Joint Selection of Source and Channel Rate for VBR Video Transmission under ATM Policing Constraints*

Chi-Yuan Hsu, Antonio Ortega,†
Integrated Media Systems Center
Dept. of Electrical Eng.-Systems
University of Southern California
Los Angeles, California

Amy R. Reibman
AT & T Laboratories
Holmdel, NJ

Abstract

VBR transmission of video over ATM networks has long been said to provide substantial benefits, both in terms of network utilization and video quality, when compared with conventional CBR approaches. However, realistic VBR transmission environments will certainly impose constraints on the rate that each source can submit to the network. In this paper we formalize the problem of optimizing the quality of the transmitted video by jointly selecting the source rate (number of bits used for a given frame) and the channel rate (number of bits transmitted during a given frame interval). This selection is subject to two sets of constraints, namely, (i) the end-to-end delay has to be constant to allow for real-time video display and (ii) the transmission rate has to be consistent with the traffic parameters negotiated by user and network.

For a general class of constraints, including such popular ones as the Leaky Bucket, we introduce an algorithm to find the optimal solution to this problem. This algorithm allows us to compare VBR and CBR under the same end-to-end delay constraints. Our results indicate that variable rate transmission can increase the quality of the decoded sequences without increases in the end to end delay.

Finally, we show that for the leaky bucket channel, the channel constraints can be combined with the buffer constraints, such that the system is identical to CBR transmission with an additional, infrequently imposed, constraint. Therefore, video quality with a leaky bucket channel can achieve the same quality of a CBR channel with larger physical buffers, without adding to the physical delay in the system.

Keywords: ATM networks, Leaky bucket, Policing function, Effective buffer size, Viterbi algorithm, VBR, Video transmission, Delay constraint.

---

1 Introduction

Future networks will carry various forms of information, such as data, audio, and video. Even high bandwidth video transmission applications over these networks are becoming feasible due to technology advances that expand the network bandwidth. Various applications such as video conferencing, video-on-demand (VOD), or multicast over the network are being studied and implemented. Video applications have some special characteristics that distinguish them from more traditional data transmission. For example, *Lossy compression* will be used so that, (i) the output generated by a video coder will intrinsically be variable bit rate (VBR) for most practical compression algorithms and (ii) due to the lossy compression, the output rate, and thus the quality of the decoded data, can be chosen by the encoder. This gives rise to interesting trade-offs between channel utilization and video quality. Additionally, video transmission requires rigorous timing synchronization between encoder and decoder in order to guarantee *real time playback*. As will be seen, data corresponding to a specific video frame becomes useless if it arrives "too late" at the decoder.

The VBR nature of compressed video has been cited for a number of years [2] as a motivation for establishing networks which allow video transmission at variable rate while providing guarantees on the end-to-end delay and delay jitter in the transmission: the so called quality of service (QOS) guarantees. VBR transmission is useful because it better matches the intrinsic characteristics of video sources without requiring additional buffering delay. The QOS guarantees are required to ensure that real time display of video at the decoding end is possible. Asynchronous Transfer Mode (ATM) networks are an example of a network architecture which would allow this type of VBR transmission with QOS guarantees, as they incorporate flexible mechanisms for resource allocation [3, 4]. The main drawback of allowing VBR transmission for improved video quality is that the network becomes vulnerable to sources which are too bursty or attempt to exceed their negotiated bandwidth allocation [5, 6].

In order to ensure that QOS guarantees are maintained, the network will likely have to negotiate a contract with each user before the start of transmission which specifies the expected traffic parameters of the transmission. In addition, the network will probably resort to a policing mechanism which will be used to prevent the negative impact on the network of sources exceeding their negotiated traffic parameters. While not precluding that other mechanisms are used within the network, we concentrate here on policing functions *that are known to the video encoder and are monitored at the user-network interface* [7]. Policing functions such as leaky bucket (LB), jumping window, sliding window, etc, are commonly

used due to their simplicity and efficiency [8, 9]. Note that our methods would also be directly applicable to shaping for ITU-T defined parameters such as Peak Cell Rate and Sustainable Cell Rate.

In this paper we focus on real-time video communications over ATM networks[1]. We show that video transmission is then subject to both end-to-end delay constraints and channel rate constraints. Previous research has studied these two constraints [5] and also considered various heuristic methods to select source and channel rates to satisfy both of them [5, 10, 11, 12]. The novelty in this work is that we formalize the problem of source and channel rate selection and propose a new algorithm to achieve the optimal solution for most policing functions of interest [1]. Independently of our work other researchers have studied a similar formulation and provided an alternative algorithm based on a "sliding window" approach [13, 14, 15]. When using a sliding window global optimality can no longer be guaranteed but on-line encoding becomes a possibility.

This paper extends previous work [16] which considered rate allocation under delay constraints for a CBR environment in two ways. First, we formulate the problem to include the delay constraints when channel rate is variable. Second, we introduce a new algorithm to find the optimal solution when the channel rate is constrained by a large class policing mechanisms. Note that while other researchers have considered shaping techniques for bit-streams generated by video coders here we go one step further by introducing the shaping requirements, in the form of policing constraints, *within the video encoder loop*. We are thus able to adjust the quality of the encoded video in a Rate-Distortion optimal manner while complying with required shape parameters. Further shaping can be also performed at the output of the encoder buffer without affecting the video quality.

Our algorithm allows us to present a global picture of a VBR video transmission system by considering all the interesting trade-offs among the available resources, namely, end-to-end delay, policing function parameters and physical buffer sizes at encoder/decoder. We can thus compute the maximum achievable video quality for a given video sequence with a variety of system configurations. From our experimental results, we also show that, for the specific environment of an ATM network with LB constraints, the optimization problem is equivalent to the constant bit rate (CBR) channel with a single buffering constraint, whenever sufficient physical buffering is available. Our methods can be used for off-line

---

[1]Note that by real-time we mean that data is read at some rate from the source and has to be played back in real-time from the network at the destination. In some cases the encoding is done on-line and thus the complexity of the encoder is limited. In others off-line encoding is possible and more complex algorithms can be used.

encoding, for benchmarking and also to derive approximate allocation algorithms which can operate under real-time encoding.

The paper is organized as follows. In Section 2 we formulate the problem by taking into account all the available resources (buffers and policing function) as well as the end-to-end delay. We can thus determine the general set of constraints on the system. Section 3 introduces an algorithm that obtains the optimal operating point for a given set of constraints. Experimental results are presented in Section 4 and are used to provide some general rules for the selection of system parameters. Finally, in Section 5, we show that for the LB channel, the channel constraints can be combined with the buffer constraints, such that the system is identical to CBR transmission with an additional, infrequently imposed, constraint.

2   Problem Formulation

Our goal in this paper is to consider the possible choices of parameters in a real-time video transmission system and study how they can be optimally traded-off to achieve various levels of video quality and network utilization. In particular we consider the following parameters: (i) A policing function which operates at the user-network interface and which specifies the number of bits that a given source can use, (ii) the end-to-end delay between encoder and decoder, and (iii) the physical buffer sizes at encoder and decoder. We start by formulating the problem with the above listed parameters.

2.1   Delay and Buffer Constraints

In a real-time video communications system, both encoder and decoder are attached to synchronous devices, and thus the end-to-end delay of a frame traversing the system should be constant[2]. If some of the information corresponding to a video frame arrives at the decoder after the scheduled decoding time, the information will be useless and the frame will thus be considered lost. Since the information received at the decoder is stored in a buffer before actually being decoded, we will call this situation decoder buffer underflow.

For a generic video communications system, the end-to-end delay, ΔT, is the time it will take for one frame to be transmitted and can be written as

$$\Delta T = \delta t_e + \delta t_c + \delta t_d, \tag{1}$$

where δt_e and δt_d are, respectively, the delays in encoder and decoder buffer, and δt_c is the channel transmission delay, see Fig. 1. The total end-to-end delay ΔT has to be constant for

---

[2]Note that the same would apply to a system where the information has been pre-encoded but real-time display at the decoder is required, as in VOD applications.

every frame in order to keep synchronized timing between encoder and decoder. Note that in general the channel delay itself is variable due to both scheduling and routing. The delay variations can be assumed to be small and disregarded. Alternatively, if $\bar{\delta t_c}$ is the average expected delay and δt_{max} is the maximum expected deviation from the average delay we can define $\delta t_c = \bar{\delta t_c} + \delta t_{max}$ as the maximum expected channel delay and use this in our formulation. To ensure that the system will work properly, we can overdimension the decoder buffer by $(\delta t_c - \delta t_{low}) \times R_{max}$, where R_{max} is the maximum rate per second that the channel can support and δt_{low} is the minimum channel delay, which could be zero[3]. Thus, without loss of generality we 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. Assume the length of the time interval for one video frame is $\mathcal{T}$. Given the end-to-end delay ΔT, the number of video frames stored in either encoder or decoder buffer is

$$\Delta N = \frac{\Delta T - \delta t_c}{\mathcal{T}}, \tag{2}$$

which can also be interpreted as the end-to-end delay expressed in units of frames.

Note that in what follows we consider that the encoder and decoder clocks are shifted by an amount equal to δt_c, the transmission delay. Thus, if the i-th frame interval starts at time t_i at the encoder, it will start at time $t_i + \delta t_c$ at the decoder. Refer to Fig 2. We will now discuss how the system is constrained by the end-to-end delay (Section 2.2) and the sizes of the buffers at encoder and decoder (Section 2.3).

2.2 Effective Buffer Size

Denote $B^e(i)$ and $B^d(i)$, respectively, the encoder buffer and decoder buffer occupancies. Let $C(i)$ and $R(i)$ be, respectively, the channel rate at time i (i.e. during the i-th frame interval) and the source rate used for i-th frame [4]. All of these variables have units of bits. The encoder and decoder buffer occupancies can be obtained as

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

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

---

[3]When delay is less than δt_c, data will be available at the decoder earlier than expected and may have to be stored longer, thus resulting in a larger buffer capacity. With this additional physical buffer capacity our formulation is still valid, and our algorithms can be used "as is".

