Two-Pass Maximum Average Smoothing Algorithm for VBR Video in ATM Environment

Joo-young Son*

Abstract

Variable-bit-rate (VBR) video encoding has been used to achieve maximum data compression ratio and to ensure constant picture quality at the receivers. Traffic smoothing has been suggested to reduce the fluctuation of data at the source so that the traffic passes the network without significant delay perturbation. Smoothing consists of moving excess data from a bursty picture frame to frames preceding it, so that the frame size is slowly varying from frame to frame. Previous works on smoothing have focused primarily on the degree of smoothness(peak-to-average ratio). However, in terms of overall performance, the number of simultaneous sessions should be kept as large as possible under given network bandwidth and receiver buffer size. This paper proposes a new smoothing algorithm for stored VBR video data to achieve this goal. This scheme also reduces the frequency of rate change so that the bandwidth renegotiation between server and client at the ATM layer happens less frequently. As this is applied only to stored video data, we assume that all frames are available for smoothing at the server at any instant.

The proposed algorithm, two-pass maximum average (TPMA) smoothing algorithm, is based on the concept of maximum average. At the first pass, the running average (average size of frames from the start to the current frame) is computed for every frame. The whole video session is divided into regions. The frames in each region are smoothed by its maximum average. It is noted that the value of maximum averages decreases as we go from the first region to the last region. This first pass aims to reduce at maximum the bandwidth- time product necessary for the video data transmission.

The second pass consists of reducing the number of regions so that the frequency of rate

* Division of Automation and Information Engineering Korea Maritime University

change is minimized. Regions with small number of frames are removed, and poured into their previous regions. In ATM networks, time needed for bandwidth renegotiation can not be ignored, and regions shorter than this time should be removed.

Experiment studies with VBR video data showed that TPMA smoothing algorithm performs very well compared to previously available smoothing schemes, in the number of simultaneous video sessions, and the number of rate changes after smoothing.

1. Introduction

Variable-bit-rate (VBR) video encoding has been used to achieve maximum data compression ratio and to ensure constant picture quality at the receivers. MPEG (Moving Picture Expert Group) video compression[11] is the most widely accepted standard for encoding high quality pictures. In MPEG, to improve the compression efficiency, three different encodings are applied: I for intraframe encoding with limited compression gain, P for predictive coding and B for bidirectional coding that produce smaller frames than I. For each I, a number of B and P frames are associated forming so-called group-of-pictures (GOP). Even though MPEG VBR encoding ensures constant picture quality at the receiving side, the frame size varies largely from I to B and P frames.

ATM (Asynchronous Transfer Mode) network is considered as the most suitable transport media for video due to its inherent capability to handle efficiently asynchronous and bursty traffic. However even ATM can not guarantee timely delivery of video data when it is largely fluctuating as in the case of raw VBR video where peak-to-average ratio is typically greater than ten. For example, in Figure 1, the frame sizes, f_k, of a VBR MPEG-encoded video stream entitled Red's Nightmare are depicted. The max-to-min ratio is approximately 96:1, and the peak-to-average ratio(PAR) is 9.3.

Traffic smoothing has been suggested to reduce the fluctuation of data at the source so that the traffic passes the network without significant delay perturbation[4]. Smoothing consists of moving excess data from a bursty picture frame to frames preceding it, so that the frame size is slowly varying from frame to frame. Smoothing techniques are evaluated by the complexity of smoothing at the video server and de-smoothing at the receiver, degree of smoothness (i.e. peak-to-average ratio), efficiency of resource utilization, required buffer size at receiver, number of simultaneous video sessions for

given network bandwidth, and frequency of traffic rate change after smoothing. Previous works on smoothing have focused primarily on the degree of smoothness.

