Wavelet-based Smoothing and Multiplexing of VBR Video Traffic

Dejian Ye[1,2], Zixiang Xiong[3], Huai-Rong Shao[2], Qiufeng Wu[1], and Wenwu Zhu[2]

[1]Department of Automation, Tsinghua University, Beijing 100084, P. R. China
[2]Microsoft Research China, No. 49, Zhichun Road. Haidian District, Beijing 100080, P. R. China
[3]Dept of Electrical Engineering, Texas A&M University, College Station, TX 77843

Abstract—**Although VBR coding is more efficient than CBR coding, the burstiness of VBR video traffic brings great difficulty to network resource management. To address this problem, we propose a novel wavelet-based traffic smoothing (WTS) algorithm. Unlike existing algorithms, which only have one resolution, the WTS algorithm has the multiresolutional property that is preferable for smoothing VBR video traffic that exhibits a self-similar behavior. WTS allows traffic smoothing at multiple resolutions and the best transmission schedule is searched as a pruned subtree of a full binary tree, which corresponds to the original VBR video traffic. WTS optimizes several metrics simultaneously for both the single and multiple-flow case while traditional algorithm only optimize one or two metrics for a single flow. The computation complexity of the WTS algorithm is also lower.**

I. Introduction

It is well known that VBR coding is more efficient than CBR coding in terms of video quality. Unfortunately, from the network point of view, it is very difficult to manage VBR video traffic because of large rate variations. Allocation of network resource based on the peak rate will results in very low network utilization. Although statistic multiplexing may help improve network utilization for VBR video traffic, it is not efficient when the number of multiplexed video clips is small. It is thus necessary to consider smoothing the VBR traffic before transmission in applications like video on demand.

For pre-recorded video, transmitting frames to a client buffer prior to each burst can reduce the rate variability. Following this philosophy, many efficient approaches [1-7] have been proposed for traffic smoothing (e.g., the MCBA algorithm [2], the PCRTT algorithm [4], and the MVBA algorithm [6]). With these algorithms, the video server can pre-compute a transmission schedule that minimizes some performance metric while preventing buffer overflow and underflow at the client side. Different metrics (e.g., minimal rate variability, minimal number of rate changes, largest lower bound on the time between two consecutive rate changes, etc.) are used in these algorithms, as it is not easy to design one algorithm that optimizes several metrics simultaneously. Another drawback of these algorithms is that they are only designed for smoothing one single video trace and they do not perform well when multiple smoothed flows are multiplexed together.

Several recent empirical studies have demonstrated the *self-similar* nature of VBR video traffic. This self-similar or *fractal*-like traffic behavior manifests itself as traffic "spikes" riding on longer-term "ripples" that in turn rides on still longer term "swells". All existing traffic smoothing algorithms fail to exploit this complex behavior of VBR traffic in the time domain because they only have one resolution -- all of them treat the whole video trace as a whole.

To rectify this shortcoming, we propose a novel wavelet-based traffic smoothing (WTS) algorithm in this paper. We assume that the Haar wavelet bases are used due to their simplicity and the averaging (smoothing) effect the lowpass Haar filter has on the video traffic. Unlike existing algorithms, the WTS algorithm allows traffic smoothing at multiple resolutions – smoothing with one mean at the lowest resolution; two local means at the next resolution, and up to four local means at yet the next resolution, etc. As illustrated in Fig. 1, we associate each possible transmission schedule in WTS with a pruned subtree of a full binary tree, which is the original VBR traffic, and search for the optimal schedule that best matches the client buffer constraint.

Because the scaling structure of wavelet bases naturally matches the self-similar nature of VBR video traffic, we conjecture that WTS should perform well. On one hand, the optimal transmission schedule in WTS in general results in different smoothing intervals for different parts of the video

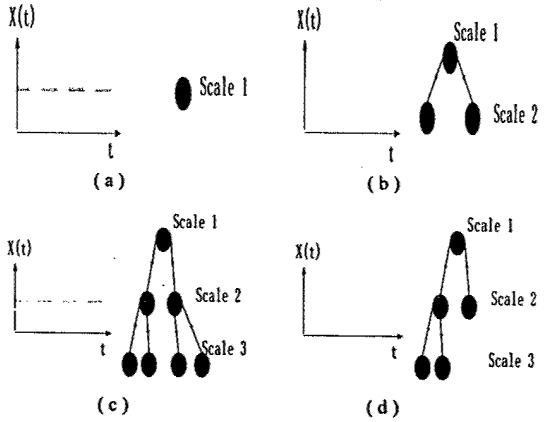


Fig.1. Each possible transmission schedule in WTS corresponds to a pruned subtree of a full binary tree and the optimal schedule is searched to best match the client buffer constraint.