[4]Note that the formulation does not change if one considers a time unit smaller than the frame interval. For example, as will be seen in our experiments, we can use the Group of Blocks (GOB) as the basic unit so that $C(i)$ and $R(i)$ become, respectively, the channel and source rates for the i-th GOB.

Because of the end-to-end delay, the decoder waits ΔN frame intervals before starting to decode the video frames available in its buffer. Here we are assuming sufficiently large buffers at encoder and decoder and thus we focus on the effect of buffer underflow. For most cases of interest, given that our goal is to minimize distortion, the set of quantizers will be such that encoder buffer underflow will not occur and we thus do not take this into account. However *decoder buffer underflow*, i.e. the situation where all the bits corresponding to a given frame are not present at the decoder in time to be decoded, may very well occur. This problem is important since it results in frame losses.

Combining the encoder buffer occupancy (3) at time i and decoder buffer occupancy (4) at time $i + \Delta N$, we have that:

$$
\begin{aligned}
B^d(i + \Delta N) &= \sum_{j=1}^{i+\Delta N} C(j) - \sum_{j=1}^{i} R(j) = \sum_{j=i+1}^{i+\Delta N} C(j) - (\sum_{j=1}^{i} R(j) - \sum_{j=1}^{i} C(j)) \\
&= \sum_{j=i+1}^{i+\Delta N} C(j) - B^e(i).
\end{aligned}
\tag{5}
$$

In order to prevent the decoder buffer from underflowing, we have to keep the right hand side of (5) always greater than zero. We introduce the concept of *effective buffer size*, $B_{eff}(i)$, which we define as the maximum level of buffer occupancy that the encoder can reach at time i such that the channel rates are adequate to transport all the bits without violating the end-to-end delay constraint (i.e. without producing decoder underflow.) From (5) the maximum level of encoder buffer occupancy is $\sum_{j=i+1}^{i+\Delta N} C(j)$, thus we have

$$
B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C(j)
\tag{6}
$$

and the effective buffer size depends on the frame interval and is equal to the sum of the future ΔN channel rates. We can guarantee that if the encoder buffer fullness $B^e(i)$ is always smaller than $B_{eff}(i)$, then the decoder buffer will not underflow. In the special case of a CBR channel, the channel rate is constant, i.e. $C(i) = C$, $\forall i$. The the effective buffer size is

$$
B_{eff}(i) = \Delta N \cdot C
\tag{7}
$$

which is also constant. Note that the buffer sizes at encoder and decoder should be the same in this case.

Although the physical size of the buffer can be very large, the actual buffer size that the encoder can use is constrained by the end-to-end delay, thus the *effective* buffer size can be potentially smaller than the physical buffer size. This concept is also useful in situations where the encoder has no control over the (variable) channel rates but seeks to avoid decoder underflow [17].

2.3   Physical Buffer Constraints

Let B_{max}^e and B_{max}^d, respectively, be the physical encoder and decoder buffer sizes, and assume that the end-to-end delay ΔN is a design parameter. In this section we study how the sizes of physical buffers at encoder and decoder affect other elements of the encoding system.

2.3.1   Constraints on the encoder buffer state

If the physical buffer size is smaller than $B_{eff}(i)$, then the constraints on the encoder buffer fullness become

$$0 \leq B^e(i) \leq B_{max}^e, \tag{8}$$

and, similarly, from (5), the conditions to prevent decoder underflow/overflow are

$$0 \leq \sum_{j=i+1}^{i+\Delta N} C(i) - B^e(i) \leq B_{max}^d \tag{9}$$

$$\text{or} \quad \sum_{j=i+1}^{i+\Delta N} C(j) - B_{max}^d \leq B^e(i) \leq \sum_{j=i+1}^{i+\Delta N} C(j) \tag{10}$$

These two constraints can be combined in a single one:

$$\max(\sum_{j=i+1}^{i+\Delta N} C(j) - B_{max}^d, 0) \leq B^e(i) \leq \min(\sum_{j=i+1}^{i+\Delta N} C(j), B_{max}^e). \tag{11}$$

From (11), if either the physical encoder or decoder buffer sizes are smaller than the effective buffer size, then the applicable buffer occupancy is not only upper bounded by the physical buffer size, but also lower bounded by a minimum buffer occupancy which may not be zero.

2.3.2   Constraints on the allowable channel rate

An intuitive interpretation of (11) is to say that the channel rates have to be sufficiently low so that information does not arrive too fast to the (small) decoder buffer; this explains the lower bound on buffer occupancy at the encoder. From (9) we have

$$B^e(i) \leq \sum_{j=i+1}^{i+\Delta N} C(j) \leq B_{max}^d + B^e(i) \tag{12}$$

which states that limited buffer sizes at encoder and decoder can actually impose a constraint on the range of channel rates that can be used. Thus, even if network policing imposes no restrictions on the admissible channel rates, arbitrary rates may not be possible for a given choice of buffer sizes at encoder and decoder.

2.3.3 Physical buffer size selection for given policing function

Obviously, a good choice of buffer sizes at encoder/decoder for given policing function parameters would be one such that the buffers are sufficiently large that they do not introduce any additional constraints on the channel rates. The physical buffer sizes constraint both the channel rates as in (12), and the encoder buffer fullness as in (11). Assume the end-to-end delay and policing function parameters are given and define

$$C_{max} = \max \sum_{j=i+1}^{i+\Delta N} C(j) \quad \forall i \tag{13}$$

as the maximum aggregate channel rate that can be allowed by the given channel policing function over a ΔN-frame interval. For example, in the leaky bucket case with bucket size LB_{max} and drain rate $\bar{C}$, the highest aggregate channel rate over any ΔN-frame period is $LB_{max} + \Delta N \cdot \bar{C}$. If the physical encoder and decoder buffer sizes are larger than C_{max}, i.e.

$$B_{max}^e \geq C_{max}, \quad B_{max}^d \geq C_{max} \tag{14}$$

then the condition $\sum_{j=i+1}^{i+\Delta N} C(j) \leq B_{max}^d + B^e(i)$ in (12) will always hold, and the following relations will always be true:

$$\sum_{j=i+1}^{i+\Delta N} C(j) \leq B_{max}^d, \quad \sum_{j=i+1}^{i+\Delta N} C(j) \leq B_{max}^e \tag{15}$$

and therefore the encoder buffer is only constrained by the effective buffer size, and there are no additional constraints due to the encoder/decoder buffer sizes.

3 Optimal Encoder and Channel Bit-Allocation

Based on the above constraints we now introduce a technique that can find the optimal operating points. Our technique is complex and may not be suitable for real time implementation but can be used for benchmarking or off-line encoding. Approximations to these techniques are possible [16] but are not considered here. In this section we concentrate on the case where the physical buffer constraints can be ignored.

3.1 CBR case

Given a discrete set of quantizers, and a buffer constraint for transmission through a CBR channel the optimal encoder bit-allocation can be found using dynamic programming techniques [16]. In this formulation, assuming a sufficiently long encoding delay, a trellis can

be formed where each branch represents a choice of quantization for the frame and has associated a distortion, refer to Fig. 3. The total distortion of a given path can be found by adding up the distortion of each of the branches comprising the path. The trellis path with minimum total distortion can be found using the Viterbi algorithm (VA) [18], a form of deterministic dynamic programming. The trellis state represents the buffer occupancy and thus each path in the trellis represents a possible solution. The basic idea is to simplify the search by eliminating the suboptimal paths, namely, those paths that overflow the buffer or those paths that reach a given node of the trellis with a cost higher than that of the minimum cost path at that node (see Fig. 3).

Note that when growing these trellises the number of states (possible buffer fullness levels) can become quite large and the set of paths can be relatively sparse compared to the number of states (thus, logically, few paths would be pruned since the paths would be unlikely to meet). In [16] it was shown that in general the granularity of the buffer states can be made coarser without affecting the result of the optimization. Thus we can consider buffer states spaced in, say, 100 or 1000 bit intervals, rather than spaced by just one bit. In this scenario paths are made to converge to the nearest buffer state. Both the error in rate and the incurred sub-optimality are minimal. The same approach will be used in the next section to quantize the channel rate levels.

3.2   VBR case

The dynamic programming method above can be extended to find the optimal encoder and channel bit-allocation jointly in the VBR channel environment. Our goal is to choose the number of bits to use for each frame in the sequence *and* the number of bits that the channel should transmit for each time slot (i.e. each frame interval), such that (i) the total video distortion is minimized and (ii) any applicable policing constraints are met.

Suppose we are given M possible quantizers for each frame, and P possible channel rates for each frame interval. Define $\mathbf{x} = \{x(1), x(2), ...x(N)\}$ as the sequence of quantizer choices, where $x(i) \in \{1, \ldots, M\}$ is the quantizer index for frame i. The number of bits generated is $R_{x(i)}(i)$ and the associated distortion is $D_{x(i)}(i)$. Also define $\mathbf{y} = \{y(1), y(2), ..., y(N + \Delta N)\}$ as the sequence of channel rate choices, where $y(i) \in \{1, \ldots, P\}$ is the index of channel rate, and the associated channel rate is $C_{y(i)}(i)$. Therefore, $\left\{ R_{x(1)}(1), R_{x(2)}(2), ..., R_{x(N)}(N) \right\}$ and $\left\{ D_{x(1)}(1), D_{x(2)}(2), ..., D_{x(N)}(N) \right\}$ are, respectively, the rate and distortion for each frame for a given choice of $\mathbf{x}$, $\left\{ C_{y(1)}(1), C_{y(2)}(2), ..., C_{y(N+\Delta N)}(N + \Delta N) \right\}$ represents the channel rates for each frame interval for given $\mathbf{y}$.

