

## **JOINT ENCODER AND VBR CHANNEL OPTIMIZATION WITH BUFFER AND LEAKY BUCKET CONSTRAINTS \***

Chi-Yuan Hsu and Antonio Ortega

Signal and Image Processing Institute  
Dept. of EE-Systems  
University of Southern California  
Los Angeles, CA 90089

### **INTRODUCTION**

Video sequences present varying degrees of complexity and thus, when encoded with most practical compression algorithms at a fixed quality level, generate a variable bit rate (VBR) sequence. In order to transmit such a VBR sequence over a constant bit rate (CBR) channel, a buffer is used to absorb the variations in source bit rate, and a rate control algorithm is needed to prevent buffer overflow. The size of the buffer depends on the total end-to-end delay allowed between encoder and decoder (excluding channel transmission delay). A larger buffer size can allow the encoder to perform better in terms of distortion with the same available channel rate, but such improvement is obtained at the price of larger end-to-end delay.

To improve the video quality without increasing the end-to-end delay one can resort to VBR transmission. For example, VBR video transmission has been proposed and implemented for datagram networks, e.g. those forming the Internet (see [1] [2] for examples of rate control in this scenario.) In these networks, the channel rate depends on the network conditions and there is no guarantee on the maximum delay and channel bandwidth available to the user. This could be a critical shortcoming when, as is the case with video, real time data is being transmitted. We thus concentrate on video transmission over Asynchronous Transfer Mode (ATM) networks where we assume that the quality of service can be guaranteed by the network if the encoder complies with a traffic “contract”, established at the

---

\*This work was supported in part by the National Science foundation under award MIP-9502227 (CAREER)

connection set up stage. In the ATM environment the video encoder can now select the number of bits to be sent to the network. Our goal is to study the potential benefits of *rate control algorithms which can adjust both the encoder rate and the channel rate*. The choice of channel rate cannot be arbitrary since it will be constrained by the agreed upon traffic contract, and monitored through some network policing function [3]. Therefore, both the network policing function and the physical buffer sizes will restrict both the encoding rate and transmission rate [4].

While previous work, [4], has considered the buffering constraints in the VBR case, in this paper we introduce the concept of *effective buffer size* which establishes the link between the channel rate and the buffer size required to prevent data loss in VBR transmission. Using the effective buffer size makes it simple to use our knowledge of the CBR case to better understand the VBR case. Given the constraints imposed by the effective buffer size and the network policing function, we introduce new techniques that allow joint optimization of the choice of encoder and channel rates. We focus on optimal techniques that will provide bounds on achievable performance and can also be used in off-line compression environments. Based on our results we can derive conclusions that are applicable to simpler coding scenarios.

We start by discussing the bit rate constraints imposed by the end-to-end delay and channel rate in the CBR and VBR channel environments. The encoder optimization in CBR channel by dynamic programming [5] is briefly reviewed. We then formalize the joint optimization problem in the VBR case and extend the optimization techniques used in the CBR to find the optimal solution. Finally, we provide some experimental results.

## BUFFER CONSTRAINTS IN CBR AND VBR CHANNEL

When a VBR source is transmitted through a CBR channel, buffers are used at both encoder and decoder to absorb the variations on the bit rate produced by the encoder. For a fixed total bit budget available to code a video sequence, the encoder can achieve less total distortion if a larger buffer size is used, but the size of buffer cannot be arbitrarily large due, as will be seen, to end-to-end delay considerations.

In a video communications system, the end-to-end delay,  $\Delta T$ , is the time it will take for one frame to be transmitted. It can be written as

$$\Delta T = \delta t_e + \delta t_c + \delta t_d, \quad (1)$$

where  $\delta t_e$  and  $\delta t_d$  are, respectively, the delays in encoder and decoder buffer, and  $\delta t_c$  is the channel transmission delay. The total end-to-end delay  $\Delta T$  has to be constant. Although the channel delay can be variable (this corresponds to the delay jitter in networking environments) we assume that the delay variations are relatively small compared to the other delay components and thus assume that the sum of delay in encoder buffer and decoder buffer  $\delta t_e + \delta t_d = \Delta T - \delta t_c$  is also constant. We will use this combined buffer delay as the end-to-end delay parameter in system design. Here we consider the frame interval as our basic time unit, but similar analyses could be derived for smaller time intervals (e.g. down to block level). Assume the length of the time interval for one video frame is  $\mathcal{T}$ . Given  $\Delta T$ , the end-to-end delay due to buffering (expressed in number of frames rather than in seconds) is