0-7803-7208-5/01/$17.00 (C) 2001 IEEE

Because the scaling structure of wavelet bases naturally matches the self-similar nature of VBR video traffic, we conjecture that WTS should perform well. On one hand, the optimal transmission schedule in WTS in general results in different smoothing intervals for different parts of the video trace, achieving a balance between having minimal rate variability in MVBA [6] and having the minimal number of rate changes in MCBA [2]. On the other hand, when WTS chooses a transmission schedule that corresponds to a full binary tree of any depth, it degenerates to PCRTT [4], which uses a fixed smoothing interval.

Indeed, our analysis and simulation results show that WTS is efficient in rate smoothing and easy to implement. Compared to PCRTT, it enforces a lower bound on the time between two consecutive rate changes while keeping the following parameters small: number of rate changes, peak rate, and bandwidth variability. Thus WTS achieves the goal of performing reasonably well under several performance metrics simultaneously in one algorithm.

When we multiplex multiple smoothed flows together after WTS, overall we obtain much better results than using other smoothing algorithms.

The rest of this paper is organized as follows: In section2 we introduce WTS in Section 2 and describe the WTS algorithm and its property in Section 3. Experiment results for single-flow smoothing and multiple-flow multiplexing are presented in Section 4. Section 5 concludes the paper.

II. WAVELET-BASED TRAFFIC SMOOTHING

The discrete wavelet transform represents a 1-D real signal $X(t)$ in terms of shifted and dilated versions of a prototype bandpass wavelet function $\psi(t)$ and shifted versions of a lowpass scaling function $\phi(t)$. For special choices of the wavelet and scaling functions, the atoms

$$\psi_{j,k}(t) := 2^{j/2}\psi(2^j t - k),$$
$$\phi_{j,k}(t) := 2^{j/2}\phi(2^j t - k), \quad j,k \in Z \tag{1}$$

form an orthonormal set of bases, and we have the signal representation

$$X(t) = \sum_k u_{J_0,k}\phi_{J_0,k}(t) + \sum_{j=J_0}^{\infty}\sum_k w_{j,k}\psi_{j,k}(t), \tag{2}$$

with

$$w_{j,k} := \int X(t)\psi_{j,k}(t)\,dt, \tag{3}$$

$$u_{j,k} := \int X(t)\phi_{j,k}(t)\,dt, \tag{4}$$

For a wavelet $\psi(t)$ centered at time zero and frequency f_0, the wavelet coefficient $w_{j,k}$ measures the signal content around time $2^{-j}k$ and frequency $2^j f_0$. The scaling coefficient $u_{j,k}$ measures the local mean around time $2^{-j}k$. In the wavelet transform, j indexes the scale of analysis: J_0 indicates the coarsest scale or lowest resolution of analysis; the larger j the higher resolution of the analysis.

The Haar scaling functions and wavelet provide the simplest example of orthonormal wavelet bases. The analysis at different scales can be represented by the binary tree as is shown in Fig 2.

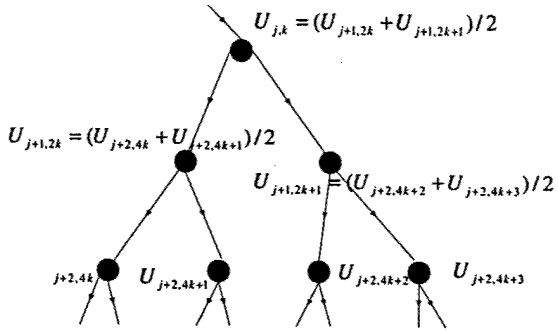


Fig. 2: Binary tree of Haar scaling coefficients from coarse to fine scales.

In traffic smoothing, the server must always transmit enough but not too much data to permit continuous playback on the client site and to avoid client buffer overflow. These are often called buffer overflow and underflow constraints. These constraints dictate that most smoothing algorithms act like lowpass filters. Since Haar scaling coefficients $u_{j,k}$ also act as low pass filters around time $2^{-j}k$, it is suitable to smooth VBR video traffic using the Haar scaling functions at different scales.