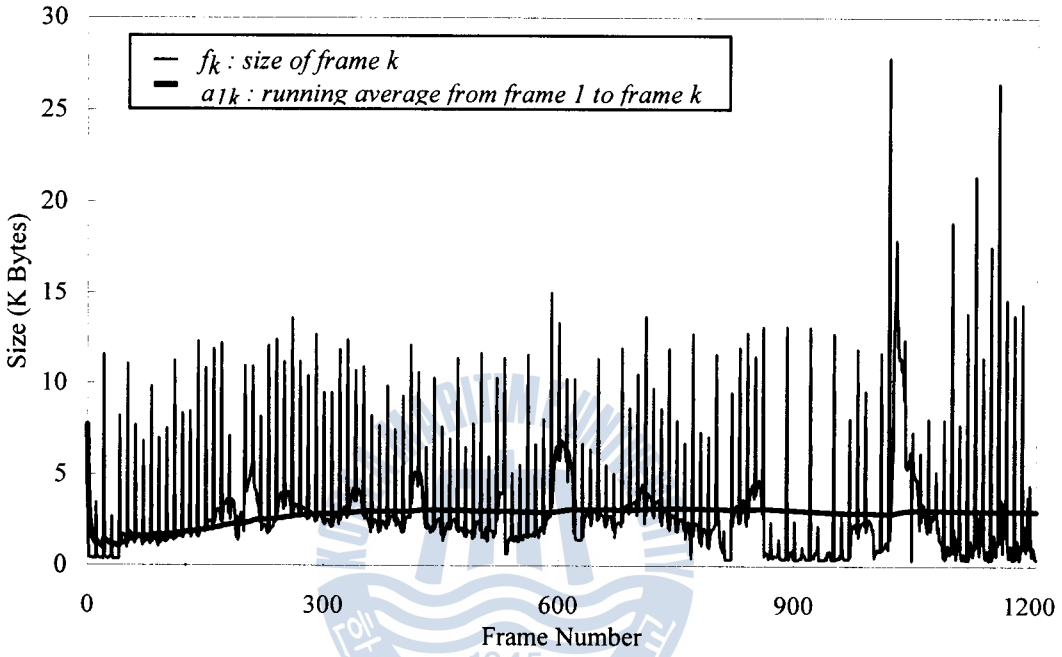


Figure 1. Frame sizes(f_k) and running average sizes(a_{1k}) of RedsNightmare.mpg

However, in terms of overall performance, the number of simultaneous sessions should be kept as large as possible under given network bandwidth and receiver buffer size. This paper proposes a new smoothing algorithm for stored VBR video data , that achieves this goal. This scheme also reduces the frequency of rate change so that the bandwidth renegotiation between server and client at the ATM layer happens less frequently. As this is applied only to stored video data, we assume that all frames are available for smoothing at the server at any instant.

Previous studies on video traffic smoothing are primarily for online real-time encoding and delivery of video such as video conferencing and television broadcasting. In this environment, since one cannot introduce long delay between sender and receiver, only limited number of frames are available for smoothing, resulting in poor smoothing performance. In this paper, we propose a new smoothing algorithm for stored video data.

In this case, as the whole data is available in advance for smoothing, maximum smoothing can be easily achieved. The proposed algorithm, two-pass maximum average (TPMA) smoothing algorithm, is based on the concept of maximum average. TPMA reduces the bandwidth-time product necessary for the video data transmission, and the frequency of transmission rate changes. Therefore TPMA ensures greater number of simultaneous video sessions at the server, and more efficient use of ATM networks, as bandwidth renegotiations do not occur frequently.

This paper is organized as follows: previous works are surveyed in section 2. In section 3, the TPMA scheme is presented in detail. The buffer requirement at the receiver is analyzed in section 4. Section 5 demonstrates the performance of TPMA, and the buffer requirements at the receiver with two typical video data examples are described. Concluding remarks are given in section 6.

2. Previous Works

To reduce the data fluctuation at the sender, several schemes have been developed : peak-rate enforcement[2], frame prioritization[3, 8], and running smoothing[9, 10]. In peak-rate enforcement, when the output rate exceeds a pre-determined threshold, the quantization scale in the video encoder is increased. This, however, causes the inevitable degradation of the video quality.

The frame prioritization scheme separates components of a video stream according to relative order of importance[7]. The priority coder is introduced to assign a particular priority to each component. This scheme is based on the assumption that the influence of the lower prioritized component losses on video quality might be much less than that of higher priority one. And the low-priority components are dropped when the data rate becomes too high. This scheme also cannot deliver constant quality video.

Previous video frame smoothing schemes have been devised both for online real-time video transmission and for stored video retrieval where the time difference between the encoding and playback of streams is not short [5, 6, 7, 9]. However, they did not consider buffer requirement and frequency of rate changes after smoothing.

The TPMA smoothing scheme proposed in this paper smoothes the data fluctuation as much as possible, and guarantees the timely delivery and the playback continuity. Also the

exact amount of buffer necessary for continuous playout can be easily calculated in advance.

3. Two-pass maximum average (TPMA) smoothing scheme

We assume that all frames are available for smoothing at the server at any instant. The TPMA scheme is designed to reduce the bit rate fluctuation of VBR video streams by large amount so that the resource utilization can be kept high. The buffering burden due to the smoothing can be shifted from sources to receivers, which results in large increase in the number of simultaneous video sessions.

At the first pass, the running average (average from the start to the current frame) is computed for every frame. And then, the whole video session is divided into regions. The first region is from the start frame to the frame whose running average is of maximum value for all frames. Then the running average is recomputed from the frame right after the first region until the last frame. The second region is now identified as from the frame after the first region to the frame with maximum running average. This process is repeated until the last frame belongs to one region. It is noted that the value of maximum averages decreases as we go from the first region to the last region. This first pass aims to reduce at maximum the bandwidth-time product necessary for the video data transmission.

The second pass consists of reducing the number of regions so that the frequency of rate change is minimized. Regions with small number of frames are removed, and poured into their previous regions. In ATM networks, time needed for bandwidth renegotiation can not be ignored, and regions shorter than this time should be removed.

3.1 The First Pass of the TPMA Smoothing Scheme

Assume that the total number of frames in a VBR video stream be N, and frame size is denoted as f_k for frame k. The total amount of video data F_{1N} is

$$F_{1N} = \sum_{i=1}^{N} f_i \tag{1}$$

The total average a_{1N} is

$$a_{1N} = \frac{F_{1N}}{N} \tag{2}$$

For each frame k ($1 \le k \le N$), the running average a_{1k} of the frames between frame 1

and k is defined as

$$a_{1k} = \frac{F_{1k}}{k} \tag{3}$$

In Figure 1, a_{1k} for RedsNightmare.mpg are also depicted. a_{1k} for small k is larger than those of large k, and, as k approaches N, it converges to the average frame size of the whole stream. It is noted that at the first frame, $a_{11}=f_1$. f_1 is always encoded in I-mode for all MPEG video streams. The frame size of I frame is usually much larger than those of the P- or B-frames.

