Rate control algorithms for video storage on disk based video servers

Zhourong Miao, Antonio Ortega
Integrated Media Systems Center, Signal and Image Processing Institute,
Department of Electrical Engineering-Systems,
University of Southern California,
Los Angeles, CA 90089, Email: {zmiao, ortega}@sipi.usc.edu

Abstract— **Emerging Video-on-Demand servers are equipped with large capacity and high speed disks, but the disk seeking time can be significant (in case when several users access different data simultaneously). Studies show different approaches to reduce the disk seek time in order to increase the number of users that can be served simultaneously. Most of them provide strategies for disk storage (disk placement) of the video data. Since most video data is encoded with Variable Bit Rate (VBR), using fixed-rate disk placement strategies may result in reduced quality (PSNR). In this paper, we will analyze the impact of disk placement strategies on the performance of VBR video stream. We then show that the performance can be improved by using rate control techniques that are aware of the disk placement constraints, so that the VBR video source stream can be optimized for those constraints. This optimization (which can be performed off-line) results is quality improvements of 0.5dB to 1.5dB in our experiments.**

I. Introduction

Video-On-Demand (VOD) services have been studied in the past few years and may soon become popular, as recording, storage and transmission of video data becomes less costly. The two challenging problems in a VOD system are video data transmission and disk storage. The output bit rate after compression is usually Variable Bit Rate (VBR),[1] but common transmission channels are based on the Constant Bit Rate (CBR) mode where the transmission bandwidth is fixed. Thus, transmission of VBR video data through a CBR channel may generate hiccups (i.e. frame loss) [5], [6], [8], [7].

In this paper, we first address the disk storage issue and, in particular, we study how video data can be encoded to achieve more efficient transmission. This will lead us to rate control algorithms optimized for specific disk placement strategies.

Since the size of video data is large, large capacity and high-speed hard disks are commonly used to store it. These modern hard disks have very high transfer bandwidth, and thus it is possible for a server to provide continuous video display to several users simultaneously. The disk drive can be multiplexed among several displays by providing *Round Robin Service* [2]. Each user is allocated a portion of one round interval to receive a block of data. This block of data should be sufficient for the user to display video until the next block arrives, otherwise a hiccup will occur. For a fair service, each user gets the same amount of compressed data in each service round. This can be modeled as a *CBR channel*. Hiccups can then be seen to occur if the decoder buffer underflows, given that the video data is VBR in nature.

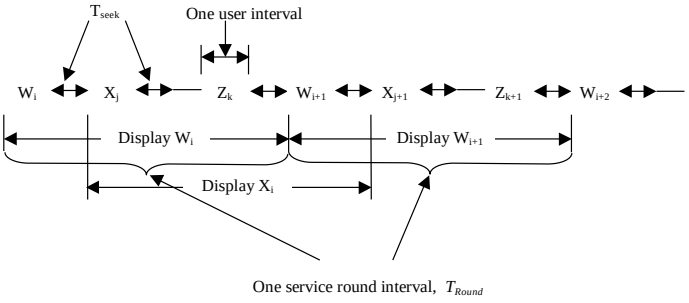

Fig. 1. Service Round

The situation gets worse if we take the disk seek-time into account. Studies show different approaches to reduce the seek-time by pre-arranging the video objects on the hard disk [2], [3], [4], [5]. Most of these approaches *partition the disk and place the video data blocks on the disk according to the display sequence*. Thus, the location of the data blocks is restricted so that only a small region needs to be searched. If blocks are placed in a more restrictive sequence (e.g. zigzag mode [2]), the seek-time will be reduced to close to zero. The resulting T_{round} is nearly constant with less overhead caused by T_{seek}, which reduces the complexity of software design of the server system and improves its performance.

These disk placement algorithms target the efficiency[2] of VOD servers, but they may affect the servers' ability to provide continuous display. For example, some data placement algorithms, may not allow random access because the location of the data blocks has been restricted. Thus it will be more difficult than that in a pure random placement approach to reduce the *duration* of hiccups. This is because, once the maximum number of users N is set, the block size $B_{size} = T_{round}/N$ is also fixed (transmitted during each

[1]The variable nature of the bit rate per frame comes from the fact that frames, when compressed to achieve a specific visual quality, require a different number of bits depending on such factors as the number of objects in the frame, the motion, the proportion of textured areas to flat areas, etc.

