VBR VIDEO DELIVERY USING MONOTONIC-DECREASING RATE SCHEDULING

H. L. Lai, L. K. Chen, Y. B. Lee

Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong, China

ABSTRACT

Variable-bit-rate (VBR) encoding is known to provide a more consistent visual quality compared to constant-bit-rate (CBR) encoding. However, the resultant bit-rate variations greatly complicated the allocation of disk and network bandwidth in a video-on-demand system. This paper investigates a new scheduling algorithm with monotonic-decreasing rate allocations for scheduling disk retrievals and data transmissions. This scheduler enables the use of a very computationally-efficient admission-control algorithm that can provide deterministic performance guarantee. Simulation results based on a large number of VBR video traces (307 DVD videos) show that the proposed scheduler can achieve over 84% disk efficiency despite long-range bit-rate variations within individual videos.

1. INTRODUCTION

Multimedia applications such as video-on-demand (VoD) are expected to consume a significant portion of the bandwidth in future broadband networks. Video compression algorithms such as MPEG introduce burstiness to the compressed video stream and this makes the allocation of disk and network bandwidth difficult. Researchers have done much work to reduce this burstiness [1-4]. This includes smoothing techniques at the encoder [5], feeding the network utilization to the encoder to control encoding parameters [6], and smart scheduling of video data transmission to minimize the magnitude or the number of bandwidth changes [7].

The previous studies mainly focus on the network aspect. By contrast, the problem of scheduling the retrieval of VBR video data from the disk storage and correspondingly the effect on other parts of the system such as client buffer has received relatively little attention. Consider existing disk scheduling algorithms [8-9]. The earliest deadline first (EDF) algorithm schedules the video data block with the earliest deadline to be retrieved first, whereas the SCAN algorithm schedules the read requests in rounds and read one block for each stream while the disk head scans back and forth across the disk surface. These algorithms are designed for either achieving the highest throughput or meeting all possible deadlines. Given the long-range bit-rate variations of VBR-encoded videos, it is very difficult to perform admission control to provide deterministic performance guarantee.

For example, consider a server using SCAN with fixed round length and variable block size for scheduling disk retrievals. Assume the server is serving S existing streams and a request for a new stream of length equivalent to R disk rounds arrives. To provide deterministic performance guarantee, the server can admit this stream only if none of the R rounds will be overloaded by the new stream. Consequently, the server will have to perform resource allocation for the R disk rounds, with up to $R(S+1)$ disk I/O time estimations. Given that a typical video server can serve over 100 streams and a movie can span over 5000 rounds, the amount of computations involved is substantial. Worst still, the server must be able to precisely estimate the I/O time in order to guarantee performance. Given that I/O time is a complex function of disk seek, head-switching, latency, etc., this will only further increase the computational complexity.

This study tackles the previous problems by a new scheduling algorithm with monotonic-decreasing rate allocations for scheduling disk retrievals and data transmissions. This scheduler enables the use of a very computationally-efficient admission-control algorithm that can provide deterministic performance guarantee, and at the same time achieves good disk efficiency.

In the next section, we present the new scheduling algorithm and in Section 3, we show the simulation setup and performance results. Finally, Section 4 concludes the paper.

2. DECREASING RATE SCHEDULING

A VBR encoded video stream of length L seconds can be divided into variable-sized video blocks according to the short-term bit-rate. Specifically, at time t, a video block of size $a[t]$ is required by the client for continuous video playback. The goal of the scheduler is to schedule the retrieval and delivery of such video blocks to ensure playback continuity without starvation at the client. Define $A(t)$ as the cumulative data consumption function:

$$A(t) = \sum_{\tau<t} a[\tau] \qquad (1)$$

Traditional smoothing algorithms [2] construct a bounding function:

$$B(t) = A(t) + b \qquad (2)$$

where b is the client buffer size.

To construct a schedule $S(t)$ with no buffer overflow nor underflow, the following condition must hold:

$$A(t) \le S(t) \le B(t) \qquad (3)$$

Many studies have been conducted to smooth this schedule $S(t)$ [1-4,7]. However, even after smoothing, part of the burstiness remains. As the efficient transmission of video data depends on the statistical multiplexing gain [10-11], server overload cannot be

0-7803-6685-9/01/$10.00©2001 IEEE

eliminated and one has to resort to statistical rather than deterministic performance guarantee. Last but not least, we observe that real-world VBR-encoded videos have vastly different long-range bit-rate variations, rendering the statistical multiplexing gain to be highly dependent on the video mix.

