ISSN 2250-3153

Comparison of Traffic Smoothing Algorithms for MPEG Videos

Manisha Mane[*], V. B. Dharmadhikari [**]

[*] Department of electronics engineering, SAKEC Mumbai,
[**] Department of electronics engineering, WCE Sangli

Abstract- In a typical video application, such as video-on-demand, videos are continuously streamed from a video server to a set of receivers. The constant-quality video encoding technique and variable bit rate (VBR) encoding techniques used. Variable bit rate (VBR) video transmission leads to the burstiness of video traffic, hence high fluctuation in bandwidth requirement. Traffic smoothing algorithms are very efficient in reducing burstiness of the VBR video stream by transmitting data in a series of fixed rates.

In this paper, we examine traffic smoothing algorithms and their performance analysis, using a MPEG encoded video and simulation results showed that our approach has small bandwidth requirement, high bandwidth utilization and low computation cost.

***Index Terms*-** Segmentation, Traffic smoothing, Variable bit rate (VBR) video, wavelets.

I. INTRODUCTION

Video applications, such as video-on-demand services, necessitate the utilization of a large amount of network bandwidth and storage space. In live video, there is a requirement that transmission decisions be made in real time and some initial delay between sender and receiver is tolerable [4]. As the popularity of video-on-demand services increases, much of network traffic will be transmission of data from prerecorded video sources.

Usually, video sources are encoded to reduce storage and bandwidth requirements. Some videos are encoded using a method called constant-bit-rate (CBR) encoding, which simplifies network bandwidth allocation. However, CBR coding produces video with varying quality, for the number of bits used to encode each frame must remain the same for every frame, even during periods of fast action or high detail when more bits are needed to represent such frames with more variation. Such variable-quality encoding is not appropriate [5].

A method called variable-bit-rate (VBR) coding produces video streams with constant quality, and for a given bandwidth VBR encoded videos have a higher perceivable quality than CBR video streams [5] [6] [7]. Yet there is a trade-off for this increase in quality; VBR encoded video streams exhibit significant rate variability. Without intelligent traffic shaping, transmission of VBR video streams would lead to inefficient network bandwidth utilization. This intelligent traffic management is implemented by a class of algorithms, known as bandwidth smoothing algorithms.

II. TRAFFIC SMOOTHING FOR VBR VIDEO

A video server can significantly reduce the bandwidth requirements for transmitting stored video by pre-fetching video data into the client playback buffer in advance of each burst. Figure1 shows transporting pre-stored video stream through networks to the client side. We consider a discrete-time model at the frame level. That is $t \in \{1,2....,N\}$, where N is the length of the video in frames. The server stores the entire video stream and generates a transmission plan based on the overflow and underflow constraints on the client buffer. The server writes the stream as a series of CBR rates denoted by $r(t)$ into a network for transmission. On the client side, the playback buffer has capacity of B bytes. $S(t)$ denotes accumulated video data transported to the client. $B(t)$ represents the buffer occupancy at time t, and $L(t)$ the accumulated video data that played back.

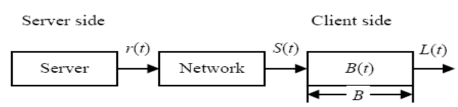

Figure1 Transport of pre-stored video through networks

To permit continuous playback at the client site, the server should always avoid underflow by serving enough data. On the other side, since the data received at the decoder is stored in a buffer, and if the client receives too many data that exceed the capacity of the client's buffer, the data corresponding to a video frame will be useless and the frame will thus be considered lost, this situation is called decoder buffer overflow.

Consider a video sequence with N frames, where frame i is fi bytes long. In order to avoid buffer underflow, the server must always transmit more data than the decoder consumes, so that by the time the client decodes the t^{th} frame, t=1, 2,…, N, it must have received at least $L(t)$ bytes from the server, where

$$L(t) = \sum_{i=1}^{t} fi \qquad (t = 1, 2, \ldots N)$$

In the same way, a client should receive data no more than fi by frame time t to prevent playback buffer overflow.

$$U(t) = B + \sum_{i=1}^{t} fi$$

A feasible transmission schedule $S(t)$ should stay within the constrained region set by the constraint curves $U(t)$ and $L(t)$ shown in Fig.2. That is

$$L(t) \le S(t) \le U(t)$$

where

$$S(t) = B + \sum_{i=1}^{t} fi$$

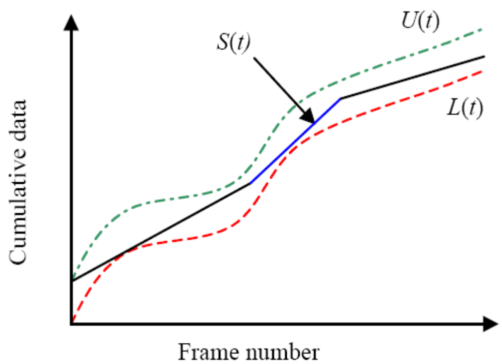

Figure.2 Transmission schedule

III. BANDWIDTH SMOOTHING ALGORITHMS

3.I): Segment-Based Traffic Smoothing Algorithm:

