Rate constraints for video transmission over ATM networks based on joint source/network criteria *

Antonio ORTEGA **
Mark W. GARRETT **
Martin VETTERLI **

Abstract

In this paper, we study the rate constraints required for variable rate video transmission over ATM *networks. Our objective is to achieve both good source quality and efficient network utilization. We show that these objectives may not be achieved simultaneously unless both network and source coding considerations are taken into account. In particular, we show that, given a rate constraint, a greedy source coding strategy will reduce the potential for statistical multiplexing gain in the network. We propose two alternative methods to achieve our goals. The first method requires* non-greedy *source coding techniques, implementable through rate control, such that video encoders will only use the bit rate needed to achieve a certain, nearly constant quality level. As a consequence, low activity scenes will use a fraction of the maximum allowable bit rate. The second method calls for increasing the number of rate constraints imposed on each connection so that the amount of bandwidth used in the worst case (*ie *by greedy coders) is limited. Experimental results for a medium length (5 min) video sequence are given.*

Key words : ATM, Videocommunication service, Traffic control, Variable bit rate, Quality of service, Statistical multiplexing, Picture coding.

Résumé

Dans cet article les contraintes devant être imposées sur le débit de transmissions vidéo sur réseaux ATM *sont étudiées. L'objectif est d'obtenir une bonne qualité pour la source, ainsi qu'une utilisation efficace du réseau. Ces deux objectifs ne peuvent pas être atteints si les critères de source et de réseau ne sont pas pris en compte simultanément. En particulier, un codage de source glouton réduit le potentiel pour le gain de multiplexage statistique. Deux méthodes pour atteindre ce but sont proposées. Une première méthode fait appel à un codage de source non glouton, tel que l'objectif du codeur vidéo est de maintenir une qualité constante. Dans ce cas, les scènes contenant peu d'activité utiliseront un nombre de bits réduit. La deuxième méthode consiste à augmenter le nombre de contraintes imposées sur le débit vidéo de façon à limiter le nombre de bits utilisés dans le pire cas (c'est-à-dire dans le cas du codeur glouton). Des résultats expérimentaux pour une séquence vidéo de durée moyenne (5 min) sont présentés.*

Mots clés : Multiplexage temporel asynchrone, Service vidéocommunication, Maîtrise trafic, Débit transmission variable, Qualité service, Multiplexage statistique, Codage image.

CONTRAINTES SUR LE DÉBIT PRENANT EN COMPTE À LA FOIS DES CRITÈRES DE SOURCE ET DE RÉSEAU POUR UNE TRANSMISSION VIDÉO SUR RÉSEAU ATM

Contents

* This work was presented in part at the 5th International Workshop on Packet Video, Berlin, Germany, Mar. 93 [1] and at the 6th Intl. Workshop on Packet Video, Portland, Oregon, Sept. 94 [2]. Work supported in part by the Fulbright Commission and the Ministry of Education of Spain. This work was done while at the Dept. of Electrical Eng. and Center for Telecom. Research, Columbia University.
** Dept. of Electrical Engineering Signal and Image Processing Institute, University of Southern California 3740 McClintock Avenue, CA 90089-2564 USA.

I. INTRODUCTION

Variable bit rate (VBR) transmission (*) of video over packet networks represents a departure from traditional problems in both the networking and coding fields and promises to have two major advantages with respect to the traditionally used constant bit rate (CBR) approach, namely : (a) constant video quality due to removal of buffering constraints at the video encoders; (b) more efficient use of network bandwidth through statistical multiplexing.

Implementations of packet video transmission have been reported recently for video over local area networks [3] or video multicasting over the Internet [4, 5]. Both cases have in common the lack of quality of service (QOS) requirements for the network performance. The user can only expect *best-effort* performance from the network and therefore a rate control at the encoder is needed in order to change the video frame rate and/or frame quality depending on the network conditions. (If the rate were not changed, the information might sometimes, *eg* when congestion occurs, be received too late to be usable by the receiver.) However there have been no implementations reported so far in a guaranteed environment such as that permitted by asynchronous transfer mode (ATM) networks [6]. Although the ATM design has sufficient flexibility to accomodate various transmission modes, so far most of the reported experiments with video over ATM have simply implemented well known CBR or best effort schemes. Here we concentrate on the more challenging scenario where the network is expected to provide statistical guarantees on performance, *ie* where each user agrees with the network on a set of performance parameters (*eg* maximum delay, delay jitter, etc.) which will have to be met on a statistical basis (*eg* the maximum delay will not be exceeded more than a certain percent of the time). This type of QOS guaranteed transmission is of interest since it is under these conditions that the advantages of ATM (constant video quality, statistical multiplexing) over other networking schemes are more likely to be realized. Throughout this paper when refering to ATM transmission we will be thus considering QOS-guaranteed transmission.