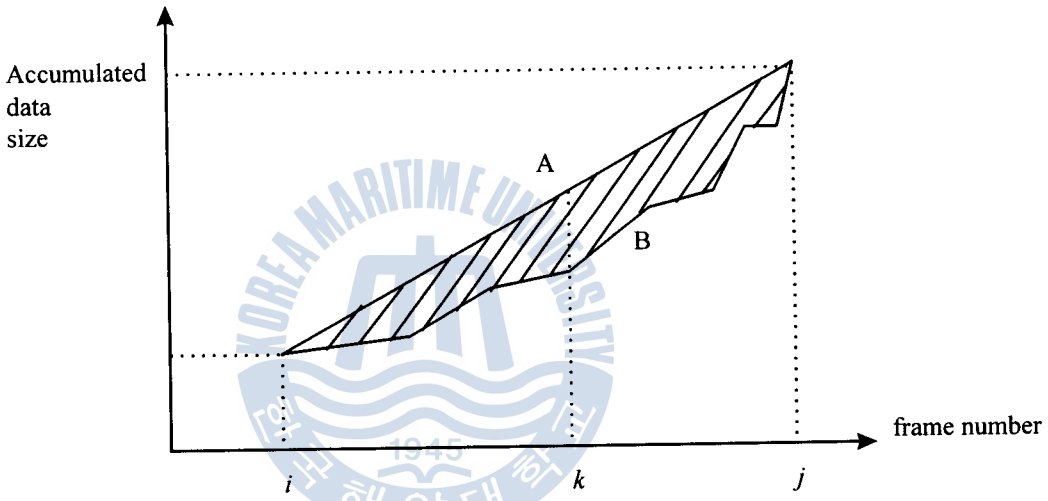


Figure 2. Proof of Theorem 1

As we are interested in the timely delivery of video data to receivers[1], the frame size after smoothing can not be set to a_{1N} equally to all frames, because with that, data exceeding a_{1N} for certain frames should be transferred to frames following the particular frame, not preceding it. In this case, because one has to wait until the whole excess data is available at the receiver buffer for correct playout, the delay may become intolerable. If the delay is limited to a predetermined value, then the frame cannot be played because of the lack of data at that moment. This effect is known as starvation problem. The running average concept is introduced in this paper to prevent the starvation problem. The running average a_{ij}, defined as average frame size for frames from i to j, is easily computed for all frames. The following theorem with running average prevents starvation.

Theorem 1. If $a_{ij} > a_{ik}$ for all k such that $i < k < j$, then smoothing occurs only in the backward direction for frames in $[i, j]$.

Proof. It is apparent that when the accumulated maximum data volume from transmission is always greater than the actual accumulated frame data size, then smoothing can occur only in the backward direction.

In Figure 2, for the interval $[i, j]$, A represents the maximum allowed data size, a_{ij}*(k-i+1). B represents the actual data, F_{ik}. The dashed area is capacity available for smoothing. When B is always under A in $[i, j]$, the smoothing occurs only in the backward direction. Mathematically, this is,

$$a_{ij} \cdot (k - i + 1) - \sum_{m=i}^{k} f_m \geq 0 \text{ for all } k \text{ in } [i, j] \tag{4}$$

which is

$$a_{ij} \geq \frac{F_{i,k}}{k - i + 1} = a_{ik} \text{ for all } k \text{ in } [i, j]. \tag{5}$$

n

In the first pass of TPMA, once all a_{1k} are calculated, then the maximum one is identified. Suppose this is a_{1k}. We know from Theorem 1, that smoothing for frames 1 to k can be done only in the backward direction. We call now frames from 1 to k, as region 1 (r_1), with smoothed frame size a_{1k}.

Since the a_{1j} ($j>k$) is of no value now, we recalculate the running average for frames after k, such that $a_{k+1,j} = \frac{F_{k+1,j}}{j-k}$. We choose the maximum running average (suppose this is $a_{k+1,m}$), and determine the second region (r_2) as frames from $k+1$ to m, and smoothed frame size for r_2 as $a_{k+1,m}$. This procedure is repeated until the last frame is included in one region. This pass ensures that the smoothing occurs only in the backward direction, and at the same time the bandwidth-time product is minimized since no bandwidth is wasted.

We now present the second theorem of TPMA.

Theorem 2. Let f_k be size for frame k, and $F_{ij} = \sum_{m=i}^{j} f_m$, then $S_k > S_{k+1}$ for all $k \geq 1$, where S_k is smoothed frame size for region k.

Proof. Let region k be from frame $h+1$ to i, and region $k+1$ be from frame $i+1$ to j, where $h < i < j$.