$$\Delta N = \frac{\Delta T - \delta t_c}{\mathcal{T}}. \quad (2)$$

The encoder has to keep both encoder and decoder buffers from overflowing to avoid the ensuing loss of data. As for underflow, while encoder buffer underflow can be easily avoided by bit-stuffing, decoder buffer underflow is a much more serious problem since it means that the coded data in the encoder buffer is not arriving to the decoder within the required end-to-end delay. If decoder underflow occurs the corresponding video frame is considered lost. Note that data loss due to decoder buffer underflow may occur *even if the physical size of the encoder buffer is large enough to hold all the coded data*.

We now describe how decoder buffer underflow can be prevented by the encoder. To do so we introduce the concept of *effective buffer size*,  $B_{eff}$ , which we define as *the maximum level of buffer occupancy that the encoder can reach without violating of end-to-end delay constraint* (i.e. without producing decoder underflow.) Note that the effective buffer size may be smaller than the physical buffer size that is available at the encoder.

### Effective Buffer Size in CBR Channels

Before tackling the VBR case, we consider the effective buffer size in the CBR channel case. Suppose  $B^e(i)$  and  $B^d(i)$  are, respectively, the encoder buffer and decoder buffer occupancies at time  $i$  (i.e. during the  $i$ -th frame interval.)  $C$  is the channel bit rate in one frame interval and  $R(i)$  is the number of bits used by the encoder to code the  $i$ th frame of video. If neither encoder or decoder buffers ever overflow or underflow, then the encoder and decoder fullness at time  $i$  will be:

$$B^e(i) = \sum_{j=1}^i R(j) - i \cdot C \quad (3)$$

$$B^d(i) = \begin{cases} i \cdot C - \sum_{j=1}^{i-\Delta N} R(j), & \text{when } i \geq \Delta N, \\ i \cdot C, & \text{when } i < \Delta N, \end{cases} \quad (4)$$

where  $\Delta N$ , the end-to-end delay, is also the time that the decoder waits after receiving the first bits of information before starting to decode. (As can be seen, data is removed from the decoder buffer only after  $i = \Delta N$ .) By combining the encoder buffer occupancy (3) at time  $i$  and decoder buffer occupancy (4) at time  $i + \Delta N$ , we can get

$$B^d(i + \Delta N) = \Delta N \cdot C - B^e(i) \quad \text{when } i \geq \Delta N \quad (5)$$

If we can keep  $\Delta N \cdot C - B^e(i) \geq 0$  all the time, then  $B^d(i + \Delta N)$  will always be greater than 0 and therefore the decoder will not underflow. We can thus define the effective buffer size in the CBR case as

$$B_{eff} = \Delta N \cdot C \quad (6)$$

and we can guarantee that if the encoder buffer fullness  $B^e(i)$  is always smaller than  $B_{eff}$ , then the decoder buffer will not underflow. From (5) we also know that the maximum value of encoder buffer occupancy,  $B^e(i)$ , will be  $\Delta N \cdot C$  when  $B^d(i) = 0$ . Therefore for the encoder, a buffer size as large as  $\Delta N \cdot C = B_{eff}$  will be enough to hold all possible coded data within the end-to-end constraint. A similar argument can also apply to the decoder buffer.

If the physical buffer size is smaller than  $B_{eff}$ , then there will be more constraints on the buffer occupancy besides  $B_{eff}$ . The buffer occupancy will then be upper bounded by the

physical buffer size as  $B^e(i) \leq B^e$   $B^d(i) \leq B^d$ .

### Effective Buffer Size in VBR Channels

There is a detailed discussion on the buffer dynamics in the VBR channel environment in Reibman and Haskell's paper [4]. Essentially the same constraints as in the CBR case apply except that now the channel rate is time varying. Denote  $C(i)$  the channel rate at time  $i$ , then (3) and (4), the encoder and decoder buffer occupancies can be rewritten as

$$B^e(i) = \sum_{j=1}^i R(j) - \sum_{j=1}^i C(j) \quad (7)$$

$$B^d(i) = \begin{cases} \sum_{j=1}^i C(j) - \sum_{j=1}^{i-\Delta N} R(j), & \text{when } i \geq \Delta N \\ \sum_{j=1}^i C(j), & \text{when } i < \Delta N \end{cases} \quad (8)$$