[2]i.e., they try to maximize the number of users that can be served simultaneously.

T_f	Period during which a frame is displayed, e.g. $1/30second$
f_p	Number of pre-fetched frames
T_p	Period during which the pre-fetched frames are displayed, $T_p = f_p \times T_f$
T_{round}	Turn-around-time for one service round
F_i	i-th frame
B_{disk}	Disk bandwidth (bits/second)
N	Maximum number of users
B_{user}	The average bandwidth per user $B_{user} = B_{disk}/N$
C_{acc}	Accumulated channel rate
R_{acc}	Accumulated frame rate
Buf_{max}	User buffer size
C_k	Channel rate at time $t = kT_f$
$x(i)$	quantization step for frame i
$R_{x(i)}(i)$	Number of bits for frame i coded with quantization step $x(i)$
$D_{x(i)}(i)$	Distortion of frame i coded with quantization step $x(i)$
N_f	Total number of frames
B_{size}	Block size in disk placement algorithms
R_N	Number of total service rounds
$N_B(i)$	Number of frames in block i

TABLE I
NOTIFICATION

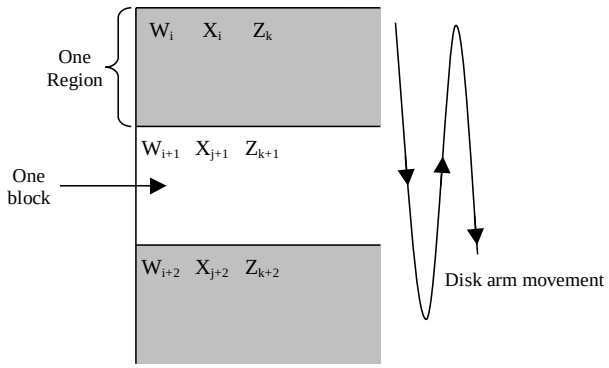

Fig. 2. Disk placement

T_{round}) . Thus, after the *disk placement* algorithm has been applied, it is *not* possible for the disk arm to reach a block at an arbitrary location, because the disk arm moves in a single direction (either towards the edge or the inside). If a block can not be displayed for T_{round} long, the result is that a user *must wait for the next service round* to get the next block. The server may have to "pause" service to other users by spending extra time on one particular client to reduce its hiccups.

Studies [5], [6] show different strategies to reduce the hiccups, such as restricting the number of concurrent users based on the QOS. Other approaches involve making a decision on whether to admit a new user based on the probability of hiccups if that user is admitted. Some of these admission algorithms may be too restrictive, and while they may prevent hiccups, they may also waste bandwidth during some service rounds (when all the users request small size blocks). Conversely, a less demanding admission policy may result in more hiccups.

In this paper, we tackle this problem from the encoder view point. We encode the video data with the given constraints set (B_{disk}, T_f, N) to guarantee continuous display. Admission of a new user will be simple, we just need to check to see if the total number of users exceeds the maximum number allowed (N). Since N is used to calculate the bandwidth per user applied in the rate control optimization, we can guarantee that no hiccups will occur if the number of users is less than N. This approach could also be incorporated with other algorithms mentioned above, if we allow limited hiccups during the display.

The organization of this paper is as follows. Section II presents the formulation of the problem. Section III describes a *Multiple Lagrange Multiplier Algorithm* to obtain the optimal solution, and Section IV provides the experimental results and conclusions. Our results show that the overall PSNR with rate control can be improved by 0.5 to 1.5 dB as compared to not using rate control under the constraints of continuous displaying.

II. Problem Formulation

User buffer constraints: We assume that the video frame rate is constant, and is the same at both the servers and clients. In a real-time video transmission system, the end-to-end delay interval must be constant, say ΔT seconds. Thus a frame encoded at time t, must arrive at the decoder (user) before $t + \Delta T$. Although the VOD is not a "fully" real-time video transmission system (the video source is already encoded before transmission and stored onto the disk), the delay constraint is the same as before. Even if all the data is available before transmission, the size of the user buffer and initial latency should be small (i.e assume the user can only pre-fetch f_p frames). Real-time transmission constraints are still applicable in VOD systems, because once we start displaying frames, we need to continuously transmit frames, and the number of frames that can be stored in the decoder buffer is limited (e.g. the whole sequence cannot be stored in the buffer).