Because there is an end-to-end delay ΔN between encoder and decoder, the decoder is actually decoding the $(i - \Delta N)$th frame at time i. We will have to select at any instant $C_{y(i)}(i)$ and $R_{x(i-\Delta N)}(i - \Delta N)$ to prevent decoder buffer underflow, i.e.

$$B^d(i) = B^d(i-1) + C_{y(i)}(i) - R_{x(i-\Delta N)}(i - \Delta N) \geq 0 \tag{16}$$

In addition to the delay constraint, $C_{y(i)}(i)$ is also constrained by a policing function. Policing is implemented by keeping track of the bit rate transmitted through the channel by means of a monitor function and then imposing constraints on the allowable state of the monitor function. Define $L(i)$ as the state of the monitor function. For most cases of interest the change in the state of the monitor function will only depend on the previous state $L(i-1)$ and the choice of channel rate $C_{y(i)}(i)$, i.e.

$$L(i) = \mathcal{F}\left(C_{y(i)}(i), L(i-1)\right) \tag{17}$$

The policing function decides whether to admit the data with bit rate $C_{y(i)}(i)$ into the network according to a criteria:

$$\text{if } L(i) = \mathcal{F}\left(C_{y(i)}(i), L(i-1)\right) \in \mathcal{L}, \text{ admit } C_{y(i)}(i) \quad \text{otherwise, reject } C_{y(i)}(i) \tag{18}$$

where $\mathcal{L}$ represents the admissible region of the policing function $L(i)$. Refer to Fig. 4 for an example of such a system. We first introduce the general formulation and then show several examples of policing functions that can be seen to fit within this framework.

In the general case, (16) and (18) jointly constrain the admissible buffer state transitions from $B^d(i-1)$ to $B^d(i)$, and the monitor function transition from $L(i-1)$ to $L(i)$. Therefore, the joint encoder and channel rate allocation can be formulated as:

Formulation 1 *Find mappings* $\mathbf{x} : (1, ...N) \rightarrow (1, ..., M)$ *and* $\mathbf{y} : (1, ..., N + \Delta N) \rightarrow (1, ..., P)$ *that solve:*

$$\min \sum_{i=1}^{N} D_{x(i)}(i) \tag{19}$$

subject to the constraints:

$$B^d(i) \geq 0 \tag{20}$$

$$L(i) \in \mathcal{L}, \quad \forall i = 1, ..., N + \Delta N. \tag{21}$$

Therefore the channel rates $C_{y(i)}(i)$ and the encoder bit rate $R_{x(i-\Delta N)}(i - \Delta N)$ should be allocated jointly to avoid decoder buffer underflow and meet (20) and (21).

We use the Viterbi algorithm (VA) [19, 18] to find out the admissible solutions $\mathbf{x}$ and $\mathbf{y}$, which have the minimum overall distortion. A trellis with $N + \Delta N$ stages is formed where each state in stage i represents a decoder buffer fullness and a monitor function value. Therefore each node in the trellis is defined by its state pair $(B^d(i), L(i))$. Each branch links two nodes and represents a transition in decoder buffer and monitor function states from stage i to stage $i + 1$. Thus each branch corresponds to a choice of channel rate $C_{y(i)}(i)$ at time i and a choice of quantization rate $R_{x(i-\Delta N)}(i - \Delta N)$ for the $(i - \Delta N)$th video frame. The new state, $(B^d(i+1), L(i+1))$ can be obtained using (16) and (17). The transitions that violate the constraints given by (20) and (21) can be avoided by discarding the corresponding branches. Each branch also has associated the distortion corresponding to coding a video frame with the chosen quantization bit rate. Thus a path, which consists of one branch for every stage, represents one possible solution of encoder and channel allocation, and the trellis grown in this manner represents all the possible admissible solutions.

The VA reduces the optimal path search complexity by keeping only one trellis path for each node of the trellis, namely the one with minimal distortion up to that node. We need to prove that paths can be pruned without eliminating the optimal solution.

Lemma 1 *For all the possible quantizer choices $x(1), x(2), ..., x(i - \Delta N)$ and channel rate choices $y(1), y(2), ..., y(i)$ which have the same buffer fullness $B(i)$ and monitor function value $L(i)$ at time i, only the one with the smallest aggregate distortion $\sum_{j=1}^{i-\Delta N} D_j(j)$ can be a candidate for the optimal overall solution. The other paths with higher distortion are sure to be sub-optimal.*

Proof: From (16), we have that $B^d(i+1)$ only depends on $B^d(i)$, $C_{y(i+1)}(i+1)$ and $R_{x(i+1-\Delta N)}(i+1-\Delta N)$ no matter how $x(1), x(2), ..., x(i - \Delta N)$ and $y(1), y(2), ..., y(i)$ are chosen. From (17), we have that $L(i+1)$ only depends on $L(i)$ and $C_{y(i+1)}(i+1)$, no matter how $y(1), y(2), ..., y(i)$ are chosen. Therefore, $B^d(i)$ and $L(i)$ completely summarize the state of the system and two different choices of $x(1), \ldots, x(i - \Delta N)$ and $y(1), \ldots, y(i)$ are completely equivalent as far as the rest of the sequence is concerned if they result in the same $B^d(i)$ and $L(i)$. Thus a path can be discarded in favour of another path with same parameters and lower distortion without loss of optimality. $\square$

Note that it is also possible to perform pruning if there exists a dominant path. For example, if two paths A and B have the same distortion, but B has used more bits so far, it is possible to prune it out since A has achieved the same distortion with fewer bits.

Before giving the details of the algorithm, we provide examples of policing mechanisms which would fit into the class considered here. Note that these mechanisms are representative

11

of the most popular approaches considered in the literature.

Example 1: Leaky Bucket In the VBR channel with leaky bucket constraints the policing function keeps an imaginary buffer with input rate $C_{y(i)}(i)$ and constant output rate $\bar{C}$. A channel rate $C_{y(i)}(i)$ which causes the imaginary buffer to overflow is thus inadmissible. In this case the state variable $L(i)$ is the leaky bucket state at time i, $LB(i) = \max(LB(i-1) + C_{y(i)}(i) - \bar{C}, 0)$, and thus $\mathcal{L} = \{LB(i) \mid LB(i) \leq LB_{max}\}$, where LB_{max} is the maximum size of the imaginary buffer.

Example 2: Double leaky bucket The double leaky bucket policing mechanism [20, 6] simultaneously uses two leaky buckets with different set of parameters (drain rate and bucket size). In this case the monitor function state is uniquely defined by $DLB(i) = (LB_1(i), LB_2(i))$ a state variable with two components, which are the states of the two leaky buckets. The rate is admissible if neither of the two leaky bucket constraints is violated, i.e. $\mathcal{L} = \{(LB_1(i), LB_2(i)) \mid LB_1(i) \leq LB1_{max}, \ LB_2(i) \leq LB2_{max}\}$, where $LB1_{max}$ and $LB2_{max}$ are the respective maximum sizes of the imaginary buffers.

Example 3: Jumping window A jumping window constraint is such that the rate over intervals of W time units cannot exceed a given maximum value, R_{max}. In this case the state variable to be used is $JW(i)$ which can be defined as $JW(i) = C_{y(i)}(i)$ if $i = k \cdot W$ for some integer k. Therefore , $\mathcal{L} = \{JW(i) \mid JW(i) \leq R_{max}\}$.

Example 4: Sliding window In this case a similar state variable formulation can be used but it results in a more complicated state than in the other examples. Here, assuming a sliding window of size W, we would need to define the state variable $SW(i) = \left(C_{y(i)}(i), C_{y(i-1)}(i-1), \ldots, C_{y(i-W+1)}(i-W+1)\right)$. Therefore, $\mathcal{L} = \{SW(i) \mid \sum_{k=0}^{W-1} C_{y(i-k)}(i-k) \leq R_{max}\}$.

3.3 Optimization by Viterbi Algorithm

With **Lemma 1** and the constraints (20) and (21), sub-optimal and inadmissible solutions can be pruned out in every intermediate stage. The algorithm can be described as follows:

Algorithm 1 *Joint encoder and channel bit-allocation by Viterbi algorithm:*

Step 0: Initialize the decoder buffer fullness $B^d(0)$ and monitor function $L(0)$. Each node in the trellis at stage i is defined by a pair $(B^d(i), L(i))$. Start the loop with $i = 1$.

Step 1: At stage i, add all possible branches to the end of every surviving path node $(B^d(i-1), L(i-1))$ at stage $i-1$. The new state is $(B^d(i), L(i))$, obtained as:

$$B^d(i) = \begin{cases} B^d(i-1) + C_{y(i)}(i), & when \ i \leq \Delta N, \\ B^d(i-1) + C_{y(i)}(i) - R_{x(i-\Delta N)}(i - \Delta N), & when \ i > \Delta N \end{cases}$$

12

$$L(i) = \mathcal{F}\left(C_{y(i)}(i), L(i-1)\right) \in \mathcal{L}$$

$$\forall x(i) \in \{1, ..., M\}, \quad \forall y(i) \in \{1, ..., P\}$$

where $C_{y(i)}(i)$ and $R_{x(i-\Delta N)}(i - \Delta N)$ are such that constraints (20) and (21) are not violated. Refer to Fig. 5.

Step 2: For all the branches arriving at node $(B^d(i), L(i))$, keep only the one with smallest aggregate distortion $\sum_{j=1}^{i-\Delta N} D_j(j)$ and prune out the others. The smallest aggregate distortion path is the surviving path for that state.

Step 3: Increment i and go to Step 1 until $i = N + \Delta N$.

Step 4: At stage $N + \Delta N$, find out the state transitions with smallest aggregate distortion $\sum_{j=1}^{N} D_j(j)$. The corresponding choices $\mathbf{x}$ and $\mathbf{y}$ are the best quantizers and channel rates choices for each frame. The associated $R_{x(i)}(i)$ and $C_{y(i)}(i)$ are the optimal encoder and channel bit-allocation for the given video sequence.

4   Experimental Results and Comparison

For our experiments, we first use an H.261[21, 22] encoder operating in "intra-frame only" mode[5]. Additional results for an MPEG interframe coder are presented in Section 4.3. To better control the encoder bit rate allocation, we apply the rate-control taking groups of blocks (GOBs) as the basic unit, so that each GOB can quantized using a different quantization step.

