Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Segment-based traffic smoothing algorithm for VBR video stream*

LIU Yun-qiang†, YU Song-yu, WANG Xiang-wen, ZHOU Jun

(*Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200030, China*)

†E-mail: liuyunq@sjtu.edu.cn

Received Jan. 16, 2005; revision accepted Oct. 21, 2005

Abstract: Transmission of variable bit rate (VBR) video, because of the burstiness of VBR video traffic, has high fluctuation in bandwidth requirement. Traffic smoothing algorithm is very efficient in reducing burstiness of the VBR video stream by transmitting data in a series of fixed rates. We propose in this paper a novel segment-based bandwidth allocation algorithm which dynamically adjusts the segmentation boundary and changes the transmission rate at the latest possible point so that the video segment will be extended as long as possible and the number of rate changes can be as small as possible while keeping the peak rate low. Simulation results showed that our approach has small bandwidth requirement, high bandwidth utilization and low computation cost.

Key words: Segmentation, Traffic smoothing, Variable bit rate (VBR) video
doi:10.1631/jzus.2006.A0543 **Document code:** A **CLC number:** TP391

INTRODUCTION

Many emerging video applications, such as video-on-demand (VOD) and distance learning, rely on the efficient transfer of compressed video. For the same average bandwidth, variable bit rate (VBR) encoded video streams have higher perceivable quality and greater opportunity for statistical multiplexing gains compared to constant bit rate (CBR) video streams (McManus and Ross, 1996). However, VBR video typically shows significant burstiness on multiple time scales, due to the frame structure of the compression algorithm as well as natural variations within and between scenes (Grossglauser *et al*., 1997; Rexford and Towsley, 1999). VBR video can be difficult to deliver over networks because of the high peak and the large rate variation which increase the network resource requirements. The efficient transfer of VBR video requires effective techniques to handle burstiness.

To reduce the end-to-end resource requirements, video server can smooth the outgoing stream through transmission of frames into the client buffer in advance of each burst. Prior knowledge of frame lengths can be utilized to reduce the burstiness of resources required for transmission of stored video. Following this philosophy, many efficient bandwidth smoothing algorithms have been proposed for delivery of stored video over networks (McManus and Ross, 1998; Hadar and Cohen, 2001). These algorithms produce a server transmission schedule consisting of a series of CBR runs while guaranteeing no overflow or underflow of the playback buffer. The schedules simplify the allocation of resources in video servers and reduce bandwidth negotiations with the network (Feng and Rexford, 1999).

McManus and Ross (1998) proposed piecewise constant rate transmission and transport (PCRTT) algorithm that divides the video stream into fixed-size intervals to create a transmission schedule. The $O(n)$ algorithm forces a rate change after every segment, and is simple but generally requires large playback buffer capacity and introduces long delay before starting of playback. The algorithm results in periodic

*Project supported by the Hi-Tech Research and Development Program (863) of China (No. 2003AA103810) and the National Natural Science Foundation of China (No. 60332030)

rate changes.

Salehi *et al*.(1998) introduced an optimal smoothing algorithm, also called the minimum variability bandwidth allocation (MVBA) algorithm, which minimizes the variability of the bandwidth requirements. MVBA changes the transmission rate as early as possible for both rate increases and rate decreases to avoid making large changes, but sometimes at the expense of a larger number of small bandwidth changes.

Instead of minimizing the variability of bandwidth allocation, the minimum changes bandwidth allocation (MCBA) algorithm (Feng *et al*., 1997) performs a search operation on the frontier of each run to find the starting point of the next segment. The algorithm produces a transmission plan with the smallest possible number of rate changes as well as the minimum peak bandwidth requirement, but it results in the worst case complexity of the algorithm to be $O(n^2\log n)$.

Ye *et al*.(2004) presented a wavelet-based traffic smoothing (WTS) algorithm that performs reasonably well under several performance metrics (e.g., rate variability, number of rate changes, peak rate) while the algorithms mentioned above only optimize one or two metrics. The algorithm produces the best transmission schedule by searching a pruned subtree of a full binary tree, which corresponds to the original VBR video traffic. However, mapping a stored VBR video stream into a binary tree will impose limitation on the location of rate changes. Explicitly, a single run must change its rate on segment number that is a multiple of powers of two.