On the other hand, one can compute the complete schedule on-the-fly by computing the disk load in a round-by-round basis for the entire duration of the new video stream. However as discussed in Section 1, this admission process is computationally expensive.

We propose a new monotonic-decreasing scheduler (MDS) to address the previous problems. Specifically, MDS allows the retrieval and transmission schedule to be computed offline based on the actual video bit-rates. In computing a feasible schedule, rates are assigned in a monotonic-decreasing manner, with the first assigned rate being the highest, and each subsequent rate being lower. The beauty of MDS is that once a stream is admitted, its load offered to each subsequent disk rounds will only decrease. Hence if we can ensure the first disk round is not overloaded, subsequent disk rounds are also guaranteed to have no overload as well. In this way, admission of a new stream requires the computation of only one disk round, thereby greatly simplified the admission control process.

2.1 *N*-Rate Scheduler

We first present a general algorithm for off-line computation of the retrieval and transmission schedule. The algorithm assumes that the client has infinite buffer space. While this is clearly not true in practice, we show in Section 3 that the resultant buffer requirement is modest for real VBR-encoded videos, and also well within the capacity of a small hard disk, standard in most set-top boxes and computers.

To determine the schedule, we begin at the origin and scan through the data-consumption curve $A(t)$ to look for a point with the greatest slope when connected to the starting point (see Fig. 2). This is then defined as the first bit-rate reduction point and will be used as the starting point to repeat the procedures until the end of the video is reached. The resultant output is a number of (bit-rate, time) tuples, which defines the current bit-rate and the time for switching to the next bit-rate. Fig. 3 shows the pseudo code of this scheduler. Note that this scheduler does not introduce any additional startup delay, which is a direct consequence of the monotonicity property.

2.2 2-Rate Scheduler

The previous *N*-rate scheduler generates the schedule with the minimum buffer requirement. A side effect is that video block size to be read in each round decreases and can become so small near the end of the video such that disk efficiency is reduced. Moreover, we note that each rate reduction also requires network-bandwidth renegotiation to reduce the amount of bandwidth reserved. Since bandwidth negotiation requires a network control node to participate, too many renegotiations can overwhelm the network. To circumvent these problems, we can reduce the number of bit-rate changes in the schedule by trading off client buffer.

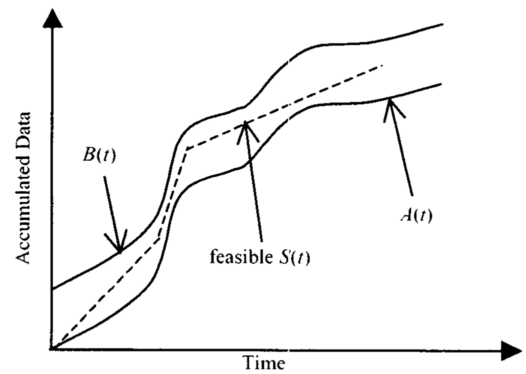

Figure 1. Constructing a feasible schedule for VBR video delivery.

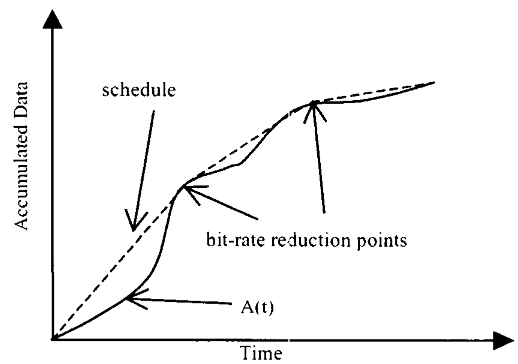

Figure 2. Constructing an *N*-rate schedule.

$\tau = 0, i = 0$

while $(\tau < L)$ {

$$r_i = \max_{t>\tau} \frac{A(t) - A(\tau)}{t - \tau}$$

$$T_i = t \ni \max_{t>\tau} \frac{A(t) - A(\tau)}{t - \tau}$$

$$\tau = T_i,\ i = i + 1$$

}

output : $(r_0, T_0), (r_1, T_1) \ldots (r_n, T_n)$

L is the length of the video in seconds,

r_i and T_i are the bit-rate and transition time for the i^{th} segment in the schedule.

Figure 3. Pseudocode for generating an *N*-rate schedule.