Assume a compressed video stream consists of n frames, where frame i requires $X(i) = f_i$ bytes of storage. First, we apply the Haar wavelet analysis to frame size signal $X(i)$ and obtain binary tree of scaling coefficients shown in Fig. 2.

Second, we calculate the cost $C_{j,k}$ related to each scaling coefficients $u_{j,k}$ according to the well-known queuing model:

$$C_{j,k} = \max_{t \in [2^{-j}k,\,2^{-j}(k+1)]} \left| \sum_{i=1}^{t} (u_{j,k} - x(i)) \right|, \tag{5}$$

here, the cost $C_{j,k}$ represents the maximum client buffer requirement when we set $u_{j,k}$ as transmit rate over interval $[2^{-j}k, 2^{-j}(k+1)]$.

At last, we search binary tree of scaling coefficients (see Fig. 2) to find the resolution scales with suitable costs.

III. WAVELET-BASED TRAFFIC SMOOTHING AND MULTIPLEXING

We design WTS algorithm based on the following fact: the lower resolution of the analysis the more smooth transmit plan. Thus, we can search along the binary tree in a *top-down* fashion and stop at nodes wherever $C_{j,k} \le b_0 / 2$.

We use the following recursion to implement our smoothing algorithm:

WTS ALGORITHM

1. Perform wavelet transform on $X(i)$ to obtain binary tree of scaling coefficients
2. $j = 0$
3. if { $C_{0,0} > b_0 / 2$ } TopdownSearch(0,0)
4. Generate the transmission plan using search results of $u_{j,k}$

Function TopdownSearch(int j, int k) is defined as:

1. function TopdownSearch(int j, int k)
2. begin
3. if { $C_{j+1,2k} > b_0 / 2$ } TopdownSearch(j+1,2k)
4. else output $u_{j+1,2k}$
5. if { $C_{j+1,2k+1} > b_0 / 2$ } TopdownSearch(j+1,2k+1)
6. else output $u_{j+1,2k+1}$
7. end

Remark: Compared with the traditional *bottom-up* search algorithm, if the parent's buffer cost $C_{j,k}$ is strictly no less than the maximum of those of its child nodes, our top-down search algorithm can obtain the same results. This is because we use the Haar wavelet transform.

The WTS algorithm has the following property: In the single flow case, assume that PCRTT uses a binary search algorithm to find the suitable interval. WTS outperforms PCRTT. That is: compared with PCRTT, WTS not only keeps smaller number of rate changes, smaller peak rate, and small variability of bandwidth requirements, but also enforces the same lower bound on the time between two consecutive rate changes.

To see this, we note that WTS in general results in an unbalanced binary tree, which is part of the whole tree generated by PCRTT. The relationship between the WTS unbalanced binary tree and the performance of transmit plan can be summarized as below: 1) a lower resolution analysis results in a smoother transmit plan; 2) the number of rate changes is proportional to the number of leaf nodes in the resulting tree; 3) the lower bound between rate changes is determined by the depth of the tree. Thus, the WTS algorithm must outperform PCRTT under these metrics.

IV. SIMULATION RESULTS

We use full-length and constant-quality video clips from Feng's library [1] as testing sequences and compare WTS with PCRTT [4], MVBA [6] and MCBA [2] algorithms across a range of client buffer sizes, video clips, and performance metrics. For PCRTT, we choose the largest possible interval size in the simulation.

A. WTS of a single flow

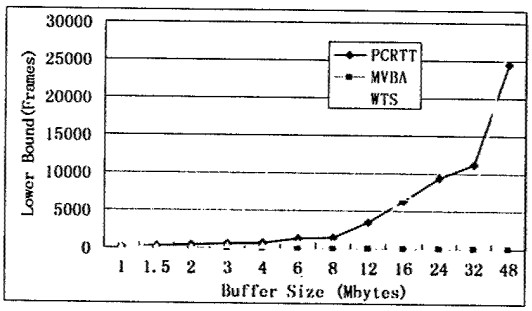
In the single flow case, all algorithms are applied to the same movie: *Jurassic Park* (M-JPEG). For a typical 2-hour video sequence with 216000 frames, the WTS and PCRTT algorithms require a few seconds of computation time on 400MHz PC in our experiment. But MVBA and MCBA algorithms needs several minutes under the same condition.