Combining the encoder buffer occupancy (7) at time  $i$  and decoder buffer occupancy (8) at time  $i + \Delta N$ , we can get the following equation:

$$\begin{aligned} B^d(i + \Delta N) &= \sum_{j=1}^{i+\Delta N} C(j) - \sum_{j=1}^i R(j) = \sum_{j=i+1}^{i+\Delta N} C(j) - \left( \sum_{j=1}^i R(j) - \sum_{j=1}^i C(j) \right) \\ &= \sum_{j=i+1}^{i+\Delta N} C(j) - B^e(i) \end{aligned} \quad (9)$$

In order to prevent the decoder buffer from underflowing, we have to keep the right side of (9) always greater than 0. Therefore in the VBR case we can again define the effective buffer size as the maximum buffer occupancy at the encoder buffer that will not cause decoder buffer underflow. The main difference is that from (9) we have

$$B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C(j) \quad (10)$$

and the effective buffer size depends on the frame interval and is equal to the sum of future  $\Delta N$  channel rates as. This is an intuitively obvious result since it just states that in order to arrive within  $\Delta N$  frames, the amount of data to be transmitted cannot exceed the total channel rate available during that period.

This formulation can also be used to design rate control algorithms in other VBR scenarios such as time-varying wireless channels [6], where future rates are not known, but stochastic models can be used instead. Here we concentrate on the case where the future  $\Delta N$  channel rates, and thus the effective buffer size, can be planned or controlled in advance, but where the choice of the channel rates is subject to a constraint, as will be the case in ATM networks.

## OPTIMAL RATE CONTROLS IN CBR AND VBR CHANNEL

### Encoder Optimization in CBR Channel

The optimal encoder bit-allocation for a discrete set of quantizers with buffer constraints in the CBR channel can be solved by dynamic programming [5]. In this formulation, assuming a sufficiently long end-to-end delay, a trellis can be formed where each branch represents

a choice of quantization for the frame and has associated a distortion. We can find out the trellis path with minimum distortion using the Viterbi algorithm (VA) [7]. In each stage, each node represents the buffer occupancy of the encoder buffer and accumulated distortion of the video. Each branch represents a possible quantizer choice with its associated bit rate and distortion. Therefore, the process of connecting the node in current stage to the nodes in the next stage by all the possible branch represents all the possible transition of buffer occupancy and accumulated distortion when the quantizer corresponding too that branch is chosen by the encoder to code the current video frame. Since the buffer occupancy is constrained by the maximum applicable buffer size, this transition of buffer occupancy cannot exceed the maximum buffer size in every stage. The branches which can cause the buffer to overflow are then eliminated. For the trellis paths arriving at the same node, the one with smallest accumulated distortion is chosen, and the rest are pruned out Refer to Fig. 1 and to [5] for the details.

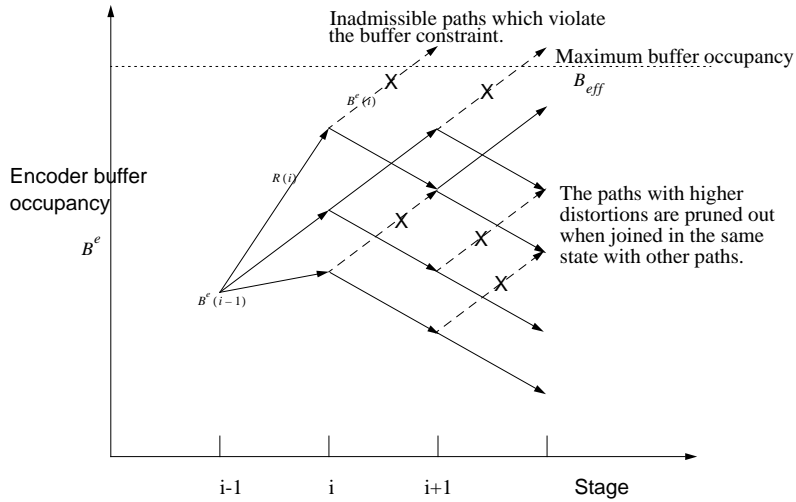


Figure 1: Buffer constrained optimization in the CBR channel case.