The H.261 public domain software implementation of [23] was used in our experiments. Our basic results would not be affected if we chose a different set of quantizers or a different channel rate. In our experiments we use the "football" sequence [6], one of the standard video sequences used in MPEG standardization. The time scale corresponds to the unit of time needed to display a GOB. The target average bit rate for each GOB is $\bar{C}$=5,200 bits/GOB, i.e. 52,000 bits/frame. We choose this value of $\bar{C}$ as being roughly the average rate per GOB achievable with the selected set of quantizers above and the chosen video sequence. $\bar{C}$ is therefore the channel rate per GOB for the CBR case and will also be used as the drain

---

[5]This choice simplifies the computation of the distortion for each operating point because the video quality for a video frame is independent of the choice of the quantization step size of other frames.

[6]Our luminance-only input sequence is in CIF format with 352x240 pixels, grouped into 10 GOBs and there are four possible choices of quantization step sizes, 8, 10, 12 and 31 (in H.261 the quantization step size can range from 1 to 31) for each GOB.

rate of the LB in the VBR case. We use Peak Signal-to-Noise Ratio (PSNR) to measure the quality of the decoded frames[7].

We use the algorithm described in the previous section to find out the best encoder and channel bit-allocation to achieve maximum PSNR in the VBR channel environment with the leaky bucket and double leaky bucket policing function [8]. We compare the best PSNR that the encoder can achieve with different policing function parameters and end-to-end delay constraints. We concentrate here on the comparison of CBR and VBR, while the comparison between different policing mechanisms is left for future work. We also select simple sets of parameters for channel rate, leaky bucket rates, etc. since our goal is to be able to qualitatively compare CBR and VBR, rather than provide definite figures for the gains of the former over the latter. These gains will depend on the application, rates, encoder and specific video sequences.

4.1 CBR and VBR Channel Comparison

In the VBR channel with leaky bucket policing function, different leaky bucket sizes from $5 \times \bar{C}$ to $30 \times \bar{C}$ are used, and we set the bucket drain rate to $\bar{C}$, since this provides the most straightforward comparison to the CBR case. We use the algorithm presented in Section 3 to find the best joint encoder and channel rate selection with different leaky bucket parameter and different delay constraints.

Figs. 6 and 7 represent, respectively, the encoder buffer fullness during each GOB interval in the CBR channel and the encoder buffer and leaky bucket fullness during each GOB interval in the VBR channel with leaky bucket policing function. Note that the VBR case requires a slightly larger physical buffer at the encoder but provides a higher quality as seen in Fig. 8. Comparing Fig. 6 with Fig. 7 (top) it can be seen that the VBR buffer occupancy exceeds the physical buffer size of the CBR case (52,000 bits) by less than 10,000 bits. Note however that this additional buffer size does not result in additional end-to-end delay.

Fig. 8 represents the best average PSNR that the encoder can achieve with given average bit rate, end-to-end delay, and LB constraints. The optimal encoder bit-allocation for the CBR channel environment is the extreme case of VBR where LB_{max} is zero and can be obtained using the dynamic programming approach of [16]. By looking at the contour lines of the optimal PSNR from Fig. 8, which are shown in Fig. 9, it can be seen that the PSNR

---

[7]Given the mean squared error, MSE, in a given frame, we have that $PSNR = 10 \log_{10}(255^2/MSE)$.

[8]We provide results only for LB-based policing mechanisms. Experiments for sliding window policing are possible but the optimization procedure becomes quite complex in this case. Jumping window experiments have been performed but are not included here due to lack of space and to the inherent limitations of this mechanism in terms of its tolerance of variability.

level depends mostly on the sum of buffer size and LB size. Thus the upper bound on achievable video quality is the same for a given value of the sums of $\bar{C} \cdot \Delta N$ (equivalent to the effective buffer size in the CBR channel case) and the LB size. This also applies to the extreme CBR channel case, where the LB size is zero. This observation will be further explored in Section 5 where we will motivate that VBR with LB constraints is similar to CBR with a larger buffer size.

The comparison between CBR and VBR under two sets of LB constraints is summarized by Figs. 10 and 11. Note that our average rates and end-to-end delays are *exactly the same for CBR and VBR*. Thus these two figures demonstrate the advantages of VBR transmission for the same overall rate. Fig. 10 shows the rate and distortion per GOB. It can be seen that the increased PSNR in the VBR case is achieved by locally increasing the source rate over what would be possible in the CBR case. Fig. 11 shows the channel rates per GOB or frame and also indicates that, given the possibility of selecting a variable channel rate, as in the VBR case, it is advantageous to do so[9]. Also, it can be observed that the larger the bucket sizes the more variable the channel rates will be.

4.2 Double Leaky Bucket Function in VBR Channel

In the simulation for the VBR channel with double leaky buckets (DLB) policing function [20], the drain rate and bucket size are 5,200 bits/GOB and 52,000 bits for the larger bucket, and are 5,600 bits/GOB and 11,200 bits for the smaller bucket. Fig. 12 shows the average PSNR of this DLB case compared to the single LB cases where each leaky bucket parameter is applied individually. From the above figure we can observe that the encoder is mainly constrained by the larger bucket in the DLB case if the end-to-end delay is large enough.

Fig. 13 shows the encoder buffer and LB fullness of the large bucket for each GOB interval. Compare the bucket fullness to that of the single leaky bucket shown in Fig. 7, where the single leaky bucket is the same as the larger of the double leaky buckets. With 2 LB, the bucket fullness rises more slowly. This is because we have introduced an additional constraint of the smaller leaky bucket, thus imposing more restrictions on the short term rate, and producing less bursty channel rates.

4.3 MPEG Video Experiments

Video coding algorithms which exploit the temporal correlation between consecutive frames through motion compensation, such as those used in the MPEG standards[24, 25], result in

---

[9]Note that to simplify the optimization we only consider a discrete set of possible channel rates. Our results would be similar with increased granularity in the choice of channel rates.

greatly improved rate-distortion efficiency as compared to intraframe-only methods as those considered in the preceding sections. However these interframe methods introduce a prediction loop and therefore also a dependency in the rate-distortion characteristics. For each quantization choice in a predictor frame a different R-D curve can be found for the predicted frame. For example, the predicted frame will require more bits for the same quantizer if the predictor frame was coded with a coarse quantizer rather than a fine quantizer. This dependency complicates optimization procedures as it multiplies the number of allowable operating points and requires specific procedures for optimal design [26].

To provide results for MPEG video we propose an approximation to the optimal solution. Due to the dependency a different R-D curve is generated for a given frame for each possible quantizer selection on the predictor. We alternately fix the predictors and compute the R-D data, then find the optimal solution as if the R-D points were independent using the algorithm of Section 3. Then we use the result of this optimization step to encode the predictor frames and re-start the iteration.

We use the public domain software encoder of [27]. Our results are summarized in Figs. 14, 15 and 16. Our goal is again to compare CBR and VBR under LB constraints. Fig. 14 shows how the use of motion estimation results in a large number of bits being used for intra coded frames. Fig. 15 demonstrates as in previous sections the advantage of VBR transmission. When comparing the total number of bits used for each group of pictures (GOP) [10] ,it can be seen that VBR helps by allowing bits to be saved and used in later GOPs. Note that for many GOPs the difference in PSNR is very small. For those GOPs where VBR and CBR the differences close to about 1dB.

While it may be expected that VBR would produce more gains compared to CBR for video compressed using inter-frame techniques such as motion compensation, Fig. 15 indicates that this is not the case. Quality performance improvements are also small in this case. In fact, in the following section, we show that performance improvements are bounded by the quality of CBR video using a physical buffer as large as the virtual buffer in the VBR case, regardless of the compression techniques used. This is also demonstrated in Fig. 16 which experimentally shows that VBR is equivalent to having a larger buffer without incurring in the additional end-to-end delay. Note that the difference in average PSNR is relatively small, however the difference in specific GOPs or scenes can be significant. Effectively, VBR borrows bits from "easy" video segments to increase the quality of "difficult" video segments.

---

[10]The set of frames including an intra-coded frame and all the predicted frames until the next intra frame. We use GOPs of size 6 and use 2 B-frames per P-frame, i.e. one GOP has the form IBBPBB.

5   Leaky bucket channel revisited

As seen in Fig. 9, the best video quality that the encoder can achieve depends primarily on the sum of $\bar{C} \cdot \Delta N$ and the leaky bucket size LB_{max}, and the performance is nearly the same as that in a CBR channel where the buffer size is $\bar{C} \cdot \Delta N + LB_{max}$. In fact, it is possible to prove that the leaky bucket constraint collapses into the encoder buffer constraint, except for an additional constraint which comes into play when the physical buffer sizes are small. In particular, the video quality with a leaky bucket channel constraint and physical buffer B_{max}^e can be no better than the quality with a constant rate channel constraint and physical buffer $B_{max}^E = B_{max}^e + LB_{max}$.

5.1   Simplifying constraints

To prove the analogy between the leaky bucket VBR channel and a CBR channel, we show how the constraints on the channel rate determined by the leaky bucket parameters can be combined with each of the constraints determined by the physical encoder and decoder buffers. A variant of some of this derivation can be found in [5].

Clearly, the fullness of the physical encoder buffer cannot exceed the size of the physical buffer. Since we are also interested in maximizing quality, we avoid underflowing the buffer. Therefore, from (3) the physical encoder buffer constraint is

$$0 \le B^e(i) = \sum_{j=0}^{i} R(j) - \sum_{j=0}^{i} C(j) \le B_{max}^e. \tag{22}$$

Similarly, we avoid underflowing the leaky bucket to maximize video quality. The leaky bucket provides a constraint on the channel rate,

$$0 \le \sum_{j=0}^{i} C(j) - i\bar{C} \le LB_{max}. \tag{23}$$

Adding equations (22) and (23) we have that

$$0 \le \sum_{j=0}^{i} R(j) - i\bar{C} \le LB_{max} + B_{max}^e. \tag{24}$$

Note that this is exactly equivalent to the constraint on the encoded bit rates $R(i)$ that would result if we had a physical buffer of size $B_{max}^E = LB_{max} + B_{max}^e$ with drain rate $\bar{C}$, since the quantity between the inequalities of equation (24) is the fullness of a virtual buffer with constant drain rate $\bar{C}$. A similar result was also shown in equation (35) of [5]. We define B_{max}^E as the virtual buffer size.

