantee $FX(t^-) \geq T(t)$ for the jitter-free playback. In this paper, we focus on designing a good transmission schedule for jitter-free playback in real-time networks. A real-time disk scheduling algorithm such as [15, 3] is assumed to be perfectly applied on the server and the given server buffer is sufficient.

3.2 Lazy Receiving Scheme

The concept of the lazy receiving scheme is to send the media stream as late as possible if no jitter may occur. Let $T_L(t)$ be the O^2CTS of the lazy receiving scheme. Given the peak transmission rate r, the curve of $T_L(t)$ can be constructed piece by piece as follows. Consider a semi-line starting at a point on the y-axis and of slope r which is above the function curve of $F(t)$. Move the semi-line right until it touches the function curve of $F(t)$ at some a point $(t_1, F(t_1))$. Assume that the resulting semi-line starts at the point $(w, 0)$. It means that the transmission should start no later than w, otherwise, the playback would have a jitter at time t_1. Thus, the segment from (w, 0) to $(t_1, F(t_1))$ forms the first ON piece of the transmission schedule curve. Then move the remaining semi-line starting at $(t_1, F(t_1))$ right further until it touches $F(t)$ again at some a point $(t_2, F(t_2))$. The segment from the new starting point to $(t_2, F(t_2))$ forms the second ON piece of the transmission schedule curve. Note that the move trace of the starting point is of slope 0. It forms the OFF piece of the transmission schedule curve. Continue the moving process, we can obtain the whole transmission schedule curve of the lazy receiving scheme. An example of the construction process of $T_L(t)$ is illustrated in Fig. 4. We can also construct this O^2CTS by a backward manner as follows.

$$
\begin{aligned}
T_L(n) &= F(n) \\
T_L(t) &= \max\{F(t), T_L(t+1) - r\}, if(t < n)
\end{aligned}
$$

The schedule and the required buffer size ($\max\{T_L(t) - F(t-1)\}$) can be computed in linear time. The required work-ahead is $w = T_L(1)/r$. It defines the time since viewer issues play command till the first frame displayed.

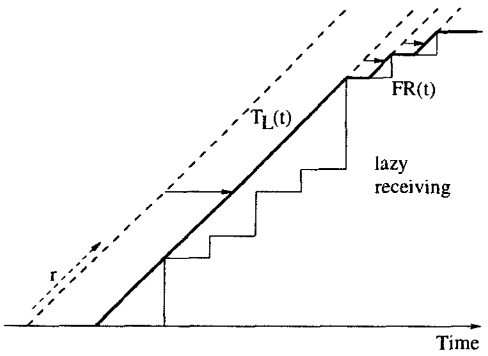

Figure 4. A simple example to illustrate the construction process of $T_L(t)$ by the lazy receiving scheme.

From the construction of the lazy-receiving transmission schedule, it is easy to know that $T_L(t)$ is the lower envelope for all the legal transmission schedules with the peak transmission rate r. By this observation, the following two theorems can be easily proved.

Theorem 1 *Given the media stream f and the peak transmission rate r. The client buffer size of the lazy receiving transmission schedule T_L is minimum among all the jitter-free transmission schedules.*

Proof: From the lazy receiving scheme, $T_L(t)$ is the minimum cumulative data size that should be transmitted up to time t for jitter-free playback. For any jitter-free transmission schedule T, we have $(T(t) - F(t-1)) \geq (T_L(t) - F(t-1))$. Thus, we conclude that the client buffer size of the lazy receiving transmission schedule T_L is minimum. Q.E.D.

Theorem 2 *Given the media stream f and the peak transmission rate r. The work-ahead time of the lazy receiving transmission schedule T_L is minimum among all the jitter-free transmission schedules.*

Proof: From the definition, $T_L(1)$ is the minimum cumulative data size that should be transmitted up to time 1 for jitter-free playback. Thus, we conclude that the required work-ahead $w = T_L(1)/r$ is minimum. Q.E.D.

4 Transmission Contexts

In the previous section, we have presented the lazy receiving scheme to construct an O^2CTS to achieve the minimum work-ahead and client buffer size. By exploiting the properties of the transmission schedules, we can find the relation of the minimum buffer sizes to the related transmission rate.

4.1 Parameters

In this subsection, a large KTV clip with over 6000 frames is applied to demonstrate the parameters used in this paper. In Fig. 5(a), the VBR frame sizes which ranges from 32 KB to 1 KB are shown. Define the worst case network utilization by the following equation.

$$Util = \frac{100 * Total\ Video\ Size}{(Transmission\ Rate\ *\ Duration)} \quad (1)$$

The transmission schedule has $Util\%$ of time is in the ON state and (1-$Util$)% in the OFF state. Fig. 5(b) shows the relation between the obtained buffer size and network utilization versus the given transmission rate. It is called *buffer-utilization-rate chart.* A native algorithm takes the $O(n^3)$ time complexity to compute the buffer-utilization-rate chart. In [11], an algorithm of complexity $O(n \log n)$ is presented.