The paper provides a novel segment-based traffic smoothing scheme to transport stored VBR video. In the proposed scheme, a cost function associated with a transmission schedule of a segment is introduced to determine the segmentation boundary and the transmission bandwidth while guaranteeing the playback buffer does not underflow or overflow. The algorithm dynamically adjusts the segmentation boundary and changes the transmission rate at the latest possible point so that the video segment will be extended as long as possible and the number of rate changes can be as small as possible while keeping the peak rate low. Simulation results showed that our approach has small bandwidth requirement, high bandwidth utilization and low computation cost.

TRAFFIC SMOOTHING FOR STORED VBR VIDEO STREAM

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. Fig.1 is a diagram of transporting pre-stored video stream through networks to the client side is given. 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.

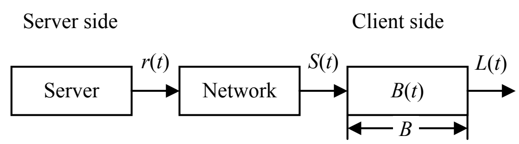


Fig.1 Transport of pre-stored video through networks

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 before actually being decoded, 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 f_i 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 tth frame, t=1, 2, …, N, it must have received at least $L(t)$ bytes from the server, where

$$L(t)=\sum_{i=1}^{t} f_i \quad (t=1,2,\cdots,N). \tag{1}$$

In the same way, a client should receive no more than

$$U(t)=B+\sum_{i=1}^{t}f_i \tag{2}$$

by frame time t to prevent playback buffer overflow. 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

$$L(t)\le S(t)\le U(t) \tag{3}$$

and $S(t)=B+\sum_{i=1}^{t}f_i$, where $r(i)$ is the transmission rate during frame slot i of the smoothed video stream.

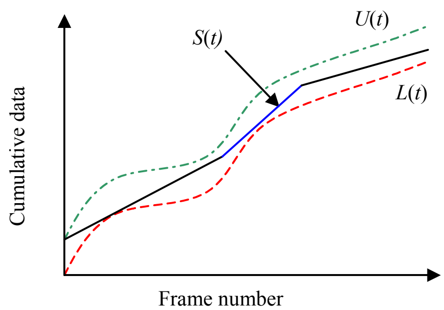


Fig.2 Transmission schedule

A bandwidth smoothing schedule draws a path from the beginning of the video to the end that stays within the region. The path consists of a series of linear segments, each with a constant transmission rate. Thus creating a bandwidth smoothing plan involves generating consecutive runs each with a constant bandwidth allocation and a duration based on the selected starting point and initial buffer occupancy. To a given video stream, there are many feasible transmission schedules. The various transmission schedules have different performance properties. A good transmission schedule is designed to minimize the peak bandwidth, the variability of the bandwidth allocations and the buffer size required to keep the network utilization as large as possible.

SEGMENT-BASED TRAFFIC SMOOTHING ALGOTRITHM

In this section, a segment-based traffic smoothing algorithm is proposed. We construct a feasible transmission schedule based on the following criteria. First, each CBR transmission segment should be as long as possible for CBR transmission in order to make the traffic as smooth as possible. Second, 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. Third, 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 as far away as possible 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. Our algorithm is also called segment-based traffic smoothing algorithm.

A cost function $C(j,k)$ associated with a transmission schedule of a segment is introduced, which represents the maximum client buffer requirement over interval $[j,k]$. Then we have Eq.(4):

$$C(j,k)=\max_{j\le t\le k}\left|\sum_{i=j}^{t}r(i)-f_i\right|, \tag{4}$$

where

$$r(i)=\sum_{t=j}^{k}f_t\Big/(k-j) \tag{5}$$

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.

Fig.3 illustrates the process of creating a schedule plan for a segment. The middle curve $M(t)$ between the upper curve $U(t)$ and the lower $L(t)$ equals $M(t)=(U(t)+L(t))/2=L(t)+B/2$. The slopes of these lines $L_1(t)$, $L_2(t)$, $L_3(t)$ correspond to the rates r_1, r_2, r_3 in the resulting transmission plans. In order to minimize the possibility of overflow or underflow, the starting point of the fixed-rate line in each interval is selected to be $B/2$. If r_1 is used as the transmission rate, the segment with the fixed transmission rate is very short. We should extend each CBR transmission segment as long as possible to make the traffic smoother. Using the rate r_3 as transmission rate makes $C(a,d)$ bigger than $B/2$ and results in client buffer starvation, and produces an unfeasible transmission plan. So the rate r_2 is selected as the transmission rate, which makes the segment as long as possible and avoids buffer underflow. Then a transmission schedule of the segment over $[a,c]$ is generated with a fixed transmission rate r_2. The next segment starts at time slot c+1. The process is then repeated until the entire video stream is segmented.