End-to-end Delay: Typically, the frame rate is *30 frames/sec*. The pre-fetched f_p frames can be displayed for $T_p = f_p/30$ *seconds* long. If a frame is scheduled to be displayed at time t during the playback, it should arrive at the user end before $t + T_p$.

Rate constraint: The delay and buffer constraints can be converted into rate constraints, which the encoder has to meet to prevent hiccups. We assume that T_f is the period each frame can be displayed (typically, $1/30second$), which is also the basic time unit. Each frame is labeled with index $i, i = (1, 2, 3 \ldots)$, and if we start display at time $t = 0$, the frame F_i will be scheduled to display at time $t = i \times T_f$. R_i is the number of bits of frame F_i.

The channel rate is the disk bandwidth allocated to each user after time multiplexing. According to Table-1, each

user will get $B_{user} = B_{disk}/N(bits/sec)$ of bandwidth on average. That is, each user will receive bandwidth B_{disk} during a fraction T_{round}/N of each service round. The constraint for *no buffer underflow* is that any frame F_i must arrive at the user no later than $t = i \times T_p$. This requires that the channel have enough capacity to transmit all the frames $(F_1, F_2 \ldots F_i)$ to the user before time $t = i \times T_p$. This leads to the following constraints:

$$Size_{pre-fetch} = \sum_{i=0}^{f_p} R_i < Buf_{max} \tag{1}$$

$$T_f(\sum_{t=0}^{R_N} N_B(i) + f_p) \geq t = nT_f, n = 1, 2, 3... \tag{2}$$

$$T_f(\sum_{t=0}^{R_N} Size_{block}(i) + f_p - \sum_{i=0}^{t} R_i) \leq Buf_{max} \tag{3}$$

$$Size_{block}(i) \leq B_{disk} \times T_{round}/N \tag{4}$$

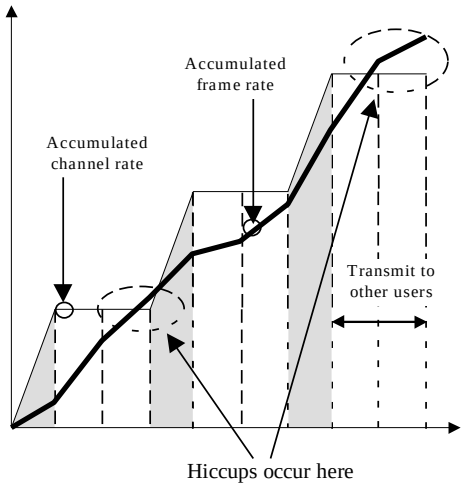

Fig. 3. Accumulated channel rate and frame rate

If different video sequences are stored on the disk, and different users request different sequences randomly, buffer underflow still can be avoided. This is because each sequence is encoded with rate constraints, and each of them will meet the requirement of non-buffer-underflow. These constraints are still met for each user even if the sequences are requested simultaneously.

To simplify (1-4), denote C_k as the channel rate at time $t = kT_f$. The condition for i-th frame, to arrive at the decoder in time for decoding is that all the data corresponding to i-th frame, as well as to all the previous frames, has to be transmitted before t_i. Thus:

$$\sum_{k=1}^{i} R_{x(k)}(k) \leq \sum_{k=1}^{i+f_p} C_k, \qquad i = 1, 2, 3, ..., N_f \tag{5}$$

where $x(k)$ is the quantizer assigned to the i-th frame. We use *Accumulated channel rate (C_{acc})* and *accumulated frame rate (R_{acc})* to describe the problem more clearly. We can re-write the above rate constraints function (with pre-fetch):

$$R_{acc}(i) = \sum_{k=1}^{i} R_{x(i)}(i), \qquad C_{acc}(i) = \sum_{k=1}^{i} C_k, \tag{6}$$

$$R_{acc}(i) \leq C_{acc}(i) + \sum_{k=1}^{f_p} C_k, \tag{7}$$

$$where \;\; C_k = \begin{cases} B_{disk}, & nT_{round} \leq k < nT_{round} + T_{user} \\ 0, & otherwise \end{cases}$$