Then $S_k = \frac{F_{h+1,i}}{i-h}$ and $S_{k+1} = \frac{F_{i+1,j}}{j-i}$. By definition of S,

$$\frac{F_{h+1,i}}{i-h} > \frac{F_{h+1,j}}{j-h} \tag{6}$$

which can be expanded like

$$\frac{F_{h+1,i}}{i-h} > \frac{F_{h+1,i}+F_{i+1,j}}{j-i+i-h} \tag{7}$$

Equation (7) is rearranged into

$$\frac{F_{h+1,i}}{i-h} > \frac{F_{i+1,j}}{j-i} \tag{8}$$

Therefore, $S_k > S_{k+1}$. n

The physical meaning of theorem 2 is that by separating region $k+1$ from region k, bandwidth-time product is saved by the amount of S_{k+1} * (length of r_{k+1}).

3.2 The Second Pass of the TPMA Smoothing Scheme

The experimental results of the first pass of TPMA algorithm show that there exist short regions with small number of frames. In order to support these small regions, it is required that bandwidth renegotiation should take place between the source and the ATM network. The renegotiation time is not negligible (in the order of milliseconds which is tens of cell times at typical access link speed). The frequent bandwidth renegotiation is not desirable, and they are removed in the second pass of TPMA.

If the number of frames in a region is less than or equal to ε, then the region is called small region. ε is the time needed to renegotiate the allocation of bandwidth in ATM environment.

At the second pass, the removal process of small regions is applied to smoothed streams from the first pass. The smoothing order in the second pass is the reverse direction of playback from the last region. Assume that an arbitrary region r_2 be a small region. r_1 and r_3 denote the predecessor and successor regions of r_2, respectively. The smoothed frame sizes of r_2, r_1, and r_3 are S_2, S_1, and S_3, respectively. The number of frames in r_2, r_1, and r_3 are denoted by N_2, N_1, and N_3, respectively. The small region r_2 is now included in r_3, and disappears. The excess data from region 2 is passed to region 1, and the smoothed frame size for region 1 is recomputed. In the second pass, three cases according to the position of small region are treated differently.

CASE 1: The small region is located between two non-small regions.

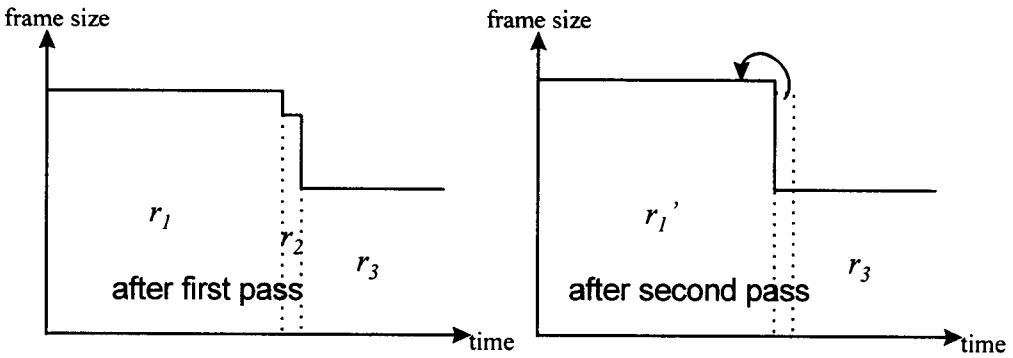


Figure 3. The second pass of the TPMA in CASE 1

In Figure 3, r_2 is partially absorbed in r_3, and the excess part of r_2 is moved to r_1, and consequently S_1 is increased to S_1'. S_1' for r_1' is calculated by

$$S_1' = \frac{\sum_{i \in r1} f_i + \sum_{i \in r2} (f_i - S_3)}{N_1 + N_2} \tag{9}$$

CASE 2 : A small region includes the first frame of a stream.

This is a special case of case 1, where S_1 and N_1 are 0. As shown in Figure 4, r_2 is absorbed in r_3, and the excess part of r_2 is shifted to the frames before 0. As a result, r_1' is created. The smoothed frame size S_1' of r_1'

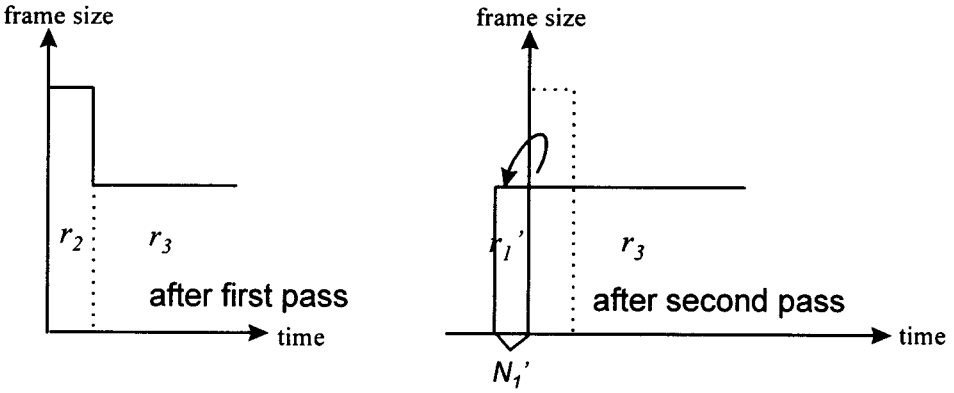


Figure 4. The second pass of the TPMA in CASE 2

was originally 0, and it is now set to r_3 to prevent transmission rate change.

$$S_1' = S_3 \tag{10}$$

The number of frames N_1' of r_1' is calculated by

$$N_1' = \left\lceil \frac{\sum_{i \in r_2} (f_i - S_3)}{S_3} \right\rceil \tag{11}$$

Under this case, the playback startup delay by the amount of N_1' is introduced.