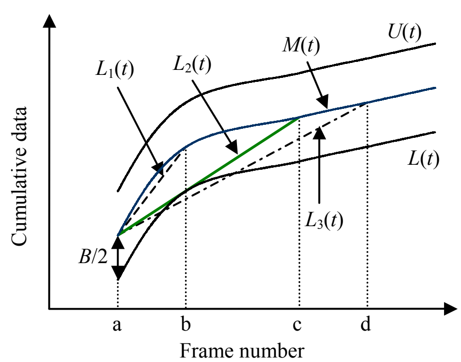


Fig.3 Construction of a segment

Based on the above analysis, Fig.4 shows the pseudo-code for our algorithm where s, e denote the starting point and terminal point of a segment respectively, c represents the cost of a segment and N is the length of the video in frames. The first segment starts at time slot 0. By comparing $C(s,e)$ with $B/2$, the scheme determines how to create a plan. If $C(s,e)<B/2$, extend the segment; otherwise, transmit the segment with corresponding bandwidth of the average rate of the segment and a new segment starts at the next time slot. The process continues iteratively until the entire video is segmented and the transmission schedule of the video stream is determined.

It can be shown that the segment-based smoothing algorithm has a searching complexity of $O(1+2+...+M)$ for a segment where M is the length of

```
Initialization: s=0; e=1; c=0;
while (e<=N)
{
  Calculate C(s,e) and the average rate r1 over interval [s, e];
  if(C(s,e)c<B/2)
  {
    r=r1;
    if(e==N)
      Output video segment of interval [s, e] with the rate r;
    e++;
  }
  else
  {
      Output video segment of interval [s, e−1] with the rate r;
      Start a new segment: s=e; e++;
  }
}
```

Fig.4 Pseudo-code of our traffic smoothing algorithm

the segment in frames. In the best case, it performs N searches when M equals 1. So the minimal computational complexity of our algorithm is $O(N)$ where N is the length of the video in frames. It performs one search for the segmentation when M equals N, the maximal computational complexity of our algorithm is $O(M(M-1)/2)=O(M^2/2)=O(N^2)$.

For a typical 90 min video sequence, our algorithm requires only several seconds of computation time on 1 GHz PC.

PERFORMANCE EVALUATION

This section presents a performance comparison of PCRTT, WTS and our algorithm based on a collection of performance metrics that includes the peak rate requirements, the variability of the bandwidth allocations, the number of bandwidth changes. The experiments are conducted using a video trace from the MPEG-encoded movie "Star Wars" with 40000 frames (about half an hour of video), which has a mean bit rate of 0.36 Mbps, a peak bit rate of 4.24 Mbps. By changing the client buffer size B, we evaluated the three algorithms. For PCRTT algorithm, we calculate the largest possible interval size given a client buffer size in the simulation.

Peak bandwidth requirements

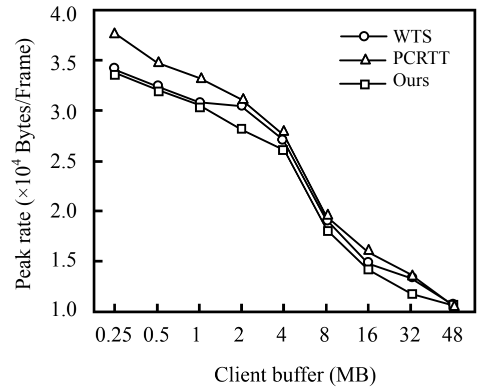
The peak rate of a smoothed video stream determines the peak bandwidth requirement across the network. Hence, most bandwidth smoothing algorithms attempt to minimize the peak rate. Fig.5a plots the peak bandwidth requirements for the three smoothing algorithms as a function of the client buffer size. Our algorithm results in smaller peak bandwidth requirements than the WTS and PCRTT algorithms for a given buffer size.