Here R_{acc} is simply the accumulated bits of all the frames that have bee sent. C_k equals to B_{disk} while the server is transmitting data to that user; it is zero while the server is serving other users. We assume there are no constraints introduced by the network bandwidth here. The number of bits per frame is based on the choice of quantization step size. In this paper, one frame is assigned a single quantization step (using intra-frame mode).

In real-time playback, lost frames will cause visual distortion. For our scenario, hiccups mean the user has to wait for the next frame, while the current frame is frozen on the screen. Thus, it may be preferable to encode a frame with fewer bits if that allows us to avoid hiccups. Although the distortion of this frame would be increased, it is better than displaying nothing in certain cases. It is hard to measure the perceptual distortion due to frame losses. Thus in order to develop the cost function below, we do not allow any frame loss in our formulation[3]. The quantization step could be chosen at the encoder end before the video streams are stored onto the disk. The problem can be formalized as follows.

Given a set of constraints (as in [7]), how do we choose the quantization step size for each frame while minimizing the total distortion. To encode N_f frames, using a given set Q of M admissible quantizers, such that, for each choice of quantizer $j = x(i)$ for a given block i, we incur a distortion cost $D_{x(i)}(i)$ while requiring a certain rate $R_{x(i)}(i)$. The objective is to *find the optimal quantizer choices $x^* \in \chi = Q^N$, for a given channel rate C_k as in (7), such that:*

$$x^*(1, ..., N_f) = \text{argmin} \sum_{i=1}^{N_f} D_{x(i)}(i) \tag{8}$$

subject to the constraint set (5) or (7). We will solve this problem using multiple Lagrange Multipliers.

III. Optimization based on Multiple Lagrange Multipliers

Using Lagrangian optimization for rate control under multiple rate constraints was previously studied in [7], [9]. In that approach, the constrained optimization problem above is equivalent to the unconstrained problem derived

[3]It is possible to develop other cost functions which may take account frame losses. This is beyond the scope of this paper.

by introducing a non-negative Lagrange multiplier λ_i associated with each constraint in (6). The optimization formulation then becomes: *find the quantizer choice x^* at the time $t_i = iT_f$ such that:*

$$x^*(1, N_f) = \arg\min_{x} \sum_{i=1}^{N_f} D_{x(i)}(i) + \sum_{j=1}^{N_f} \lambda_j \Big(\sum_{i=1}^{j} R_{x(i)}(i) \Big), \quad (9)$$

We introduce N_f Lagrange multipliers to replace the N_f constraints in equation (7). To find the *optimal* quantizer set $x^*(1, N_f)$ is the same as to to find the appropriate multipliers $\{\lambda_i\}$ to meet the constraints. From [8], we can introduce another set of multipliers $\lambda'_i = \sum_{j=i}^{N_f} \lambda_j, (i = 1, 2, ...N_f)$ to rearrange (9) as:

$$x^*(1, N_f) = \arg\min_{x} \sum_{i=1}^{N_f} \big(D_{x(i)}(i) + \lambda'_i R_{x(i)}(i) \big), \quad (10)$$

Finding the solution for (9) is equivalent to finding the appropriate non-negative values of the set $\{\lambda'_i\}$. Define $J_i\big(\lambda'_i, x(i)\big)$, the cost for frame i, as:

$$J_i\big(\lambda'_i, x(i)\big) = D_{x(i)}(i) + \lambda'_i R_{x(i)}(i), \quad (11)$$

If we use intra-frame mode, the quantizer for each frame can be chosen independently while minimizing the cost for each block $J_i\big(\lambda'_i, x(i)\big)$ as:

$$x^*(i) = \arg\min_{x(i) \in Q} J_i\big(\lambda'_i, x(i)\big), \quad \forall i \in \{1, 2, ..., N_f\} \quad (12)$$

In [7], [8] a similar problem is solved by iteratively increasing the lower bounds on the multipliers, defined as $\{\Lambda'_i\}$, such that the violation of rate constraints can be prevented, and adjusting the values of $\{\lambda'_i\}$ until an optimal bit allocation, where none of the constraints is violated, is found. The details of the search for these multiple Lagrange multipliers can be found in [7], [8], [1]. Here we outline the basic procedures.