### Joint optimization of encoder and VBR channel

Our goal is to jointly optimize the encoder and VBR channel bit-allocation with buffer and leaky bucket constraints. The encoder can choose not only the coding bit rate  $R(i)$ , which is constrained by the effective buffer size, but also the channel rate  $C(i)$ , which is constrained by leaky bucket policing function.

We consider the VBR channel case with leaky bucket constraint as an example of VBR rate constrained scenarios. Let  $\bar{C}$  be the leaky bucket drain rate and let  $LB_{max}$  be the leaky bucket size in bits (i.e. if the bucket can store  $L$  tokens and each token represents  $r$  bits to be transmitted, then  $LB_{max} = r \cdot L$ ). Refer to [3] for a more detailed description of the leaky bucket mechanism. In order to apply dynamic programming to solve this joint optimization problem, we assume that the number of possible channel rates available to the encoder is finite. The encoder will choose one rate from this finite set of possible channel rates to transmit the data in each time interval. The allowable bit rates will be constrained by the leaky bucket mechanism.

Assume the buffer and leaky bucket never underflow or overflow, and the physical buffer size is large enough to hold all the coded data. Under this assumption we do not have to take into account the physical buffer size in optimization. We define this joint optimization problem as follows.

Given the encoder buffer occupancy,  $B^e(i) = B^e(i-1) + R(i) - C(i)$ , the leaky bucket fullness,  $LB(i) = LB(i-1) + C(i) - \bar{C}$  and the effective buffer size  $B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C(j)$ , find  $R(i)$  and  $C(i)$  for all  $i$ , to minimize the total distortion of the coded video sequence:

$$\min \sum_{j=1}^N D(i) \quad (11)$$

$$\text{subject to } LB(i) \leq LB_{max} \quad \text{and} \quad B^e(i) \leq B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C(j), \quad \forall i \quad (12)$$

## Two-Variables Viterbi Algorithm

The effective buffer size constraint is needed to prevent decoder underflow and it depends on the choice of channel rates  $\Delta N$  frames ahead. At time  $i$ , the variables to be chosen by the encoder are thus the encoding bit rate  $R(i)$  at time  $i$  and channel bit rate  $C(i + \Delta N)$  at time  $i + \Delta N$ .

From (9) the decoder buffer occupancy can be expressed as

$$B^d(i + \Delta N) = B^d(i + \Delta N - 1) + C(i + \Delta N) - R(i) \quad (13)$$

except for the first  $\Delta N$  frames. The leaky bucket fullness at time  $i + \Delta N$  can also be expressed as

$$LB(i + \Delta N) = LB(i + \Delta N - 1) + C(i + \Delta N) - \bar{C} \quad (14)$$

The goal of this constrained optimization is to choose  $R(i)$  and  $C(i + \Delta N)$  which will not cause decoder buffer underflow nor exceed the leaky bucket constraint. Such constraints can be expressed as

$$B^d(i + \Delta N) = B^d(i + \Delta N - 1) + C(i + \Delta N) - R(i) \geq 0 \quad (15)$$

$$LB(i + \Delta N) = LB(i + \Delta N - 1) + C(i + \Delta N) - \bar{C} \leq LB_{max} \quad (16)$$

From these applicable  $R(i)$  and  $C(i + \Delta N)$  for all  $i$ , we want to find out the ones with minimum accumulated distortion. Such choice of  $R(i)$  and  $C(i + \Delta N)$  with minimum cost can be found by using the Viterbi algorithm (VA). The method is similar to that used in the CBR case except that there are two state variables:  $B^d(i + \Delta N)$  and  $LB(i + \Delta N)$ . In the VA, if there are two or more trellis paths which end in the same state at some intermediate stage, all the paths can be pruned out except for the one with minimum cost. The buffer and leaky bucket constraints are used within the VA to prevent the trellis paths from going into the inadmissible states, i.e. the states where  $B^d(i + \Delta N) < 0$  or  $LB(i + \Delta N) > LB_{max}$ . Conceptually the trellis paths at each stage can extend into a two dimensional space defined by all possible values of the state variables  $B^d(i + \Delta N)$  and  $LB(i + \Delta N)$ . The whole “state



60,000 bits per frames in both VBR and CBR case. In the VBR channel cases, leaky bucket sizes equal to  $10 \times C$ ,  $15 \times C$  and  $20 \times C$  are implemented.

Fig. 4 is the evolution of the bit rates used by encoder and channel in CBR and VBR channel case. The end-to-end delay equals to 15 frames in both VBR and CBR channel cases. The leaky bucket size of the VBR channel equals to  $20 \times C$ .