The achievable QOS depends on many factors, such as routing strategies and queueing disciplines, which will be specific of the network implementation. In this paper we concentrate on the rate constraints that will be imposed on each of the video sources. We will argue that choosing the *right* set of constraints will be beneficial to network performance regardless of the specific implementation considered.

One of the factors which has slowed down progress in the integration of video within ATM networks has been the lack of interaction between the network and source coding fields. On the one hand, analyses of network performance have been based on the assumption that a video source could be characterized by a more or less elaborate probabilistic model [7, 8, 9, 10, 11, 12], with models being obtained from traces of source bit rate under fixed quantization settings for the encoder. On the other hand, work on encoding schemes for packet video coders has tended to see the network as a black box, of which only those parameters relevant to the encoding process, *eg* rate constraints and packet loss rates, are known [12]. Thus, the predicted performance gains for sources and network could be achieved provided that the sources and network behaved as assumed in their respective models. Analyses of network performance based on source models could be misleading in that (i) it may be hard to characterize the sources when more than a few seconds of encoded video are considered [13] and, more importantly, (ii) the models do not take into account that, for a given network constraint, the source will very likely be rate controlled or self-regulated. Figure 1 illustrates the idea of *self-regulating coders* [14, 15]. Typically, source models tend to characterize sources operating with a *constant quantizer* mode (see Fig. 1a). However, a constant quantizer mode may violate the transmission constraints, which are agreed upon by network and user

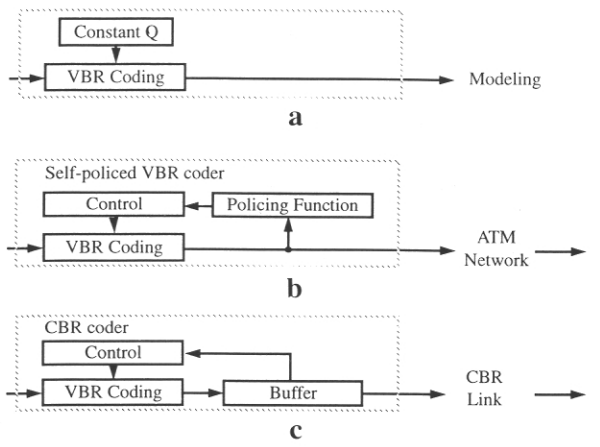

FIG. 1. — Three configurations for transmission of VBR coded video. Note that the control box sets a quantization parameter Q.

(a) Typical configuration for studying the statistical behavior of video source and modeling the output bit rate. (b) Self policing for transmission over a packet network. (c) Transmission over a CBR link.

Trois configurations pour la transmission de vidéo codée à débit variable (VBR).
On remarque que l'élément de commande modifie le paramètre Q.

(a) Configuration pour l'étude du comportement statistique d'une source vidéo et pour la modélisation de son débit. (b) Régulation automatique pour la transmission sur réseau par paquets. (c) Transmission sur une connexion à débit constant (CBR).

(*) Note that we refer to VBR and constant bit rate (CBR) *transmission*, while we assume the source coding algorithm to be VBR, *ie* to produce a variable number of bits per frame at a given quantization scale.

as part of the contract negotiated at the connection set up stage (*) [6]. Thus, rate control will be required on the sources (Fig. 1b) and therefore the *constant quantizer* models are likely to be incorrect. Note the similarity between (Fig. 1b) and the typical buffer feedback that is used for transmission over CBR channels (Fig. 1c). If their respective policing and buffering constraints are analogous then the video quality will be the same in both cases. For example, leaky bucket policing [16, 17] imposes exactly the same constraint on the quantizer as a fixed buffer and fixed output rate. Since statistical multiplexing gain (SMG) is achievable, VBR should still be superior to CBR, even under the same rate constraints. However, as will be seen, SMG is not *always* achieved : the conditions that make statistical multiplexing possible are one the key issues addressed in this paper.

Traditional video coding is *greedy* in that the performance criterion which is maximized is the average signal-to-noise ratio (SNR). This is true in the theoretical rate-distortion (R-D) framework and also in the more *ad hoc* design techniques used in practical video coding. While this might be appropriate for images or short sequences it may not be for long video sequences, which include many heterogeneous scenes. There the perceived quality is as good as that of the *worst case* scene (*ie* the scene presenting the most coding artifacts). Traditionally, the assumed bandwidth resource is a constant-rate circuit which may be fully exploited. Since there is no penalty for using the full bandwidth, all bits are used, even on frames which may be acceptably coded at a rate lower than that available. With packet transport, it is possible through statistical multiplexing for the cells left unused by a video service to be used by another source or another service. Furthermore, it is possible to allocate such resources with acceptable (albeit statistical) reliability. The appropriate aim for VBR video coding is then to have constant (or more realistically, *consistent*) quality rather than to maximize quality subject to a resource constraint. This can be done with surprisingly little change to the current design process for codecs.