In this section, a segment-based traffic smoothing algorithm is proposed.[2] We construct a feasible transmission schedule based on the following criteria. 1) each CBR transmission segment should be as long as possible, in order to make the traffic as smooth as possible. 2) to avoid underflow, the server should transmit enough data to permit continuous playback at the client site; to avoid overflow, the server must limit the amount of prefetching to client buffer; so the transmission rate must be increased or decreased to ensure feasibility. 3) in order to minimize the possibility of overflow or underflow but make each video segment as long as possible, the starting point of a segment should be far away from the boundary of the constrained region, so the middle point of the constrained region is selected as the starting point of each run.

Half of the capacity of the client buffer size ($B/2$) is selected as the initialization buffer size. Taking the average transmission rate of each segment as the transmission rate of the segment makes the buffer occupancy at the starting of each segment the same as the initialization buffer size. Each run ends at the middle of the capacity of the client buffer where it starts, each portion of the video stream corresponding to a run is called a segment. And so algorithm is also called segment-based traffic smoothing algorithm.

A cost function $C(j,k)$ associated with a transmission schedule, which represents the maximum client buffer requirement over interval $[j,k]$, is given by,

$$C(j,k) = \max_{j \le t \le k} \left| \sum_{i=j}^{t} r_{(i)} - fi \right|$$

where

$$r(i) = \sum_{t=j}^{k} f_t \Big/ (k-j)$$

is the transmission rate during frame slot i of the smoothed video stream which equals the average rate of the segment over interval $[j,k]$. The cost must be obviously smaller than $B/2$ to guarantee no overflow or underflow of the playback buffer, because half of the capacity of client buffer size ($B/2$) is selected as the initialization buffer size.

3.II): Piecewise Constant Rate Transmission and Transport (PCRTT)

The piecewise constant-rate transmission and transport (PCRTT) algorithm [4] divides the video stream into segments with fixed size intervals to create a transmission schedule. The transmission rate for the algorithm is set by taking the average frame size for each segment; each segment corresponds to one run in the transmission schedule.

Transmission rate during frame slot i of the smoothed video stream which equals the average rate of the segment over interval $[j,k]$ is given by,

$$r(i) = \sum\nolimits_{t=j}^{k} \frac{f_t}{(k-j)}$$

Then the algorithm raises the transmission schedule to avoid client buffer underflow. An initial delay is introduced to the plan so that the client buffer will contain data when playback begins. From the transmission schedule, the algorithm computes the minimum client buffer size to avoid overflow.

The main advantage of this method over other methods is that for small buffer sizes, PCRTT creates bandwidth plans that have near optimal peak bandwidth requirements, while requiring very little computation time. Since a PCRTT plan consists of fixed-size intervals, the bandwidth changes occur after constant times. This can be useful for the multiplexing of several streams. Another advantage of PCRTT is that it can produce bandwidth plans with a meaningful lower bound on the minimum time between rate changes

3.III):Dynamic Programming Based Smoothing:

This dynamic-programming (DP) smoothing algorithm is a bandwidth smoothing algorithm which permitting rate changes at any multiple of L without restriction. The algorithm produces the optimal transmission schedule. We assume that the video length N is a multiple of the segment size L, N = ML, with M time segments numbered from 0 to M-1.

We apply DP to find the optimal transmission schedule for subset of the video stream containing frames [0, kL-1] for $k \in \{1, 2, \ldots, M\}$. Fig.3 provides an explanation of DP approach. First, we focus on the first two segments of the video stream [0, 2L-1]. There are two possible transmission schedules, and finding the winning one with the smaller cost involves little computational effort. Then, we consider the optimal segmentation of the first three segments [0, 3L-1]. There are four possible ways of placing rate changes into the transmission schedule no rate changes, at frame L, at frame 2L, and at both L and 2L but our previous comparison allows us to reduce the comparisons to three options, since we know the optimal transmission schedule for the two cases where there is a rate change at frame 2L. figure 3 explains the description for this algorithm is presented in [1].