CASE 3 : A small region includes the last frame of a stream.

This is an another special case of case 1, where S_3 and N_3 are 0. As shown in Figure 5, all the frames in r_2 are moved to r_1. As a result, r_2 is removed. The smoothed frame size S_1' of r_1' is re-calculated.

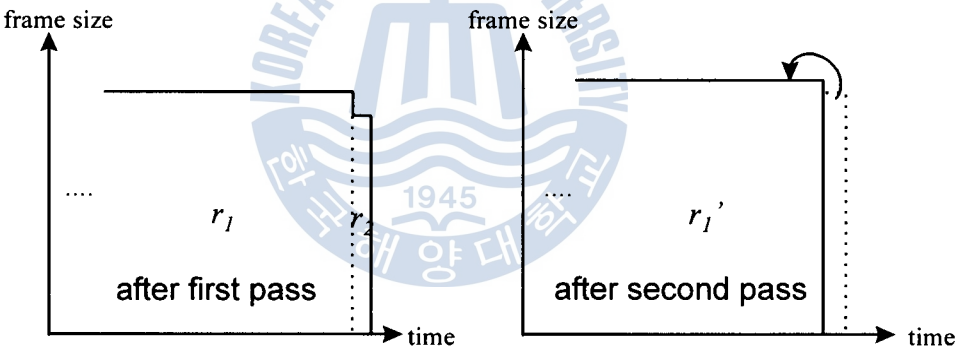


Figure 5. The second pass of the TPMA in CASE 3

This means that the actual transmission time for the entire video stream is shortened by N_2. S_1' is evaluated by

$$S_1' = \frac{\sum_{i \in r_1, r_2} f_i}{N_1} \tag{12}$$

4. Buffer Requirements at the Receiver

The excess data of a frame k larger than the smoothed frame size in a region is moved

into some smaller frames preceding k. When the excess part of a frame is moved to predecessor frames smaller than the smoothed frame size, at first the frame among the smaller predecessor frames nearest to k is filled up to the smoothed frame size. This frame is called $NEAR_k$. This filling operation is continued until the excess part is transferred in total. The filled part in a smaller predecessor frame is called segment. The farthest frame from k that has segment from k is called FAR_k.

It can be easily observed that the stream smoothed by this procedure has the following characteristics.

(1) All the excess data of frame k is moved to frames from FAR_k to $NEAR_k$.
(2) If $FAR_k \neq NEAR_k$, every segment in frame i ($FAR_k \leq i \leq NEAR_k$) belongs to k.
(3) All the segments of the predecessor of k in playback time are located in frame i, $i \leq FAR_k$.
(4) All the segments of the successor of k in playback time are located in frame i, $i \geq NEAR_k$.

Theorem 3. Let k and l be two arbitrary frames in a VBR video stream, and $k < l$. The two-pass maximum average smoothing scheme guarantees that Equation 13 always holds.

$$NEAR_k \leq FAR_l \tag{13}$$

Proof. It is easily derived from the characteristic (1) through (4). n

The buffer requirement B_k for frame k can be evaluated with Theorem 3.

$$\begin{aligned} B_k &= (\text{excess data from frame } k) + (\text{excess data from frames after } k) \\ &= \sum_{i=FAR_k}^{NEAR_k} e_i + \sum_{i=NEAR_k+1}^{k} e_i \end{aligned} \tag{14}$$

where e_i is segment size for frame i.

It is noted that the first term in the right-hand side disappears when size of original frame k is smaller than the smoothed frame size.

Consequently, the buffer requirement at a client system to playback a smoothed video stream is the maximum of B_k.

$$\text{Client\_Buffer\_Size} = \max_{k=1}^{N}(B_k) \tag{15}$$

5. Experiments

TPMA smoothing scheme is experimented with several well-known VBR MPEG video streams used in the previous studies. We present here the results of two video streams only, RedsNightmare.mpg and mobile.mpg. The complete results can be found in [12].

The results of the TPMA smoothing scheme shown in Figure 6, 7, 8 and 9 describe well the smoothing effect. In case of RedsNightmare.mpg, the bit rate in smoothed stream is changed only four times during the whole transmission. Moreover, it is reduced to only one in mobile.mpg case.