As an illustration, we study the simplest case of a 2-rate scheduler. Since the schedule can be constructed offline, we can use exhaustive search with a reasonably small step size to find the two bit-rates and the corresponding transition point that minimizes the client buffer requirement. We evaluate and compare these schedulers in the next section.

3. SIMULATION RESULTS

To obtain realistic performance results, we used a wide-variety of 307 different video titles for simulation. These are full-length, MPEG-2 encoded videos with bit-rate variations ranging from 0.408 Mbps to over 18.757 Mbps and an average length of 6120 seconds. Long-range (tens of minutes) bit-rate variations are common in these real MPEG-2 encoded videos.

3.1 Buffer Requirement

We first compute the client buffer requirements using the 2-rate and *N*-rate schedulers and summarize the results in Fig. 4. On average, the 2-rate scheduler requires 84.0 Mbytes of client buffer and the *N*-rate scheduler requires 76.5 Mbytes. As expected, more client buffer is required by the 2-rate scheduler. For both cases, 99% (305 out of 307) of the tested videos require no more than 400 Mbytes of client buffer. The two exceptions require 443/470 Mbytes and 593/609 Mbytes using the *N*-rate/2-rate schedulers respectively. Given that future set-top boxes are becoming entertainment centers equipped with web browsing, gaming and other multimedia services, a local hard disk is needed anyway and hence the buffer required for the proposed schedulers can easily be accommodated.

3.2 Disk Efficiency

To evaluate the disk efficiency achieved by the proposed schedulers, we simulated a system with 10 Quantum Atlas10K 9GB hard disks in a spindle synchronized disk array, storing 20 full-length videos that totals 90GB. The average video bit-rate is 5.856 Mbps and the sustained bit-rate of the hard disk drives specified by the manufacturer is 144 Mbps (so the sustained bit-rate of our simulated disk array is 1.44 Gbps). User requests are generated according to a Poisson process with mean inter-arrival time of 25 seconds. Each request selects a random video with uniform probability. The simulation is then run for a simulated time of one day and multiple simulation runs with randomly selected random seeds are averaged to obtain the final results.

Fig. 5 tabulates the performance of the 2-rate scheduler, the *N*-rate scheduler, the peak-rate scheduler which performs admission based on peak video bit-rates, the average-rate scheduler which performs admission based on average video bit-rates, and the brute-force scheduler which performs admission by computing the load of all future disk rounds affected by the new video stream. For simulation purposes, we ignore the computation time required by the admission process, and schedule all streams to start in the disk round right after the requests are received.

As expected, the peak-rate scheduler has the worst performance while the brute-force scheduler has the best performance. The proposed schedulers on the other hand, perform surprisingly well, with less than 1% of difference in disk efficiency compared to the brute-force scheduler. Given that the proposed schedulers are much more efficient in terms of admission control, the small trade-off in capacity is negligible. The performance of MDS is also comparable with that of the average rate scheduler, which achieved its efficiency at the cost of substantial startup delay (average 83 seconds, worst-case 1,353 seconds).

Finally, we compared the 2-rate scheduler to the *N*-rate scheduler. The total number of streams admitted by the 2-rate scheduler is larger, but the average number of simultaneous streams by the *N*-rate scheduler is higher. This is resulted from the shorter transmission time by the 2-rate scheduler than the original video length, and thus the average bit-rate is higher, 5.905 Mbps for our tested videos, compared to the original 5.856 Mbps. The difference in performance between the two schedulers is very small, showing that the 2-rate scheduler can be used as a good approximation for the general *N*-rate scheduler.

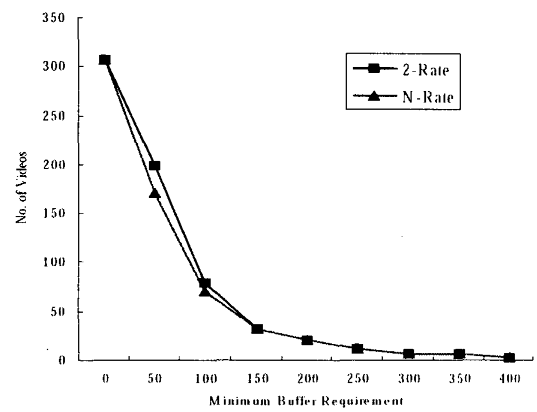

Figure 4. Distribution of buffer requirements for *N*-rate and 2-rate schedules (total 307 VBR videos tested).