Now, we examine the decoder buffer when $i > \Delta N$, after the start-up time. From (4)

$$0 \leq \sum_{j=0}^{i+\Delta N} C(j) - \sum_{j=0}^{i} R(j) \leq B_{max}^{d}. \tag{25}$$

Multiplying equation (25) by minus one and adding the result to equation (23) with time index $i + \Delta N$, we have

$$-B_{max}^{d} \leq \sum_{j=0}^{i} R(j) - (i + \Delta N)\bar{C} \leq LB_{max} \quad \text{or}$$

$$-B_{max}^{d} + \bar{C} \cdot \Delta N \leq \sum_{j=0}^{i} R(j) - i\bar{C} \leq LB_{max} + \bar{C} \cdot \Delta N. \tag{26}$$

Again, the quantity between the inequalities in equation (26) is the fullness of a virtual buffer with constant drain rate $\bar{C}$. Now, combining the constraint on the virtual buffer fullness for the encoder in equation (24) and for the decoder in equation (26), we have

$$\max\{0, \bar{C} \cdot \Delta N - B_{max}^{d}\} \leq \sum_{j=0}^{i} R(j) - i\bar{C} \leq LB_{max} + \min\{\bar{C} \cdot \Delta N, B_{max}^{e}\} \tag{27}$$

If the physical buffer sizes are chosen such that $B_{max}^{e} \geq \bar{C} \cdot \Delta N$ and $B_{max}^{d} \geq \bar{C} \cdot \Delta N$, this constraint reduces to the well-known constraint with a constant-rate channel,

$$0 \leq \sum_{j=0}^{i} R(j) - i\bar{C} \leq B_{max}^{E}.$$

However, as pointed out in [5], there is also a constraint on the sum of any ΔN adjacent encoded frames, which can be obtained by adding (22) and (25),

$$0 \leq \sum_{j=i}^{i+\Delta N-1} R(j) \leq B_{max}^{d} + B_{max}^{e}. \tag{28}$$

because ΔN frames must be stored at either the encoder or decoder at any given time after the start-up period. This constraint does not come into play when the physical buffers are sufficiently large (see the discussion in Section 2.3.3). However, if the leaky bucket is large, or the physical buffer is small then the constraint in (28) can reduce the video quality below that generated by an equivalent virtual buffer size B_{max}^{E}.

To summarize, the leaky bucket constraints can be combined not only with the physical encoder buffer constraint [5], but also with the physical decoder buffer constraint. Therefore, the problem of encoder rate control in a leaky-bucket channel reduces to the well-known encoder rate control problem in a CBR channel, with the exception of the constraint imposed by equation (28).

18

5.2 Quality optimization

Because the constraints condense, it is no longer necessary to jointly optimize the selection of both the source and channel rates. An optimal selection of the source rate can be found using a two-dimensional trellis similar to that in section 3.3. However, we do not include it here for space considerations. On the other hand, a simple suboptimal approach would be to use the optimization method in [10] while simultaneously verifying the constraint in equation (28). That is, we can optimize the quantizer selection for a CBR channel and physical buffer the size of the virtual buffer B_{max}^E using the one-dimensional trellis in [16], but only paths which satisfy the constraint in equation (28) would be added to the trellis for a given quantizer step size. This requires the previous $\Delta N - 1$ quantizer step sizes to be stored, in addition to the current one.

Once the source rate is optimized, there are a wide range of available channel rates that can transport the compressed video. One promising method for selecting channel rates is to smooth the traffic using the available decoder buffering and the tolerable end-to-end delay [10, 28, 29, 30, 31]. Salehi et. al. [30] have shown that this so-called "workahead smoothing" can improve network performance by an order of magnitude.

6 Conclusions

In this paper we have studied real time video transmission over networks where VBR transmission is possible under constraints on the channel rates that the encoder can select. ATM networks are an example of this type of scenario. We have shown how two sets of constraints come into play for such a system when it comes to selecting the source and channel rates, namely (i) constraints due to the constant end-to-end delay needed to maintain real time video playback, and (ii) constraints due to network policing of the channel rate usage. We have considered end-to-end delay, policing constraints and physical buffer sizes at encoder and decoder as our major design parameters and have shown how they each can affect the resulting video quality.

We have formulated an optimization problem where the goal is to select the source and channel rates to maximize the video quality without violating the above constraints. We have introduced an algorithm based on dynamic programming which solves this optimization problem for most practical policing functions, including the leaky bucket and jumping window. We have used our algorithm to demonstrate experimentally the advantages of VBR transmission. We have shown how increased PSNR is possible in the VBR case with the same

average channel rate and end-to-end delay. For the special case of VBR with LB constraint we have shown that the constraints are equivalent to those of a CBR scenario with a larger buffer, as long as the actual physical buffers are sufficiently large.

Acknowledgement: The authors would like to acknowledge the reviewers for their useful and constructive comments.

References

[1] C.-Y. Hsu and A. Ortega, "Joint encoder and VBR channel optimization with buffer and leaky bucket constraints," in *Symposium on Multimedia Communications and Video Coding*, (Brooklyn, NY), Oct 1995.

[2] W. Verbiest, L. Pinnoo, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE JSAC*, vol. 6, pp. 1623–1632, Dec. 1988.

[3] J. Y. Hui, "Resource allocation for broadband networks," *IEEE JSAC*, vol. 6, pp. 1598–1608, December 1988.

[4] G. de Veciana, G. Kesidis, and J. Walrand, "Resource management in wide-area ATM networks using effective bandwidths," *IEEE JSAC*, vol. 13, pp. 1081–1090, August 1995.

[5] 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.

[6] A. Ortega, M. W. Garrett, and M. Vetterli, "Rate constraints for video transmission over ATM networks based on joint source/network criteria," *Annales des Télécommunications*, vol. 50, pp. 603–616, Jul.-Aug. 1995.

[7] ATM Forum, *ATM User-Network Interface Specification, Version 3.0.* Prentice-Hall, 1993.

[8] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE JSAC*, vol. 9, pp. 325–334, April 1991.

[9] L. Dittmann, S. B. Jacobsen, and K. Moth, "Flow enforcement algorithms for ATM networks," *IEEE JSAC*, vol. 9, pp. 343–350, April 1991.

[10] A. R. Reibman and A. W. Berger, "Traffic descriptors for VBR video teleconferencing," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 329–339, April 1995.

[11] H. Harasaki and M. Yano, "A study on VBR coder control under usage parameter control," in *Proc. of Fifth International Workshop on Packet Video*, (Berlin), pp. F2.1–F2.5, March 1993.

[12] M. Kawashima and H. Tominaga, "A study on VBR video transmission under usage parameter control," in *Proc. Fifth International Workshop on Packet Video*, (Berlin), pp. F3.1–F3.5, 1993.

[13] J.-J. Chen and D. W. Lin, "Optimal coding of video sequence over ATM networks," in *Proc. of ICIP'95*, vol. I, (Washington, DC), pp. 21–24, October 1995.

[14] J.-J. Chen and D. W. Lin, "Optimal bit allocation for video coding under multiple constraints," in *Proc. of ICIP'96*, vol. III, (Lausanne, Switzerland), pp. 403–406, Sept. 1996.

[15] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding video signal over ATM networks," *IEEE JSAC*, 1997. This issue.

[16] 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.

[17] A. Ortega and M. Khansari, "Rate control for video coding over variable bit rate channels with applications to wireless transmission," in *Proc. of ICIP'95*, (Washington, D.C.), Oct. 1995.

[18] G. D. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[19] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[20] A. Ortega and M. Vetterli, "Multiple leaky buckets for increased statistical multiplexing of ATM video," in *Proc. of the 6th Packet Video Workshop*, (Portland, OR), Sep. 1994.

[21] *ITU-T Recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbits*, Mar. 1993.

[22] M. Liou, "Overview of the px64 kbit/s video coding standard," *Comm. of the ACM*, vol. 34, pp. 59–63, Apr. 1991.

[23] PVRG-P64 CODEC v. 1.2, The Portable Video Research Group (PVRG), Stanford University, `ftp://havefun.stanford.edu/pub/p64/P64v1.2.tar.Z`.

[24] D. LeGall, "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, pp. 46–58, Apr. 1991.

[25] Inform. Technology - Generic Coding of Moving Pictures and Associated Audio, ITU Draft Rec. H.262, ISO/IEC 13818-2, Mar. 1994.

[26] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Proc.*, vol. 3, pp. 533–545, Sept. 1994.

[27] PVRG-MPEG CODEC v. 1.2, The Portable Video Research Group (PVRG), Stanford University, `ftp://havefun.stanford.edu/pub/mpeg/MPEGv1.2.tar.Z`.

[28] W. Ding, "Joint encoder and channel rate control of VBR video over ATM." preprint, Oct. 1995.

[29] S. Jung and J. S. Meditch, "Adaptive prediction and smoothing of MPEG video in ATM networks," in *Proc. of IEEE Intl. Conf. on Comm.*, June 1995.

[30] J. Salehi, Z.-L. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing," in *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pp. 222–231, May 1996.

[31] J. McManus and K. Ross, "Video-on-demand over ATM: constant-rate transmission and transport," *IEEE JSAC*, vol. 14, pp. 1087–1098, Aug. 1996.

List of Figures

Biography

Chi-Yuan Hsu (S'96) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1991, and the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1994. He is currently working toward the Ph.D. degree in the electrical engineering at the University of Southern California.

Since 1995 he has been a Teaching Assistant and Research Assistant in the Signal and Image Processing Institute and Integrated Media Systems Center at the University of Southern California. His research interests include joint source-channel coding and rate control of video transmission over broadband and wireless networks.

Antonio Ortega (S'92, M'95) was born in Madrid, Spain, in 1965. He received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid (UPM), Madrid, Spain in 1989 and the Ph.D. in Electrical Engineering from Columbia University, New York, NY in 1994.

He was a research assistant at the Image Processing Group at UPM (1990). At Columbia he was a graduate research assistant at the Center for Telecommunications Research (1991-94) and was supported by a scholarship from the Fulbright commission and the Ministry of Education of Spain.