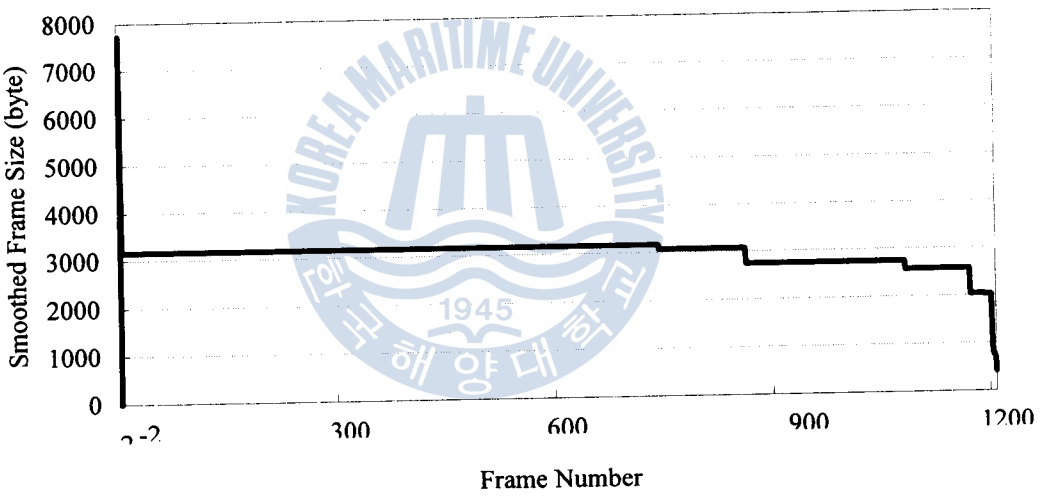


Figure 6. The smoothed stream of RedsNightmare.mpg after the first pass

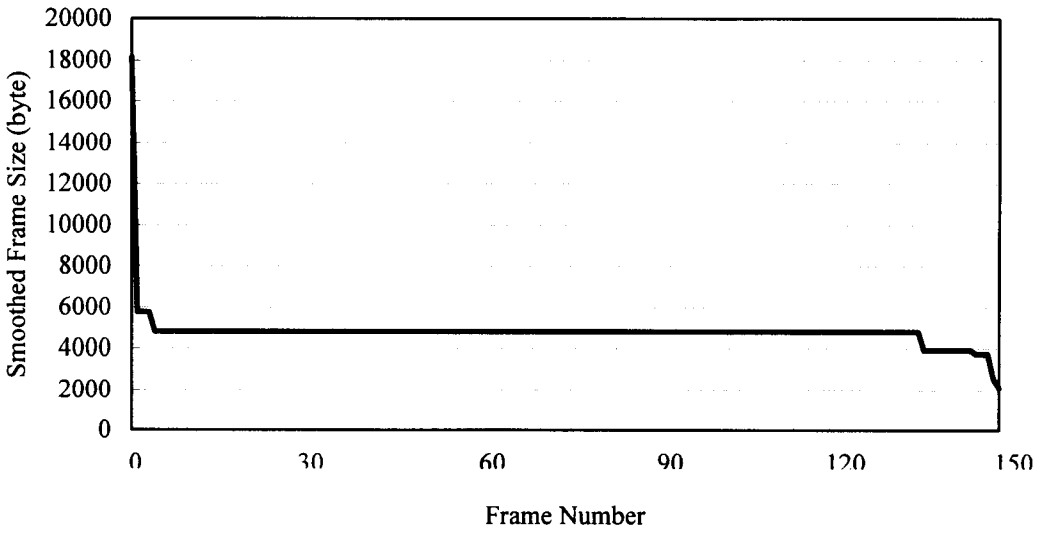


Figure 7. The smoothed stream of mobile.mpg after the first pass

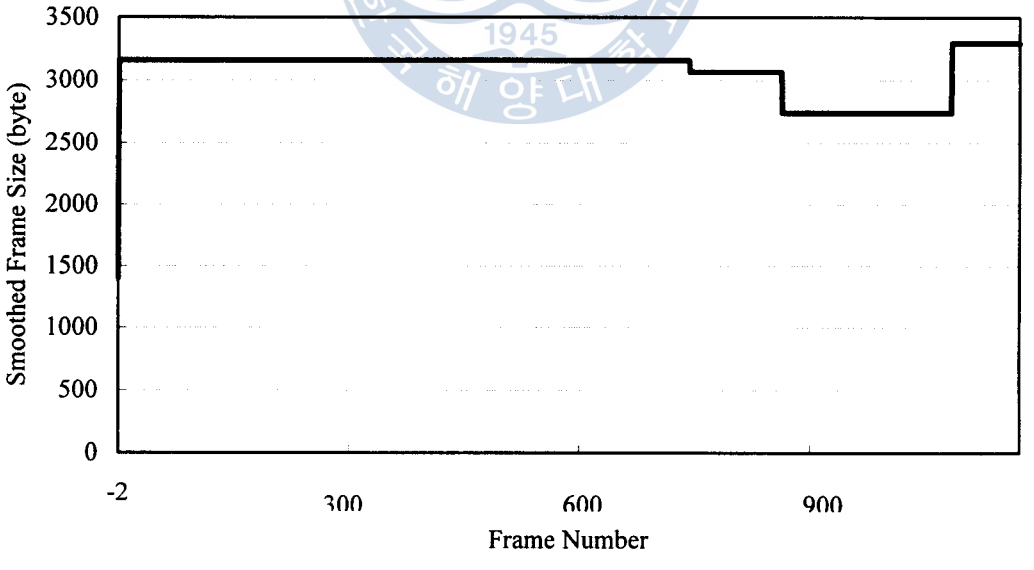


Figure 8. The smoothed stream of RedsNightmare.mpg after the second pass

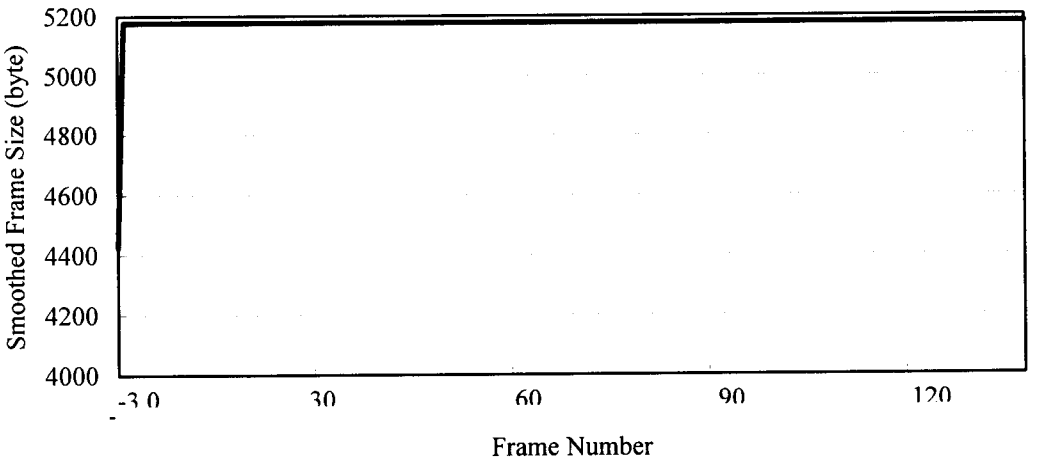


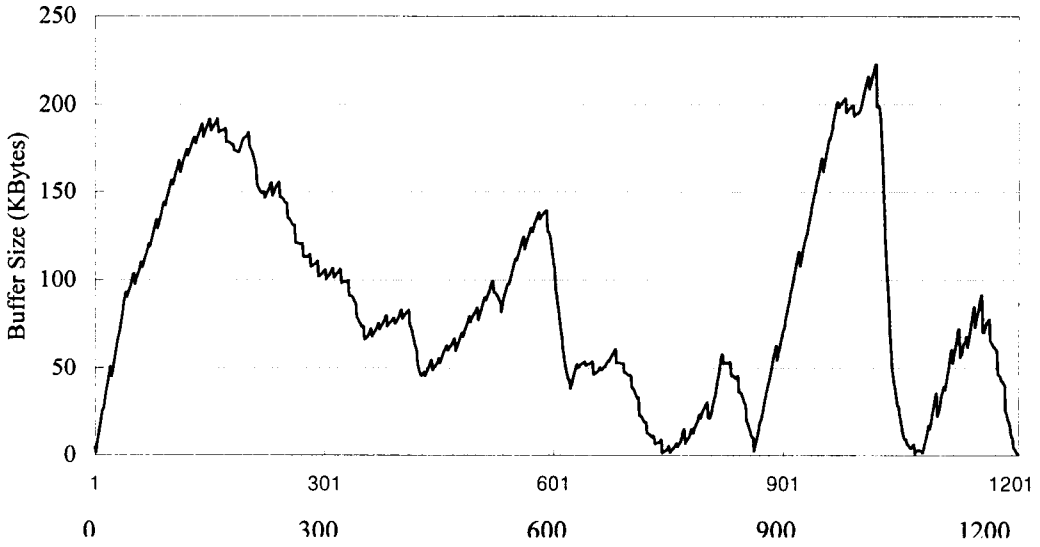
Figure 9. The smoothed stream of mobile.mpg after the second pass

Table 1. Maximum and Average Buffer requirements at the receiver

	Max. Buffer Size	Ave. Buffer Size
RedsNightmare.mpg	222,398 bytes(74.3 frames)	87,923 bytes(29.4 frames)
Mobile.mpg	66,623 bytes(13.9 frames)	34,785 bytes(7.3 frames)

To remove small regions, we assume ε as 30. It actually depends on the link access speed, and the signaling delay. The results are summarized in Figures 6, 7, 8, and 9. These are the smoothed streams after first pass and second pass respectively. In the final results, the first frame number of region 1 is negative. Negative number indicates that region 1 is extended to the area during which frames are transmitted but not played back yet.