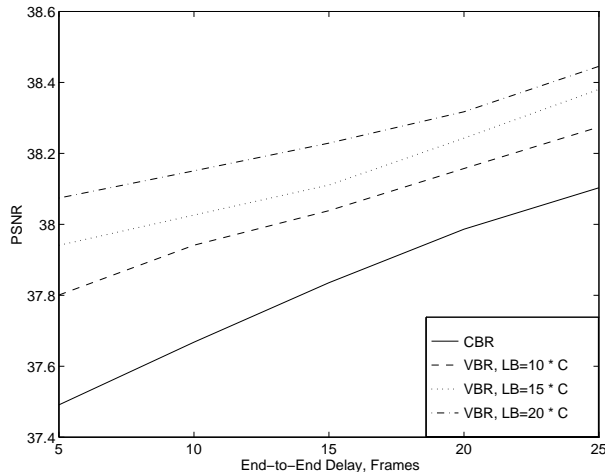


Figure 3: PSNR of coded video sequence in VBR and CBR channel.

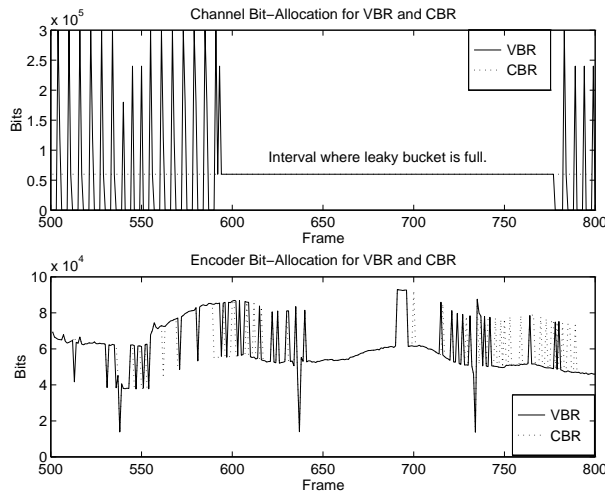


Figure 4: Bit rates used by encoder and channel in VBR and CBR channel.

Reibman and Haskell [4] have shown that in the VBR channel with leaky bucket policing function, the constraint for the encoder to choose the bit rate for each frame is the same as in the CBR channel case with virtual encoder buffer size which equals to the sum of physical buffer size and leaky bucket size. The equivalent channel rate is equal to the leaky bucket



drain rate. Our experiments show that the performance of these two cases in terms of total distortion are the same. This equivalence comes from the fact that the leaky bucket can be treated as an internal buffer inside the network, but which will not cost any extra delay to the data flow. In this specific case, the joint optimization of the encoder and VBR channel with leaky bucket policing function can be done by using the 1-variable VA used in the CBR channel case with larger buffer size which equals to the sum of physical buffer size and leaky bucket size. In the real world implementation of such solution, the encoder can code the video sequence with the optimum bit-allocation obtained from this 1-variable VA assuming that the larger buffer size (adding the physical buffer and the virtual leaky bucket size) is available. The the encoder transmits the data from the encoder buffer at the highest possible rate which does not violate the leaky bucket constraint. Once the channel cannot carry the coded data inside the encoder buffer because the leaky bucket is full, then the data is left in the physical buffer.

This equivalence however is not necessary true if other policing function other than leaky bucket is used. The 2-variables VA is capable to solve more general joint optimization problem. Therefore by properly defining the state variables, this algorithm can applied to find the optimum bit-allocations for encoder and channel when other policing functions are applied on the communication channel.

## References

- [1] H. Kanakia, P. P. Mishra, and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport," in *Proc. ACM SIGCOMM'93*, (San Francisco, CA), Sept. 1993.
- [2] J. C. Bolot and T. Turletti, "A rate control mechanism for packet video in the internet," in *Proc. Infocom '94*, 1994.
- [3] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. on sel. area in comm.*, vol. 9, pp. 325–334, April 1991.
- [4] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for atm networks," *IEEE Trans. on CAS for video tech.*, vol. 2, pp. 361–372, Dec. 1992.
- [5] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. on Image Proc.*, vol. 3, pp. 26–40, Jan. 1994.
- [6] A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," in *ICIP'95*, (Washington, D.C.), Oct. 1995.
- [7] G. D. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268–278, Mar. 1973.