	MDS 2-Rate	MDS *N*-Rate	Peak Rate	Avg. Rate	Brute Force
Concurrent Streams					
- Average	206.563	206.855	107.704	207.339	208.584
- Std Dev	3.754	3.715	1.588	3.807	2.802
- Max	215	215	112	215	220
- Min	194	192	101	192	196
Total Requests Served (out of 3468)	2977	2960	1564	2966	2980
Throughput (Mbps)	121.975	121.134	63.071	121.418	122.147
Efficiency *	84.71%	84.12%	43.80%	84.32%	84.82%

* Assuming the max achievable disk throughput is 144 Mbps.

Figure 5. Performance comparison of various schedulers.

4. CONCLUSION

By allocating disk and network bandwidth in a monotonic-decreasing manner, we addressed three challenges in VBR video delivery, namely minimizing admission control complexity, minimizing startup delay, and providing deterministic performance guarantee. Through extensive simulations using real-world VBR-encoded videos, the proposed schedulers are shown to achieve good disk efficiency that is comparable to the brute-force scheduler, which requires computationally-intensive admission control procedures; and also comparable to the average-rate scheduler, which requires substantial startup delay. The proposed monotonic-decreasing rate schedulers enable the use of a very simple admission controller, have zero start-up delay, and still provide deterministic performance guarantees. Given that local storage is likely to be abundant in future set-top boxes, the proposed schedulers provides an efficient solution for delivering high-quality, VBR-encoded videos in future video-on-demand services.

5. REFERENCES

[1] Dallas E. Wrege, Edward W. Knightly, Hui Zhang and Jörg Liebeherr, "Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical trade-offs", *IEEE/ACM Transactions on Networking*, pp.352-362, vol.4(3), 1996.

[2] James D. Salehi, Zhi-Li Zhang, James F. Kurose, and Don Towsley, "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing", *Proceedings of the ACM SIGMETRICS conference on Measurement & modeling of computer systems*, pp.222-231, 1996.

[3] Marwan Krunz, "Bandwidth Allocation Strategies for Transporting Variable-Bit-Rate Video Traffic", *IEEE Communication Magazine*, pp.40-46, Jan 1999.

[4] Simon S. Lam, Simon Chow and David K. Y. Yau, "A lossless smoothing algorithm for compressed video", *IEEE/ACM Transactions on Networking*, pp.697-708, vol.4(5), 1996.

[5] S.C. Liew and Derek C.Y. Tse, "A Control-Theoretic Approach to Adapting VBR Compressed Video for Transport over a CBR Communications Channel", *IEEE/ACM Transactions on Networking*, pp.42-55, vol.6(1), 1998.

[6] N. G. Duffield, K. K. Ramakrishnan and Amy R. Reibman, "SAVE: An Algorithm for Smoothed Adaptive Video Over Explicit Rate Networks", *IEEE/ACM Transactions on Networking*, pp.717-728, vol.6(6), 1998.

[7] W.C. Feng and J. Rexford, "Performance Evaluation of Smoothing Algorithms for Transmitting Prerecorded Variable-Bit-Rate Video", *IEEE Transactions Multimedia*, pp.302-313, Sept. 1999.

[8] D.J. Gemmell, H.M. Vin, D.D. Kandlur, P.V. Rangan, and L.A. Rowe, "Multimedia Storage Servers: A Tutorial", *IEEE Computer*, pp.40-49, vol.28(5), May 1995.

[9] H.J. Lee and D.H.C. Du, "The effect of disk scheduling schemes on a video server for supporting quality MPEG video accesses", *IEEE International Conference on Multimedia Computing and Systems '97. Proceedings*, pp.194-201, 1997.

[10] J. Mata, G. Pagan, and S. Sallent, "Multiplexing and Resource Allocation of VBR MPEG Video Traffic on ATM Networks", *IEEE International Conference on Converging Technologies for Tomorrow's Applications, Communications Conference Record, ICC '96*, pp.1401-1405, vol. 3, June 1996.

[11] D.T. Hoang and J.S. Vitter, "Multiplexing VBR Video Sequences onto a CBR Channel with Lexicographic Optimization", *International Conference on Image Processing '97*, pp. 369-372, vol.1, 1997.

[12] Greg Ganger, Bruce Worthington and Yale Patt, "The DiskSim Simulation Environment (Version 2.0)", *http://www.ece.cmu.edu/~ganger/disksim/.*