Number of bandwidth changes

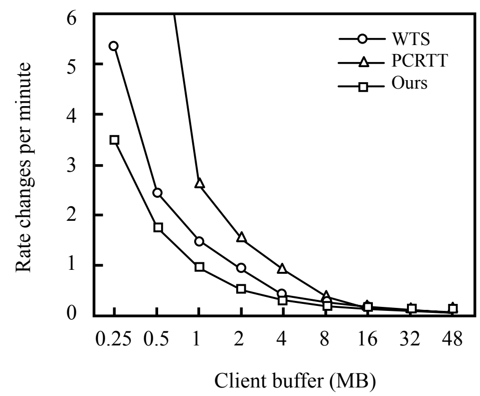
A rate change in a transmission plan allows server and network to reallocate resources based on the new rate. To reduce the complexity of the server and user sides, a bandwidth allocation algorithm could strive to reduce the number of bandwidth changes in the transmission schedule. It can be seen from Fig.5b that our algorithm achieves much fewer rate changes than WTS and PCRTT algorithms for a given buffer size.

Variability of the bandwidth allocations

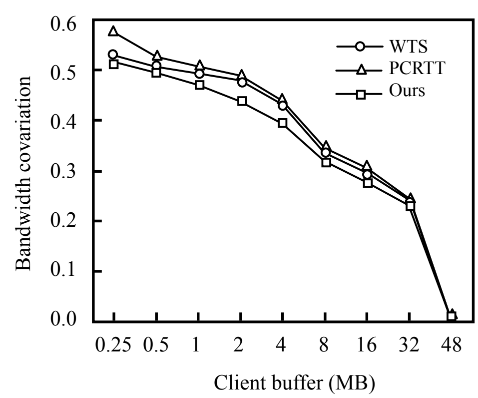
The variability of the bandwidth allocations determines the overall variability in the transmission rate requirements for the video stream. Plans with smaller rate variation have lower effective bandwidth requirements, allowing the server and the network to statistically multiplex the maximum number of streams. Fig.5c shows the coefficient of variation of the bandwidth allocations versus buffer capacity under the three schemes. Compared with PCRTT and WTS schemes, our scheme contributes to reducing the coefficient of variation of the bandwidth allocations.



(a)



(b)



(c)

Fig.5 Performance evaluation. (a) The peak bandwidth requirements; (b) The number of bandwidth changes; (c) The variability of the bandwidth allocations

CONCLUSION

This paper presents a novel segment-based traffic smoothing scheme for delivery of pre-stored VBR video stream to achieve efficient use of network resources. The proposed approach can dynamically decide the suitable segmentation point and get an optimal transmission schedule to smooth the VBR traffic. The performance of the proposed approach have been evaluated with an MPEG video trace and compared with two other schemes in terms of the peak band width requirements, the variability of the bandwidth allocations and the number of bandwidth changes. Experimental results showed that our approach has small bandwidth requirement, high bandwidth utilization and low computation cost.

References

Feng, W., Rexford, J., 1999. Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video. *IEEE Trans. Multimedia*, **1**(3):302-312. [doi:10.1109/6046.784468]

Feng, W., Jahanian, F., Sechrest, S., 1997. Optimal buffering for the delivery of compressed video. *Multimedia Systems*, **5**(5):297-309. [doi:10.1007/s005300050062]

Grossglauser, M., Keshav, S., Tse, D., 1997. RCBR: A simple and efficient service for multiple time-scale traffic. *IEEE/ACM Trans. Networking*, **5**(6):741-755. [doi:10.1109/90.650136]

Hadar, O., Cohen, R., 2001. PCRTT enhancement for off-line video smoothing. *Real-time Imaging*, **7**(3):301-304. [doi:10.1006/rtim.2001.0229]

McManus, J.M., Ross, K.W., 1996. Video on demand over ATM: Constant rate transmission and transport. *IEEE J. Selected Areas Commun.*, **14**(6):1087-1098. [doi:10.1109/49.508280]

McManus, J.M., Ross, K.W., 1998. A dynamic programming methodology for managing prerecorded VBR sources in packet-switched networks. *Telecommunication Systems*, **9**(2):223-247. [doi:10.1023/A:1019147923657]

Rexford, J., Towsley, D., 1999. Smoothing variable-bit-rate video in an internetwork. *IEEE/ACM Trans. Networking*, **7**(2):202-215. [doi:10.1109/90.769768]

Salehi, J.D., Zhang, Z., Kurose, J., Towsley, D., 1998. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Trans. Networking*, **6**(4):397-410. [doi:10.1109/90.720873]

Ye, D., Barker, J., Xiong, Z., Zhu, W., 2004. Wavelet-based VBR video smoothing. *IEEE Trans. Multimedia*, **6**(4): 611-623. [doi:10.1109/TMM.2004.830817]