Since September 1994 he has been an Assistant Professor in the Electrical Engineering-Systems department at the University of Southern California. At USC he is also a member of the Integrated Media Systems Center, an NSF Engineering Research Center, and the Signal and Image Processing Institute. In 1995 he received the NSF Faculty Early Career Development (CAREER) award. He is also an Associate Editor for the IEEE Transactions on Image Processing. His research interests are in the areas of image and video compression and communications. They include topics such as joint source-channel coding for robust video transmission, rate control and video transmission over packet wired or wireless networks.

Amy R. Reibman received the B.S., M.S. and Ph.D. degrees in electrical engineering from Duke University in 1983, 1984, and 1987, respectively. From 1988 to 1991, she was an assistant professor in the Department of Electrical Engineering at Princeton University. She is currently a Distinguished Member of Technical Staff in the Image Processing Software and Technology Research at AT&T Laboratories.

She was the Technical Program Chair for the Sixth International Workshop on Packet Video in 1994, and is currently Technical Co-chair for the First IEEE Workshop on Mul-

timedia Signal Processing. She is also a member of the IEEE Signal Processing Society's Technical Committees on Multimedia and on Image and Multidimensional Signal Processing. Her research interests include video compression systems for transport over non-traditional networks.
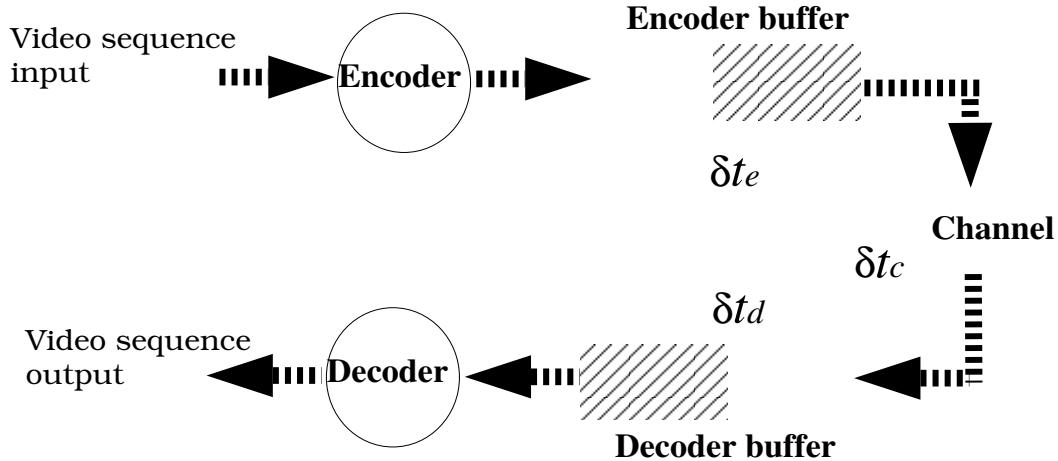
Figure 1: Delay components of a generic video communication system.
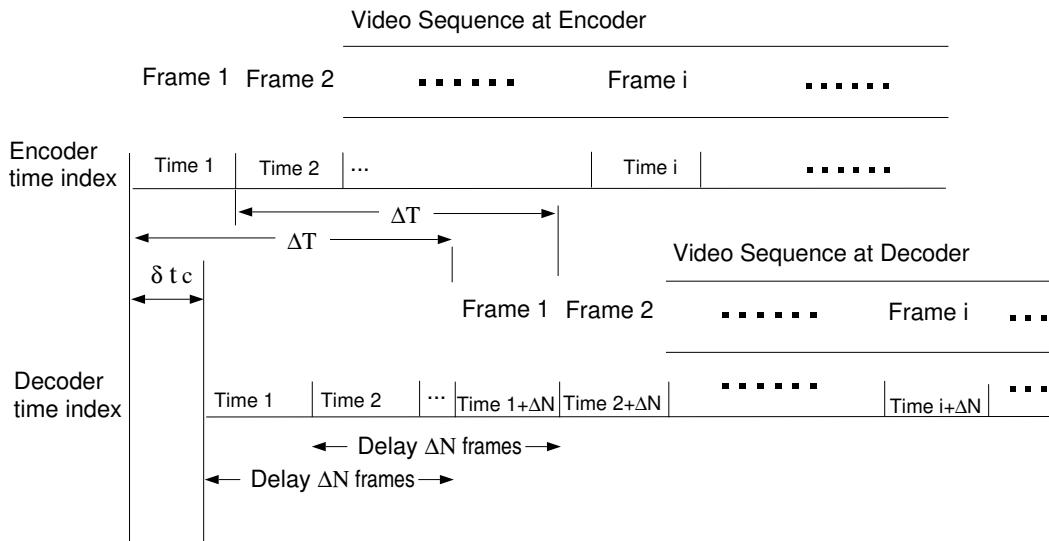
Figure 2: Timing diagram of encoder and decoder, note that the clocks at encoder and decoder are shifted by δt_c the transmission delay.