Fig. 3 shows the simulation results for single flow. Fig. 3(a) plots the lower bound on the time between rate changes versus buffer size. We can see from Fig. 3(a) that WTS algorithm achieves much lower bound on the time between rate changes than that of MVBA algorithm, which remains only 1 frame across the above buffer size range. Fig. 3(b) plots the number of rate changes. It can be seen from Fig. 3(b) that the WTS algorithm achieves much smaller rate changes than MVBA and PCRTT algorithms. Fig. 3(c) plots the coefficient of variation (see [1] for definition). Fig. 3(d) plots the peak bandwidth. From Figs. 3(c) and (d), we can see that peak rate and bandwidth variability of WTS are better than those of PCRTT for single flow case.

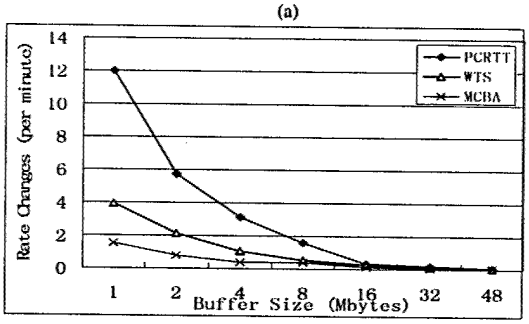
From these experiments, we find that, unlike MBVA and MCBA, WTS can enforce a lower bound on the time between rate changes. This property is important to simplify network resource management requirements since network cannot re-allocate resource at high frequency. It can be further seen from the simulation results that though PCRTT can also enforce a lower bound on the time between rate changes, compared with WTS, PCRTT is not very efficient to reduce rate variability since it only has one resolution.

B. Smoothing and multiplexing of multiple flows

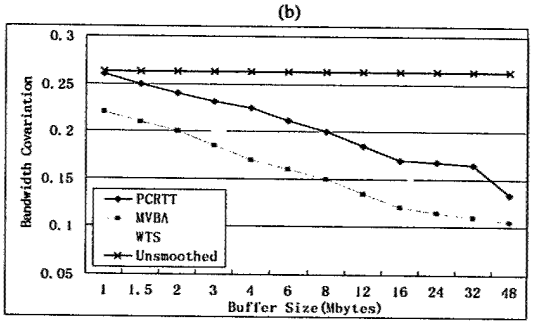
Since video clips are often multiplexed together in one link, it is important to study the performance of WTS for multiple-flow case. For this purpose, we do experiments to multiplex several clips together. Before multiplexing, each clip is smoothed using WTS and the starting points of each video stream are synchronized.



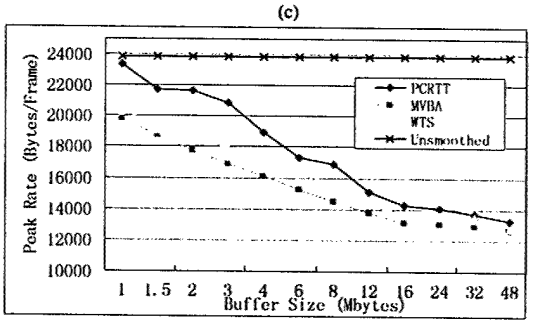
(a)



(b)



(c)

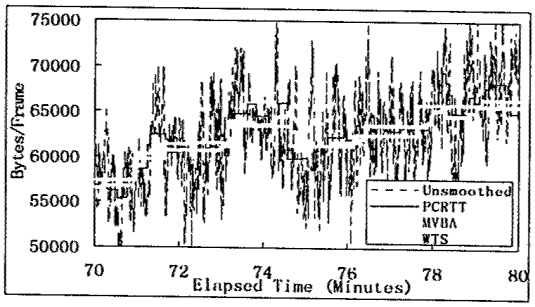


(d)

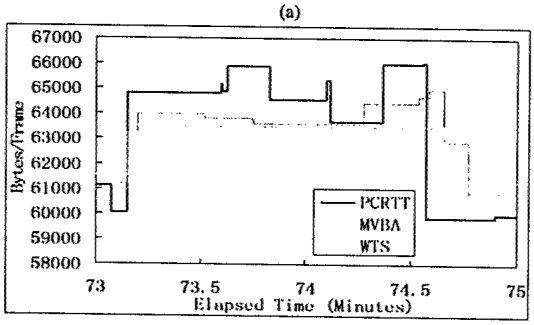
Fig. 3. Performance evaluation: single video.