Step 1: Initially the quantizer choices $\hat{x} = \{x(1), x(2), ..., x(N_f)\}$ are obtained by using a single Lagrange multiplier λ'_{N_f} for all the frames in (12), subject to only one constraint: $\sum_{k=1}^{N_f} R_{i,x(i)} \leq \sum_{k=1}^{N_f + f_p} C_k$.

Step 2: If $\hat{x}$ is such that all rate constraints in (2) are met, then $\hat{x}$ is the optimal solution x^* for problem (8). Otherwise, assume that frame v is the last frame which violates the rate constraint, that is, $v < N_f$ and no other frame between frame $v+1$ and frame N_f violates the rate constraint. Find the minimum value of Lagrange multiplier $\Lambda'_v = \min \lambda'_v$ for the video stream from frame 1 to frame v which prevents violation of the rate constraint: $\sum_{k=1}^{v} R_{i,x(i)} \leq \sum_{k=1}^{v+f_p} C_k$.

Step 3: Find the quantizer choices $\hat{x} = \{x(1), x(2), ..., x(N_f)\}$ as in Step 1 except that the Lagrangian multiplier for the video streams from frame 1 to frame v is lower-bounded by Λ'_v as $\lambda'_v \Leftarrow \max(\Lambda'_v, \lambda'_v)$.

Step 4: Go to Step 2. Repeat until all the rate constraints in (2) are met.

IV. Experimental Results

In order to test our proposed algorithms, we simulated the transmission behavior with and without the rate control. We use 5000 and 10,000 frames from the movie "Mission Impossible" for our simulation. Each frame was encoded in intra-frame mode and we use 7 different quantization steps encoded by JPEG, thus generating 7 source streams with different rate-distortion performance. Each source stream uses a fixed quantizer. We test with different parameters for B_{disk}, T_{round} and N.

Fig. 4 shows the accumulated channel and frame rate (with and without rate control). The rates are added up from the time the video transmission starts. Curve (1) is accumulated channel rate ($C_{(acc)}$) which is the upper bound of the frame rate. The other curves (2-3) are accumulated frame rate ($R_{(acc)}$). The hiccups will occur if $R_{(acc)}$ exceeds the $C_{(acc)}$. Among curves (2-3), curve (2) is closest to the bound (1), which is based on rate control processing (Lagrange iteration). Curve (3,4) use fixed quantization steps, with a smaller quantizer step size for curve (4) (high frame rate, low distortion), and a larger one for curve (3). Those two quantization step are the closest two consecutive steps size of all the available steps.

The channel rate is the disk bandwidth, the block size is a typical size from certain disk placement algorithms. These two parameters decide the shape of curve (1), the upper frame rate bound. It shows that with a smaller fixed quantization step, there are more hiccups, while with a larger one, the channel capacity is wasted.

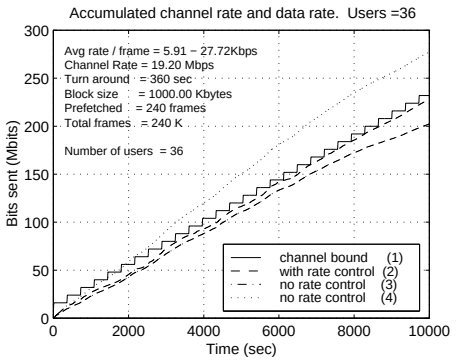

Fig. 4. Accumulated channel rate and frame rate

After applying the rate control over the whole sequence, a different quantization step can be chosen for each frame, and the accumulated frame rate can be set very close to the channel bound without exceeding it. We compared the PSNR with the distortion using rate controls with the frames of fixed quantization (its rate also not exceed the channel bound). Since we do not know how to measure the perceptual distortion of a lost frame, the comparison is made based on that no hiccups occur for any choice of

the quantizers (with or without rate control). Based on our experimental result, we will have about 0.5 to 1.5 dB for overall frame sequences. Of course, if there are less available quantization steps, we can get larger PSNR gain, and small PSNR gain vice versa[4].