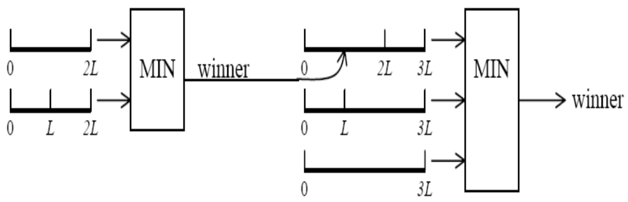

Figure.3: fig shows dynamic programming process.

If we define Ck* as the minimum cost associated with the best transmission schedule x* for frames [0, kL-1], then the relation

$$C^*_k = \min_{0\le t\le k}\left[\max(C^*_t, \Delta C_{tk})\right]$$

Is satisfied for k ∈ {1, 2, … , M} where ΔCt,k is the minimum cost associated with transmitting at rate u(j) over tL ≤ j ≤ kL-1 and u(j) is the transmission rate at frame-time n. when k=M,

$$\Delta C_{t,k} = \max_{i\in[tL,kL-1]}\left|\sum_{j=0}^{i}(u_{(j)} - f_j)\right|$$

to find the optimal transmission schedule for the video stream, once the minimum costs found for k ∈ {1, 2, … , M}, then a backtracking function b(k) of k, is defined as

$$b(k) = \arg\min_{0\le t\le k}\left[\max(C^*_t, \Delta C_{t,k})\right]$$

to recover the optimal transmission schedule.

3.IV): Wavelet-based Traffic Smoothing (WTS) Algorithm:

Recently, wavelets [8]–[9] have been well developed as a multi resolution signal analysis and processing tool and applied successfully in many fields, such as image compression, denoising and network traffic modelling. Since wavelets can analyze and reconstruct signals with multiple resolutions, they are suitable for approximating video traffic at multiple scales. We use the Haar wavelet due to its simplicity and the averaging (smoothing) effect the low-pass Haar filter has on video traffic.
Existing traffic smoothing algorithms do not address variability on multiple levels because they implicitly treat a video stream as having only one resolution In contrast, proposed WTS algorithm provides traffic smoothing at multiple resolutions. The wavelet tree structure in WTS makes the algorithm flexible in its behaviour.

The discrete Haar wavelet transform represents a 1-D signal x(t) of length N in multiple resolutions. The analysis at different scales can be represented by a binary tree T_{max} of lowpass wavelet coefficients as shown. Assume that T_{max} has depth J_{max}. In the wavelet transform, j indexes the scale of analysis: j=0 indicates the coarsest scale or lowest resolution of analysis, which corresponds to the root of the binary tree; the larger the j, the higher the resolution of the analysis. We index each node of T_{max} at depth j($0\le j\le J_{max}$) by a tuple (j,k) and associate it with the low-pass wavelet coefficient given by

$$u_{j,k} = \frac{2^j}{N}\sum_{i=N\frac{k}{2^j}}^{\frac{Nk+1}{2^j}-1} x(i),$$

for k=0,…..2^j-1.

The wavelet-based traffic smoothing (WTS) algorithm [8] calculates a binary tree in which each node represents smoothing at different resolutions. The full tree corresponds to the original video, with each segment of the video stream matching to a leaf node, while a node at a higher level stores one transmission rate for multiple segments of the video. This algorithm builds a binary tree by setting transmission rates for all leaf nodes to the average frame size for each segment; then, in a bottom-up traversal, non-leaf node transmission rates are set to the average rates of each node's children. Next, the algorithm associates a cost with each node at resolution *j* and offset *k* using as the cost metric the minimum client buffer requirement when *rj,k* is set as the transmit rate over frames

$[2^{-j}KN, 2^{-j}(K+1)N-1]$ is given by

$$Cj,k = \max_{t\in[2^{-j}KN,2^{-j}(K+1)N-1]}\left|\sum_{i=0}^{t} r_{(j,k)} - fi\right|$$

Finally, the WTS algorithm prunes the binary tree to the smallest size that satisfies the constraint requiring the maximum cost of the pruned tree's leaf nodes to be less than the client buffer size. Typically, a node higher in the tree, one which generates a longer run, requires a larger client buffer size to avoid buffer underflow than does one of its child nodes, so a balance is made between runs of greater length and the client buffer size required by a transmission schedule comprised of such runs.

IV. PERFORMANCE EVALUATION RESULTS

In this section, we compare the segment based algorithm, DPS, PCRTT and WTS smoothing algorithms based on widely accepted performance metrics including rate changes per unit time, peak rate requirements, and variability of bandwidth requirements.

To evaluate how these algorithms perform over time, we utilize M-PEG encoded video. For our comparisons we select a 13245ms long video, encoded at different frames per second quality, resulting in an average bit rates. By testing performance of smoothing algorithms across a wide range of typical client buffer sizes, we get an accurate measure of how they perform under realistic settings.