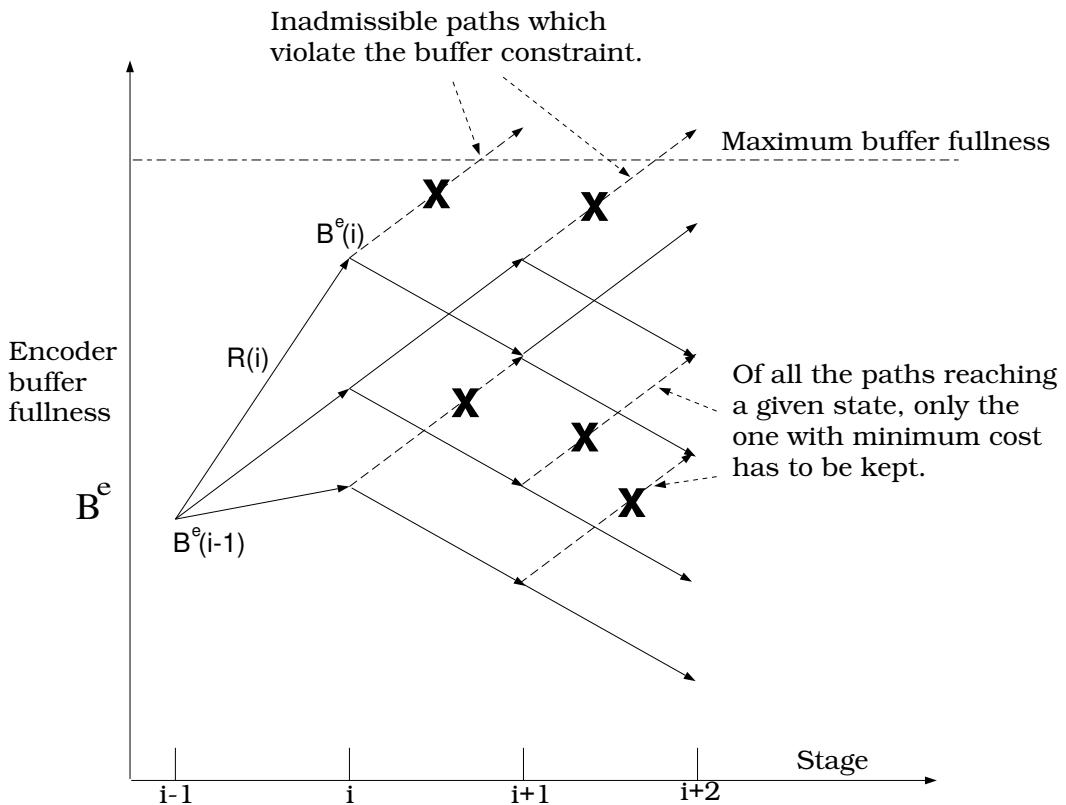
Figure 3: Buffer constrained optimization in the CBR channel case.
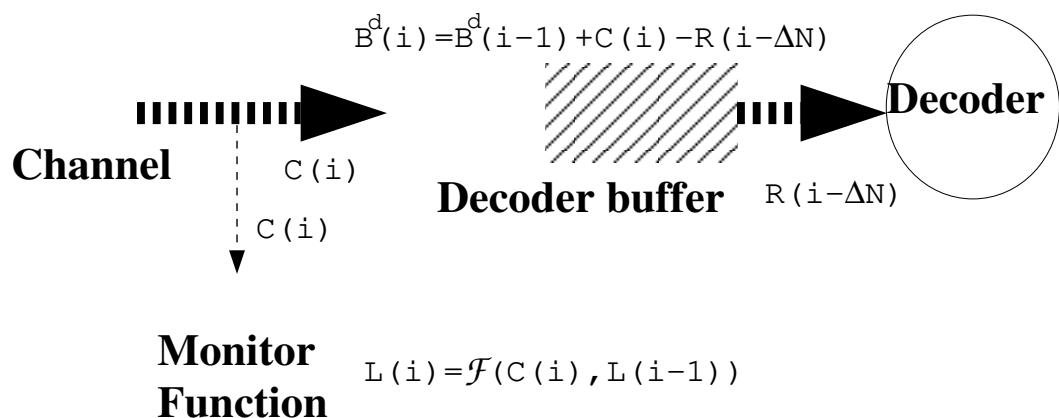
Figure 4: Decoder buffer in the receiver end.
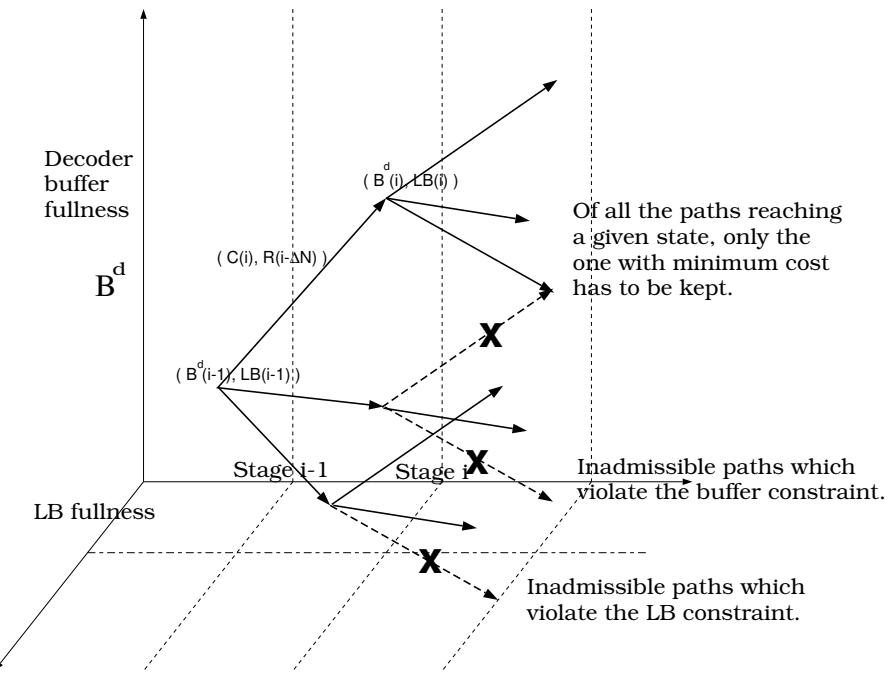
Figure 5: Buffer and leaky bucket constrained optimization in the VBR channel case.
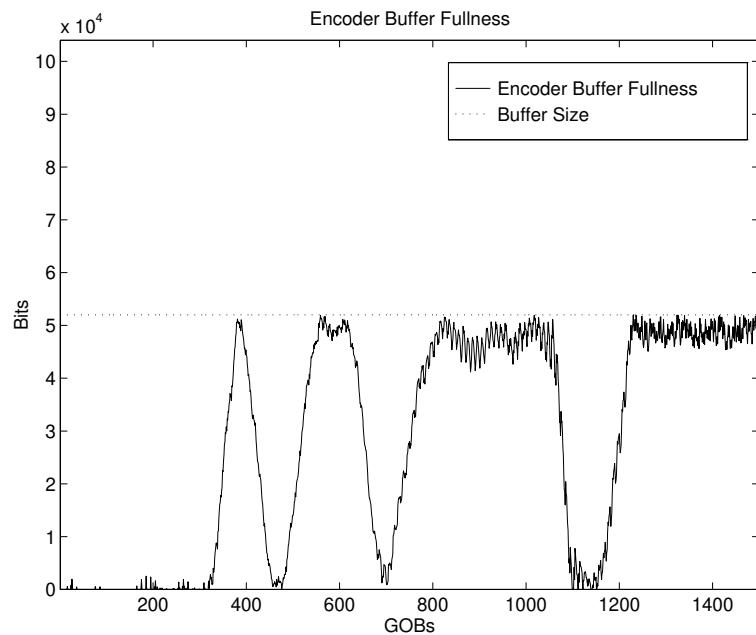
Figure 6: Encoder buffer fullness for CBR channel. Delay = 10 GOBs.
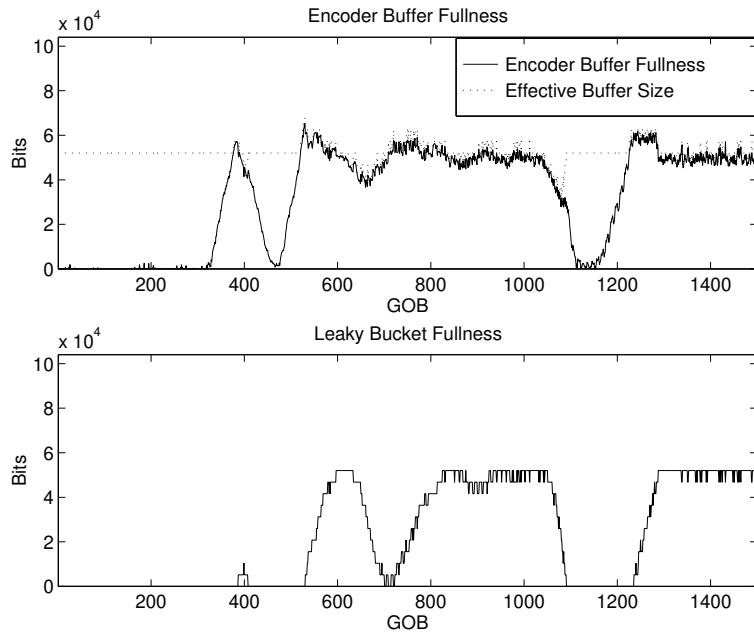
Figure 7: Encoder buffer and leaky bucket fullness for VBR channel. Delay = 10 GOBs, Leaky Bucket Size = $10 \cdot \bar{C} =$52,000 bits. Note that using variable channel rates allows us to increase the effective buffer size when needed.
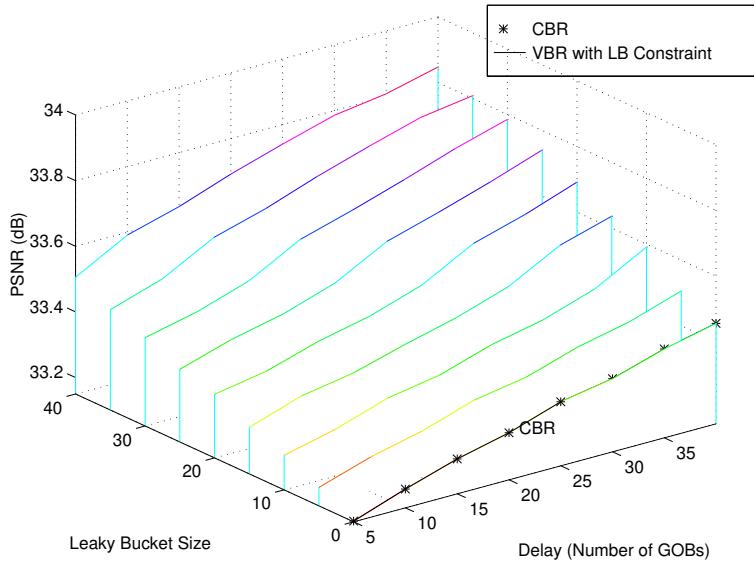
Figure 8: Average PSNR of video coded with different delay and leaky bucket constraints.
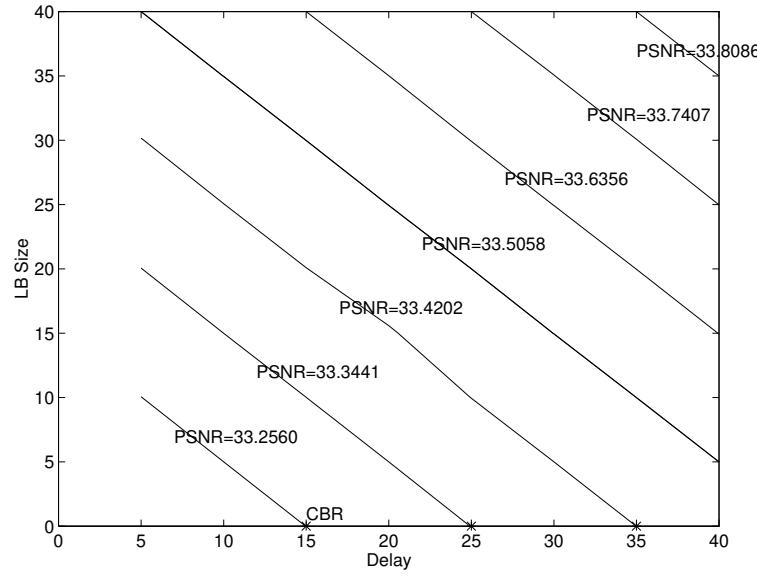
Figure 9: Contour of the average PSNR.
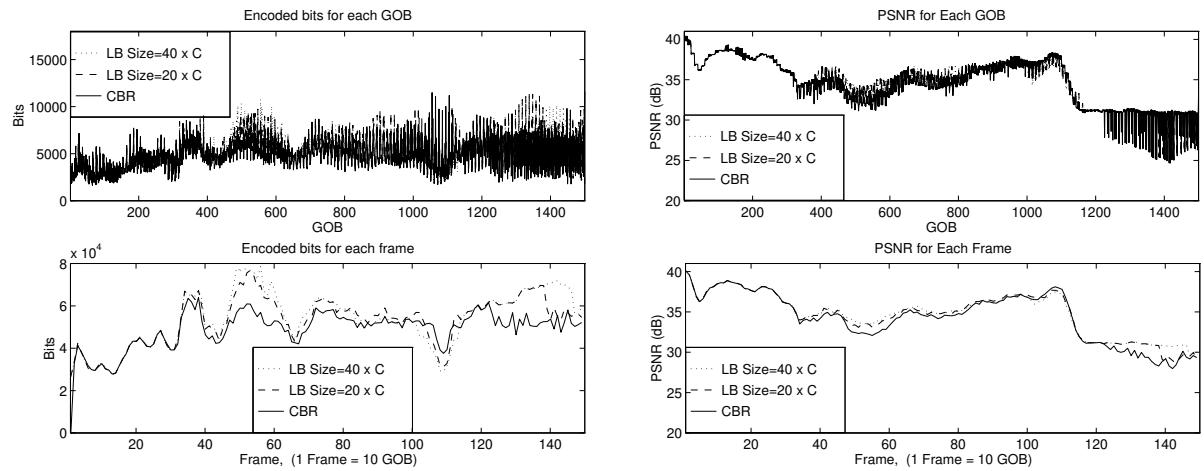
Figure 10: Encoded bits and PSNR for each GOB and frame when source rate and channel rate are jointly selected in the VBR channel with leaky bucket constraints.
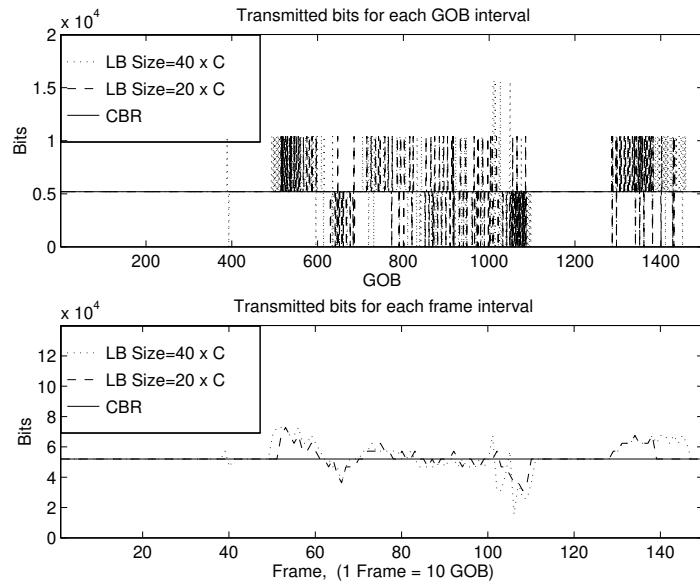
Figure 11: Transmitted bits for each GOB and frame interval when source rate and channel rate are jointly selected in the VBR channel with leaky bucket constraints.
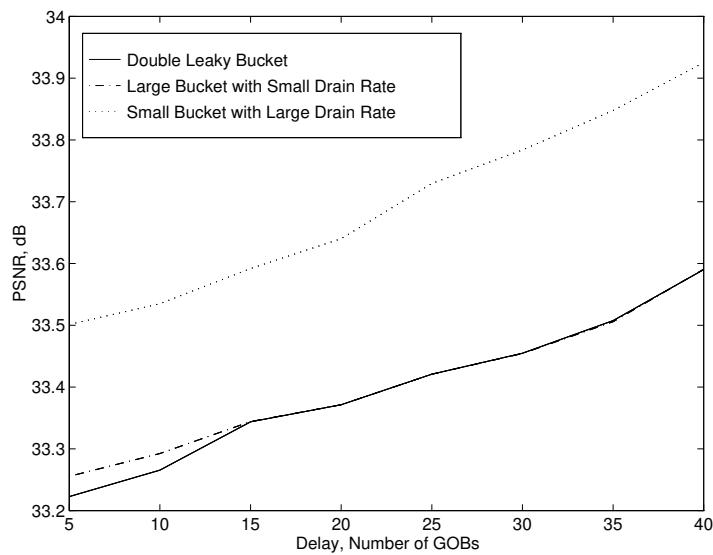
Figure 12: Best average PSNR of the video sequence for the double leaky bucket case and the other single leaky bucket cases.
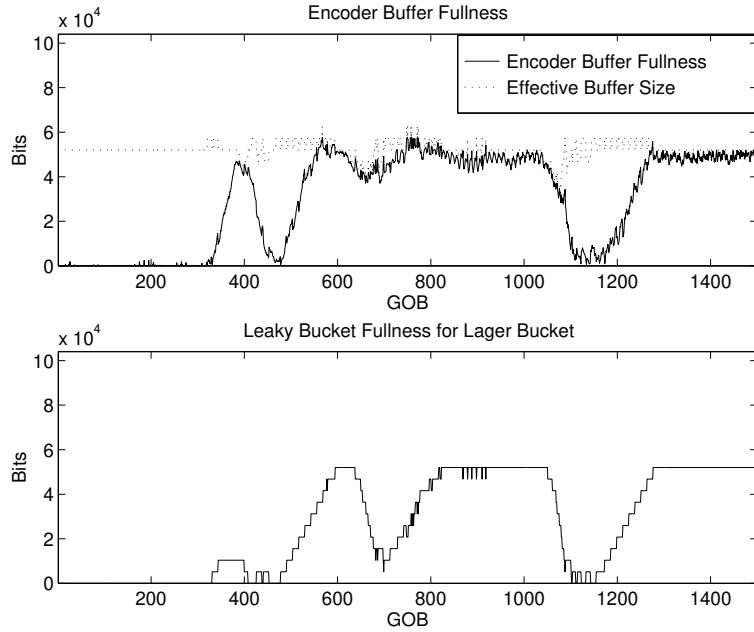
Figure 13: Encoder buffer and bucket fullness of larger bucket in DLB case. Delay = 10 GOBs. Larger bucket: size=52,000 bits, drain rate=5,200 bits/GOB. Smaller bucket: size=10,200 bits, drain rate= 5,600 bits/GOB.
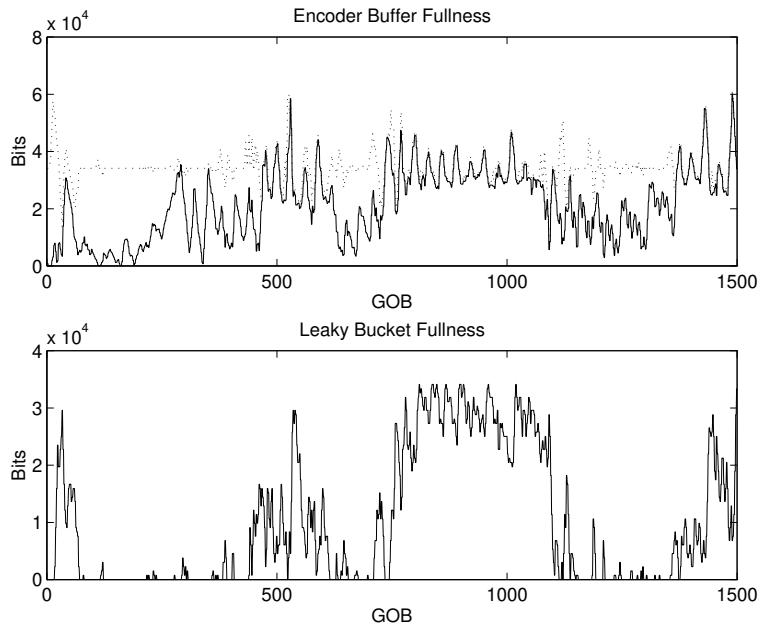