The peak to average ratio(PAR) in frame size is also measured in both cases of unsmoothed and smoothed stream. For RedsNightmare.mpg, PAR of unsmoothed stream is 9.3, which it is largely reduced to 1.1 after smoothing. In mobile.mpg case, 4.31 is changed to 1.1.



igure 10. The required buffer size at receiver (RedsNightmare.mpg)

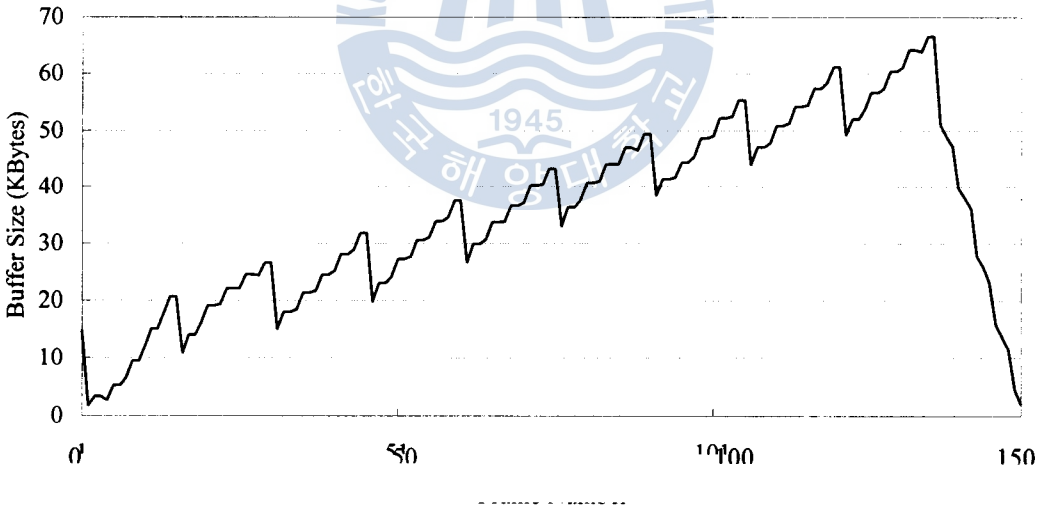


Figure 11. The required buffer size at receiver (mobile.mpg)

5.2 Buffer requirements at the receiver

The maximum and the average amount of buffer requirements for

RedsNightmare.mpg and mobile.mpg are summarized in Table 1. The buffer requirements are represented in bytes and also in the number of average-sized frames.

The maximum buffer size must be considered at receivers. For RedsNightmare.mpg, 222 Kbytes are needed to playback without data starvation. It is roughly for four seconds' playback.

Such a buffer requirement is easily feasible in ordinary PC. In Figure 10, the required buffer size is reduced to 0 at four positions. These positions are just the boundaries between regions. In Figure 11, since the number of regions finally produced by TPMA smoothing scheme is only one, the buffer size becomes to 0 only once. In Figure 11,the saw-tooth pattern is due to periodic consumption of I-frames.

6. Concluding Remarks

We proposed in this paper a new and efficient video traffic smoothing scheme that is applicable to stored video data. The TPMA smoothing scheme ensures large reduction in the number of rate changes at the source output. Bandwidth renegotiation between the video source and the ATM network thus happens less frequently. The peak-to-average ratio after smoothing is shown to approach one (the ideal case) in the experiments performed on the well-known video stream data.

The buffer requirement at the receiver is easily calculated and shown to be feasible at ordinary PC systems. The constant video quality in playback is fully guaranteed under TPMA as opposed to previous works on smoothing. As for further research items, smoothing under buffer restriction, and extension to online realtime smoothing look promising.

References

[1] P.V. Rangan, H. M. Vin, and S. Ramanathan, "Designing an On-Demand Multimedia Service," IEEE Communications Magazine, Vol. 30, No. 7, pp. 56-64, July 1992.

[2] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta, and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks," Proc. IEEE INFOCOM '93, pp.919-926, 1993.

[3] P. Pancha and M. E. Zarki, "MPEG Coding For Variable Bit Rate Video Transmission," IEEE Communications Magazine, pp. 54-66, May 1994.

[4] N. Shroff and M. Schwarz, "Video Modeling within Networks using Deterministic Smoothing at the Source," Proc. IEEE INFOCOM '94, pp.342-349, 1994.

[5] R. P. Tsang, D. H. C. Du, and A. Pavan, "Experiments with Video Transmission over an Asynchronous Transfer Mode (ATM) network," ACM Multimedia Systems, Vol. 4, No. 4, pp.157-171, 1996.

[6] T. Ott, T.V. Lakshman, and A. Tabatabai, "A Scheme for Smoothing Delay-Sensitive Traffic Offered To ATM Networks," Proc. IEEE INFOCOM '92, pp.776-785, 1992.

[7] S. Jung and J. S. Meditch, "Adaptive Prediction and Smoothing of MPEG Video in ATM Networks," Proc. IEEE ICC '95, pp.832-836, 1995.

[8] P. Pancha, and M. E. Zarki, "Prioritized Transmission of Variable Bit Rate MPEG Video," Proc. of IEEE GLOBECOM '92, pp.1135-1139, 1992.

[9] W. Feng, and S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Video," Computer Communications, Vol.18, No.10, pp.709-717, Oct. 1995.

[10] P. Skelly, M. Schwartz, and S. Dixit, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer," IEEE/ACM Transactions on Networking, Vol.1, No.4, pp.446-459, Aug. 1993.

[11] ISO/IEC IS 11172-2, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s".

[12] J. Y. Son, "Storage and Retrieval Schemes for VBR Streams in Video Servers," Ph.D. Dissertation, Seoul National University, Aug. 1997.