Fig. 4 shows the multiplexed transmission plans generated by WTS, PCRTT, and MVBA algorithms, applying to five movies: Big, the Extra-Terrestrial (quality 100), Home Alone 2, Speed, and Rookie of the Year. In this case, we choose a moderate buffer size (4M bytes) for client buffer to smooth each video stream before multiplexing. We can see from Fig. 4 that WTS achieves much larger lower bound and much smaller number of rate changes than MVBA algorithms in the multiplexing case. Table 1 tabulates the performance of multiplexing using WTS, PCRTT, and MVBA smoothing algorithms. From Table1, we can observe that the number of rate changes of the multiplexed transmission plan of WTS is much smaller than that of MVBA or PCRTT algorithm. It can be further seen that the interval lower bound of WTS in the multiplexing case achieves much larger bound than that of MVBA.

These multiplexing experiments show that WTS is efficient for multiple-flow case. The multiplexing performance can be explained as follow: WTS and PCRTT only change rate at fixed interval, so the transmit plans can be synchronized when several clips are multiplexed together; and it is synchronization that makes WTS and PCRTT to achieve small number of rate changes for multiple-flow case.



(a)



(b)

Fig. 4. Multiplexed bandwidth plans.

In short, we can conclude that WTS and PCRTT achieve a balance between single flow performance and multiple-flow performance while MVBA and MCBA focus on single flow performance.

V. CONCLUSIONS

This paper proposes a novel wavelet-based traffic smoothing scheme for VBR streams. Our contributions can be summarized as follows: (1) unlike existing smoothing algorithms which only have one resolution, our scheme has multi-resolution property and can be viewed as multiresolution version of PCRTT. (2)WTS enforces a lower bound on the time between two consecutive rate changes while keeping small number of rate changes, small peak rate, and small variability of bandwidth requirement for both the single and multiple-flow case. (3) WTS has low computation complexity.

Table 1. Multiplexed performance

		Single video: *Big*	Multiplexing of 5 video	
			Smoothed	Unsmoothed
Number of rate change	PCRTT	97	594	169579
	MVBA	196	844	
	WTS	57	98	
Lower bound (Frames)	PCRTT	1743	855	1
	MVBA	1	1	
	WTS	1344	672	
Peak Rate (Bytes/ Frame)	PCRTT		82643	93115
	MVBA		76331	
	WTS		78560	
Variance	PCRTT		0.0765	0.1036
	MVBA		0.0678	
	WTS		0.0724	

ACKNOLEDGEMENT

Authors would like to thank Prof. Wu-Chi Feng from Ohio-state University for providing video testing sequences.

REFERENCES

[1] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video," *IEEE Trans. Multimedia*, vol. 1, pp. 302-312, Sep. 1999.

[2] W. Feng, F. Jahanian, and S. Sechrest, "Optimal buffering for the delivery of compressed prerecorded video," *ACM Multimedia Syst. J.*, pp. 297–309, Sept. 1997.

[3] W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video," *Proc. IS&T/SPIE Multimedia Networking and Computing*, pp. 58–66, Feb. 1997.

[4] J. M. McManus and K. W. Ross, "Video on demand over ATM: Constant-rate transmission and transport," *IEEE J. Select. Areas Comm.*, vol. 14, pp. 1087–1098, Aug. 1996.

[5] J. Zhang and J. Hui, "Applying traffic smoothing techniques for quality of service control in VBR video transmissions," *Comput. Comm.*, vol. 21, pp. 375–389, Apr. 1998.

[6] 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," *IEEE/ACM Trans. Networking*, vol. 6, pp. 397–410, Aug. 1998.

[7] J. Zhang and J. Hui, "Traffic characteristics and smoothness criteria in VBR video transmission," *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, June 1997.

[8] Weiping Li; Fan Ling; Xuemin Chen "Fine granularity scalability in MPEG-4 for streaming video," *Proc. ISCAS'00*, vol. 1, pp. 299 –302, 2000.

[9] N.G.Duffield, K.K.Ramakrishnan, and A. R. Reibman, "Issues of quality and multiplexing when smoothing rate adaptive video," *IEEE Trans. Multimedia*, vol. 1, pp.352-364, Dec. 1999.

[10] Z.-L. Zhang, J. F. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE J. Select. Areas Comm.*, vol. 15, pp. 1148-1166, Aug. 1997.

[11] R. Guirguis and S. Mahmoud, "Transmission of real-time multi-layered MPEG-4 video over ATM/ABR service," *Proc. ICC'00*, vol. 1, pp. 259 –263, 2000.