Figure 14: Encoder buffer and leaky bucket fullness using MPEG encoder in the VBR channel. Target average rate $\bar{C}$ = 3,400 bits/GOB. Delay = 10 GOBs, Leaky Bucket Size = $10 \cdot \bar{C}$ =34,000 bits/GOB.
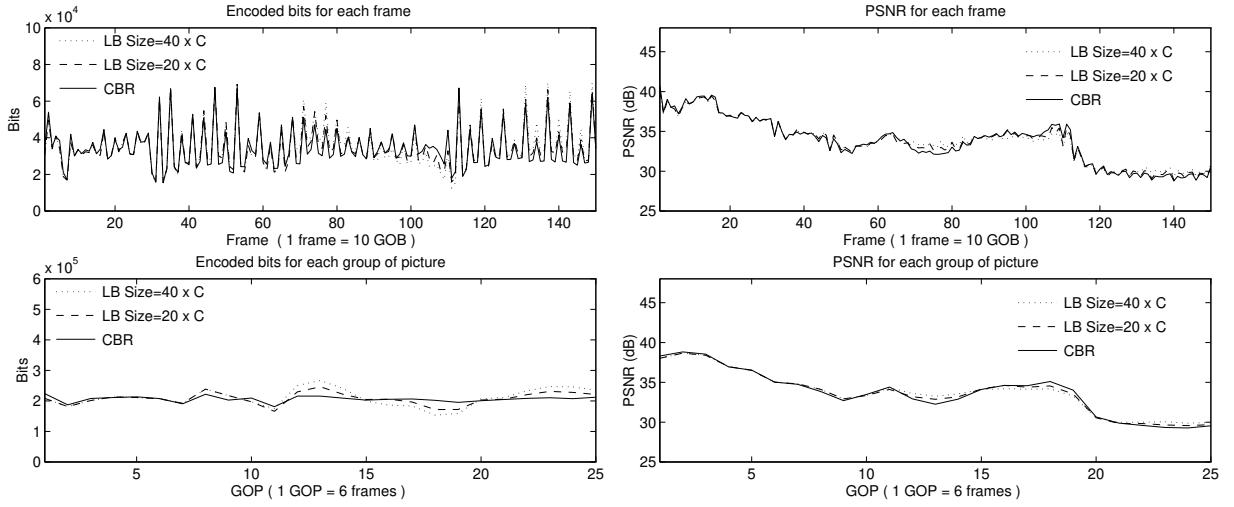
Figure 15: Encoded bits and PSNR for each frame and group of picture (GOP) using MPEG encoder in the VBR channel. Target average rate $\bar{C} = 3,400$ bits/GOB. Delay = 10 GOBs, Leaky Bucket Size = $10 \cdot \bar{C}$ =34,000 bits/GOB
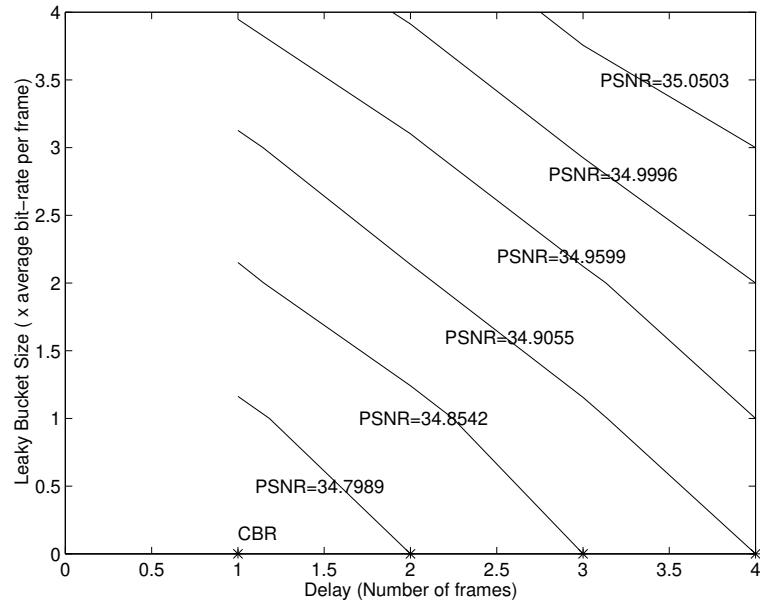

Figure 16: Contour of the average PSNR using MPEG encoder in the VBR channel. Target average rate $\bar{C}$=5,200 bits/GOB