Initially, the required minimum buffer size decreases linearly by a constant slope as the transmission rate increases. With the increase of transmission rate, the decreasing slope of the buffer size hops to a lower one and the required buffer size decreases linearly by this new slope as the transmission rate increases. When the transmission rate hits the turning point around 1.34 Mbps, the value of minimum buffer size becomes flat. Finally, the value maintains constant value, even the transmission rate keeps increasing. This constant value is just the size of the largest frame of the video stream. By considering the worse case network utilization, the utilization is 100% when the transmission rate is below 1.33 Mbps. It means that the transmission rate is below video display rate. Hence the client has to a large buffer for preloading frames. After passing the turning point, the utilization rate decreases proportional to the inverse of the transmission rate. Note that, in this paper, we consider only the worst case network utilization. As described in [18], it is only dependent on the given rate and the first moments of ON-OFF periods. The network utilization would be highly improved if the temporal multiplexing mechanism is applied.

4.2 Negotiations of Transmission Context

When a client request to view a movie, it sends its preference, available buffer size, affordable transmission rate, and endurable work-ahead to the server. The server looks up the buffer-utilization-rate chart, following clients instructions, to decide the transmission context. If the client is memory-conscious, the server will determine a transmission context with as small buffer as possible, while the transmission rate and work-ahead are under the client constraints. If the client is transmission-conscious, the server will optimize the transmission rate with the constraints of buffer size and work-ahead.

For example, a viewer wants to watch the MTV clip shown in Fig. 5(a). He can afford 100KB memory buffer in client machine and 1.5Mbps transmission rate for network bandwidth. Check the buffer-utilization-rate chart shown in Fig. 5(b). If the viewer is memory-conscious, the server may determine a transmission context of 60 KB memory and 1.5Mbps transmission rate. On the other hand, if the viewer is communication-conscious, the transmission context is of 100KB memory and 1.4 Mbps transmission rate. If the viewer can only afford 200KB memory and 1.3 Mbps transmission rate, the server will reject the request. It can be found that both the admission and transmission policy are very efficient in this design. The server checks the context against its available resources, and decide to admit or

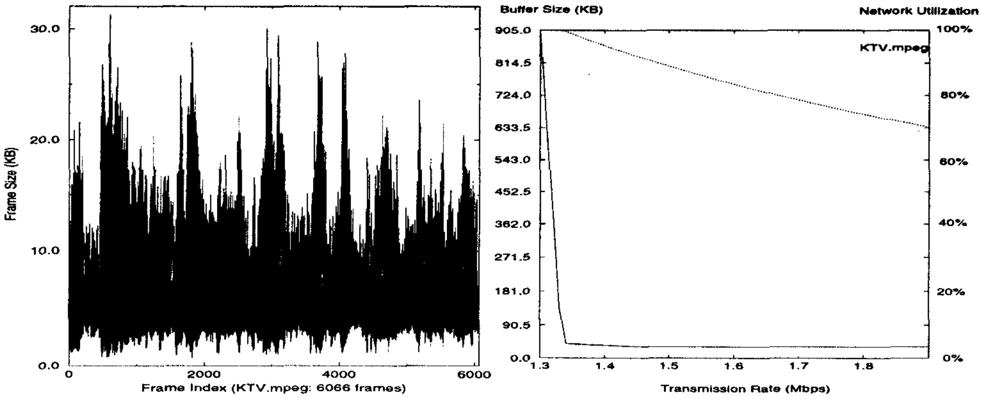

Figure 5. A KTV video clip: (a) Frame sizes. (b) Buffer and utilization versus transmission rates.

not. Based on the design, the session set-up protocol can be as simple as a request-reply.

4.3 Empirical Study

The proposed approach is examined by several MPEG movie clips downloaded from the movie clip archive [1]. Their statistics (the frame number, the maximum frame size, and the average frame size) are listed in Table 1. Each movie clip represents a different type of network traffic. In Fig. 6, the movie clip is an animation that the frames are almost same size. The second movie clip is a singer singing that the frame sizes change regularly as shown in Fig. 7. The last one is an action movie clip that the frame sizes change smoothly until the scene changes incurring an abrupt size change. It is shown in Fig. 8.

Video	Frame No.	Max Size	Avg. Size
KTV	6066	32.0 KB	8.00 KB
Bird	61	3.38 KB	3.30 KB
EricC	151	5.23 KB	2.29 KB
Indiana	620	7.69 KB	3.52 KB

Table 1. Statistics of Movie Clips.