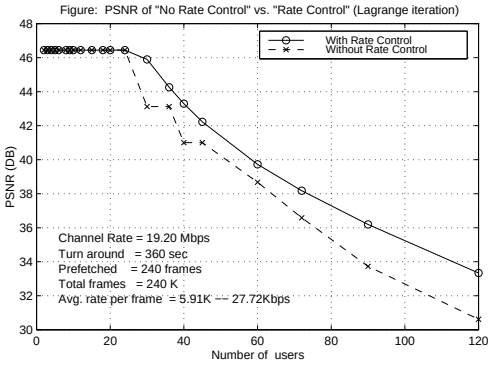

Fig. 5. Overall PSNR

For the other choices of quantization steps, we show the total hiccups and average waiting time between hiccups. As estimated, when the number of designed maximum user increases, more hiccups will occur. Also the average waiting time (the frame-freeze time) for next video block will also increase.

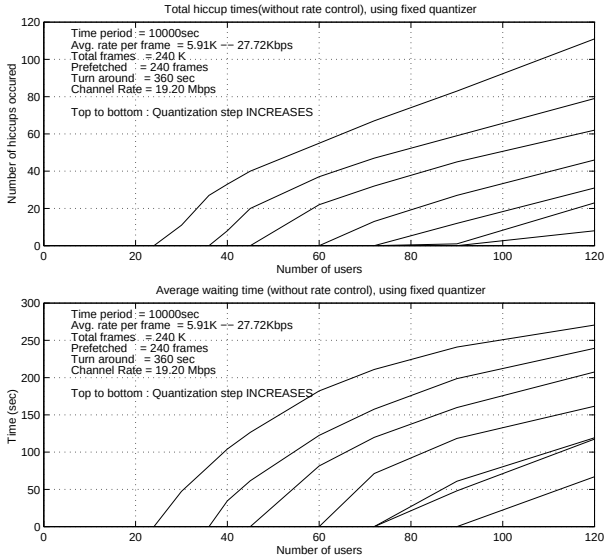

Fig. 6. Average waiting time and hiccups without rate control

References

[1] G. M. Schuster, A. K. Katsaggelos, *Rate-Distortion Based Video Compression*, Kluwer Academic Publishers.

[2] S. Ghandeharizaheh, S. H. Kim, C. Shahabi, "On Configuring a Single disk Continuous Media Server", *in Proceedings of the ACM SIGMETRICS/PERFORMANCE, May 1995.*

[3] D. W. Brubeck, L. A. Rowe, "Hierarchical storage Management in a Distributed VOD system", *in IEEE, Multimedia, pp. 37-47, Fall 1996.*

[4]If the set of available quantization scales is small than it will be more likely that a solution which does not violate the rate constraints is far below the bound.

[4] E. Chang, H. Garcia-Molina, "Reducing Initial Latency in Media Servers", *in IEEE, Multimedia, pp. 50-61, Fall 1997.*

[5] E. Chang, A. Zakhor, "Disk-Based Storage for Scalable Video", *in IEEE Trans. On Circuits and Systems for Video Technology, Vol. 7, NO. 5, pp. 758-770, Oct. 1997.*

[6] D. Makaroff, G. Neufeld, N. Hutchinson, "An Evaluation of VBR Disk Admission Algorithms for Continuous Media File Servers", *in ACM Multimedia '97, pp. 143-153, Seattle Washington.*

[7] A. Ortega, "Optimal rate allocation under multiple rate constraints", *in Data Compression Conference, Snowbird, Utah, Mar. 1996.*

[8] C. Y. Hsu, A Ortega, M. Khansari, "Rate Control for Robust Video Transmission over Burst-Error Wireless Channels". *Accepted for publication, IEEE JSAC Special Issue On Multimedia Network Radios, July 1998*

[9] J-J. Chen, D. W. Lin, "Optimal bit allocation for coding of video signals over ATM networks", *in IEEE JSAC, vol. 15, pp. 1002-1015, Aug. 1997.*