SUMMERY TABLE

Name of algorithm	Segment based algorithm	DPS	PCRTT	WTS
Rate variation (min-max)in bytes	760	3036	3036	29
	330586	324345	324345	467883
Peak rate(bytes)	330586	324345	324345	467883
Max Buffer size (bytes)	165421	14687	297733	137040

a) Bandwidth variability:

A transmission schedule with a lower bandwidth variation is desirable, as it requires fewer resources from the server and network. Bandwidth variation is decided by the rate changes. The DPS and PCRTT algorithm results in fewer rate changes. Segment based algorithm results in fewer rate changes than the WTS algorithms but its rate changes are more than DPS & PCRTT for a given buffer size.

b) Peak bandwidth requirements:

The peak rate of a smoothed video stream determines the peak bandwidth requirement across the network. Hence, most bandwidth smoothing algorithms attempt to minimize the peak rate. Table results shows DPS & PCRTT algorithm require same low peak rate and segment based algorithms require low peak rate, whereas WTS require less high peak rate.

c) Buffer utilization:

Buffer occupancy is higher for video streams with high activity and high variability in the original frame sizes. If the original stream is less bursty, buffer utilization can be improved because all intervals demand almost the same buffer size.

Buffer occupancy during the playback of a video stream is the number of bytes at the buffer. And it is determined by subtracting the decoding rate at the buffer output from the smoothing rate-plan at the buffer input. From the results presented as in table, DPS and segment based algorithms require less buffer size than PCRTT & WTS.

V. CONCLUSION

In this paper, we examine existing algorithms such as Segment based bandwidth algorithm, PCRTT, DPS and WTS, Their experimental results shown in table. Comparison of the traffic smoothing algorithms based on 1. less rate variability, 2. lower peak rates, and 3. smaller client buffer size requirements.

DPS algorithm gives best results as its rate variation is less, it has low peak rate value, and maximum buffer size requirement is also less comparative to other algorithms.

PCRTT algorithm gives same results as DPS but its buffer size requirement is more than DPS.

Moderate results are given by the algorithm called segment based. Its rate variation is moderate,
Peak rate requirement is similar to DPS & PCRTT algorithms; its maximum buffer size requirement is also low.

The algorithm WTS fails to give good results on the performance metrics as it require large value of peak rate, large buffer size and its min to max rate variation is also high.

From the results we can conclude that WTS performs worst for the given analysis but still WTS is applicable for the network with multiple-clients, whereas the other methods are for single client users.

Thus the applications that require the storage and transmission of compressed video, such as video-on-demand services and digital libraries, such bandwidth smoothing techniques plays an important role in efficient network management.

REFERENCES

[1] Cam Barker1, Zixiang Xiong1, and Anthony Kuh2 "Dynamic Programming Based Smoothing of VBR Video Traffic" March 6, 2002.

[2] LIU Yun-qiang, YU Song-yu, WANG Xiang-wen, ZHOU Jun, "Segment-based traffic smoothing algorithm for VBR video stream" Oct. 21, 2005.

[3] O. Hadar and R. Cohen, "PCRTT enhancement for off-line video smoothing," *Real-time imaging*, vol. 7, pp. 301-304, June 2001.

[4] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video," *IEEE Transactions on Multimedia*, vol. 1, September 1999.

[5] A. Reibman and B. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, pp. 361-372, December 1992.

[6] T. V. Lakshman, A. Ortega, and A. R. Reibman, "Variable bit-rate (VBR) video: Tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, pp. 952-973, May 1998.

[7] J. M. McManus and K.W. Ross, "Video on demand over ATM: Constant rate transmission and transport," *IEEE J. Select Areas Commun.*, vol 14, pp. 1807-1098, August 1996.

[8] Dejian Ye, J. Cam Barker, Zixiang Xiong, and Wenwu Zhu, "Wavelet-Based VBR Video Traffic Smoothing", *IEEE Transactions on Multimedia*, VOL. 6, NO. 4, AUGUST 2004.

[9] Dejian Ye, Zixiang Xiong, Huai_Rong Shao,Qiufeng Wu and Wenwu Zhu, "Wavelet-based Smoothing and multiplexing of VBR Video traffic" *IEEE Transactions on Multimedia* 0-7803-7206-9/01, 2001

AUTHORS

First Author – Manisha Mane, Department of electronics engineering, SAKEC Mumbai,
Email:manishavivekmane@yahoo.co.in
Second Author – V. B. Dharmadhikari, Department of electronics engineering, WCE Sangli,
Email:vb_dharmadhikari@yahoo.com