For each movie clip, we apply the proposed algorithm and find the buffer-utilization-rate chart (as shown in the middle of each figure set). Then, we select three transmission contexts: one is with average transmission rate, the other is with the maximum transmission rate, and the last one is at the turning point (called *turn-pt*) shown in the chart. The third figure of each clip depicts the amount of data in the client buffer during movie playback by these three transmission contexts. We denote r as the transmission rate, b as the buffer size, and u as the worst case network utilization. The obtained results hint that the turning point is a good communication context candidate. It usually achieves small buffer size with low transmission rate.

5 Conclusion

In this paper, we introduce the ON-OFF CBR transmission schedule for jitter-free playback of VBR media streams in real-time networks. Unlike conventional approaches, our approach determines the rate before transmitting data and the rate is fixed during the session. Thus, renegotiation within the session and the overhead are no longer. This fixed rate transmission schedule is particularly effective for real-time applications in future high-speed networks such as ATM. We have also presented a lazy receiving scheme to find the minimum resource transmission schedule. The proposed approach can be used in pre-processing a VBR media stream to decide the minimum resources for jitter-free playback. Then, depending on the preferences of client and server, minimum buffer size or minimum transmission rate or their combinations can be decided. In general, the transmission context defined by the turning point of buffer-rate chart is a good recommendation. It is close to the minimum buffer size with a low transmission rate. Our approach is

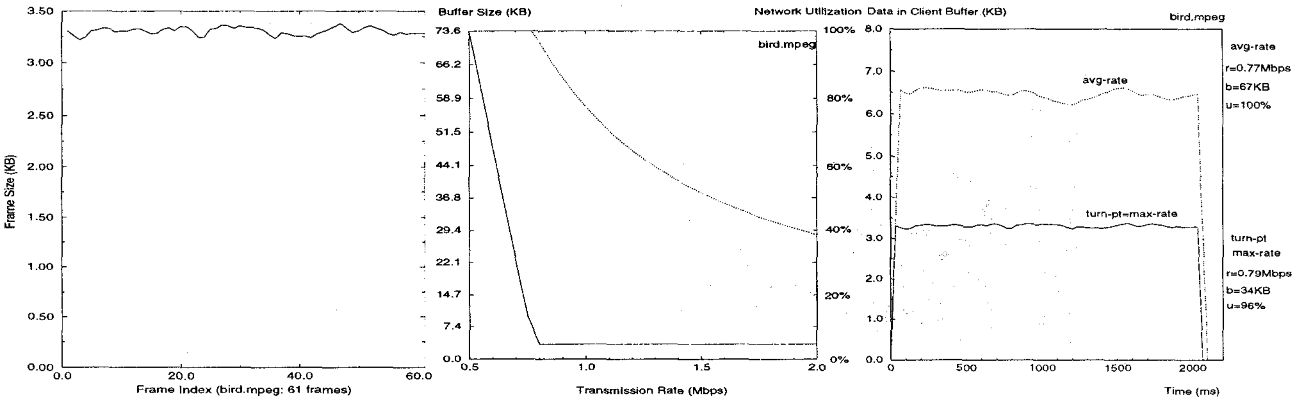

Figure 6. Statistics of Video Clip: Animation.

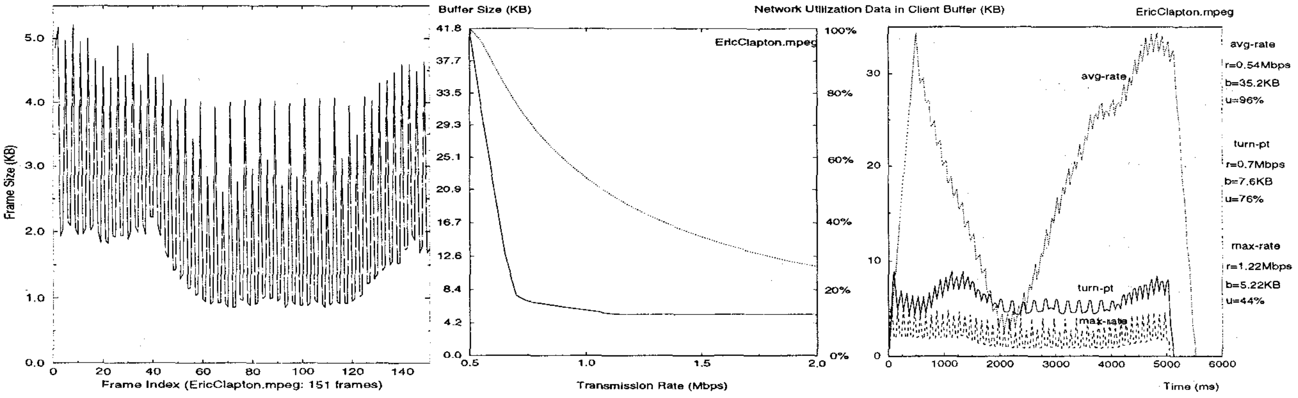

Figure 7. Statistics of Video Clip: Singer.

efficient and shows great flexibility to allow various clients to set up their best transmission contexts.

References

[1] MPEG Movie Archive. URL http::/www.eeb.ele.tue.nl/mpeg/.

[2] E. Chang and A. Zakhor. Scalable video data placement on parallel disk arrays. In *IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, February 1994.

[3] R. I. Chang, W. K. Shih, and R. C. Chang. Hybrid approaches with local/global seek-optimizing schemes for multimedia real-time disk scheduling. Technical report, submitted for publication, 1996.

[4] W. Feng and S. Sechrest. Smoothing and buffering for delivery of prerecorded compressed video. In *IS&T/SPIE Multimedia Computing and Networking*, February 1995.

[5] D. Ferrari. Client requirements for real-time communication services. *IEEE Network Magazine*, Novemeber 1990.

[6] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, April 1990.

[7] M. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM*, August 1994.

[8] M. Grossglauser, S. Keshav, and D. Tse. RCBR: a simple and efficient service for multiple time-scale traffic. In *Proc. ACM SIGCOMM*, August 1995.

[9] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Proc. ACM SIGMETRICS*, May 1995.

[10] E. W. Knightly and H. Zhang. Traffic characterization and switch utilization using a deterministic bounding interval dependent traffic model. In *IEEE IMFOCOM*, 1995.

[11] Ming-Tat Ko and Jan-Ming Ho. An O($n\log n$) algorithm for finding the relation of minimum buffer size and transmission rate of an ON-OFF CBR function. Technical report, private communication.

[12] S. S. Lam, S. Chow, and D. K. Y. Yau. An algorithm for lossless smoothing of MPEG video. In *Proc. ACM SIGCOMM*, 1994.

[13] J. M. McManus and K. W. Ross. Video on demand over ATM: Constant-rate transmission and transport. In *Proc. IEEE INFOCOM*, March 1996.

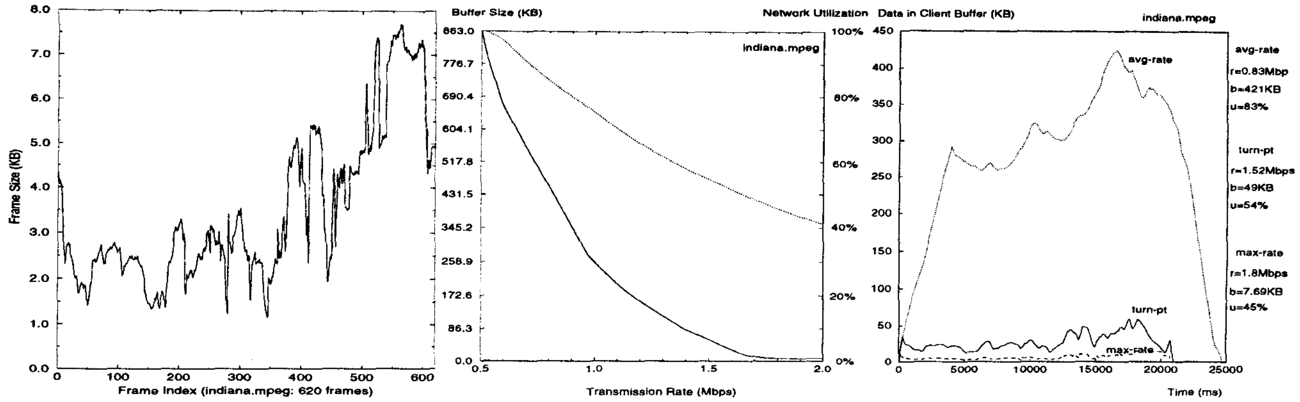

Figure 8. Statistics of Video Clip: Movie.

[14] T. Ott, T. V. Lakshman, and A. Tabatabai. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *Proc. IEEE INFOCOM*, 1992.

[15] A. L. N. Reddy and J. Wyllie. Disk scheduling in a multimedia I/O system. In *ACM Multimedia Conference*, pages 225–233, 1993.

[16] A.R. Reibman and A.W. Berger. Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking*, June 1995.

[17] J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. Towsley. Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing. In *Proc. ACM SIGMETRICS*, 1996.

[18] K. Sohraby. On the theory of general ON-OFF source with applications in high-speed networks. In *IEEE INFOCOM*, pages 401–410, 1993.

[19] H. Zhang and D. Ferrari. Improving utilization for deterministic service in multimedia communication. In *Proc. IEEE International Conference on Multimedia Computing and Systems*, 1994.

[20] H. Zhang and E. W. Knightly. A new approach to support delay-sensitive VBR video in packet-switched networks. In *Proc. 5th Workshop on Network and Operating Systems Support for Digital Audio and Video*, 1995.