Some of the contributions of this work are as follows. In Section II, we compare VBR and CBR, from the point of view of rate, distortion and delay. While it is well known that VBR allows statistical multiplexing we show that this will *only* be possible if certain conditions on the input bit rates are met. We motivate the importance of the rate constraints imposed on the VBR connections as ultimately they will restrict the bit rates that each source can produce. We show that the key to the overall performance of the network is whether the capacity allowed by the rate constraints is fully used. Fully using the maximum allowed bit rate, a strategy we will denote *greedy* source coding, would preclude any significant statistical multiplexing. We propose two ways of preventing this worst case situation. First, in Section III we propose a *non-greedy* encoder design to maintain constant perceptual quality. The idea of non-greedy coding has been introduced before either directly, as part of the rate control algorithm [25, 15] or indirectly, as a feature of a constant distortion encoder (*eg* see [18]). The novelty here is that we include constant perceptual quality as an objective within the rate control algorithm. This non-greedy encoding scheme has the advantage of providing the same quality performance as greedy encoders for the more difficult scenes, while greatly reducing the bit rate in the easy scenes. We present experiments for a five minutes long, heterogenous video sequence containing a variety of scene types.

In Section IV, a second alternative proposes to resort to more complex rate constraints to ensure that the worst case scenario (*ie* that of a greedy coder) is not as damaging to the network. Recent work has also suggested increasing the constraints on the video streams in order to achieve reliable network operation without curtailing the potential benefits of VBR video [19]. In our approach we choose to use several leaky buckets which impose different rate constraints, each specifying the allowable rate at a given time-scale. All of these constraints have to be met in order for a given sequence to be admissible. Our approach has the twofold advantage of simple implementation and of resorting to a well known technique such as the leaky bucket as the basic building block.

II. COMPARISON OF VBR AND CBR VIDEO TRANSMISSION

In this section we compare the CBR and VBR transmission modes. Our goal is to describe their relative merits in terms of source coding (rate-distortion performance, end-to-end delay) and network efficiency (statistical multiplexing). A realistic VBR scenario will include the specification of a network-source contract and thus there will exist rate constraints on the source. In particular, we consider the class of constraints known as leaky buckets. We motivate that the key issues in the comparison are the rate constraints imposed on the video transmission and how tightly these constraints are met.

II.1. Delay vs.distortion trade-off.

In the buffered CBR case there is a simple, measurable trade-off between delay and distortion : for a given total rate, *ie* channel rate fixed, one can reduce the distortion by increasing the buffer size or, equivalently, the total delay.

(*) We emphasize that these constraints might be different from, although they should obviously be related to, the policing function implemented by the network. As specified in [6] the usage parameter control (UPC) or source policing function need not be known to the source. However, each source will have knowledge of the various rate constraints (on long term averages, bursts, etc.) that will be imposed.

Assume that the video encoder and decoder are connected through a CBR channel with rate R (in bit/s). The encoder outputs the coded bitstream to a buffer of size $B_{\max}$ (bits), while the decoder retrieves the bits to be decoded from a buffer of identical size. If r_i is the number of bits used for frame i then, in order for the bit rate sequence to be admissible, the encoder and decoder buffers should never be in overflow or underflow. It has been shown [15] that the choice of buffer size is directly related to the end-to-end delay in the system (see also [20]). If there is a delay of L frames between the time the encoder processes frame i and the time the decoder displays frame i then the buffer size at the encoder/decoder should be :

$$B_{\max} = LR\Delta t, \tag{1}$$

where Δt is the interframe interval in seconds (*).

For the buffer constraints to be met the bit rate generated by the source has to be such that (**) :

$$0 \leq \sum_{k=1}^{i} r_i - R\Delta t \leq RL\Delta t, \quad \forall i. \tag{2}$$

A detailed analysis of the constraints on buffering and delay can be found in [15].

The buffer control problem consists in choosing the bit rates r_i of each of the frames such that the conditions of (2) are met [21, 22]. Here we propose to use the solution presented in [20] which is optimal in an R-D sense. Although this solution assumes knowledge of the complete sequence to be coded, and thus could not be used in a real time implementation, it serves as a benchmark for other approaches and our results are thus relevant to general buffer control environments.

The problem is, given N frames and a discrete set of M available quantizers, to find a mapping x from the set of frames to the set of quantizers such that

$$x = \arg\min_{x} \left(\sum_{i=1}^{N} d_{ix(i)} \right), \tag{3}$$

subject to :

$$B_i \leq B_{\max}, \quad \forall i = 1, \cdots, N, \tag{4}$$

where :

$$B_i = \max(0, B_{i-1} + r_{ix(i)} - R\Delta t) \quad \text{with } B_1 = r_{1x(1)}, \tag{5}$$

and d_{ij} and r_{ij} are, respectively, the distortion and rate of frame i when quantizer j is used. This problem was solved using deterministic dynamic programming. Details can be found in [20].

(*) Here we ignore the transmission delay. If this delay were significant then $L = L_c + L_t$, where L_c is the delay due to coding and L_t is the delay due to transmission, and we would have $B_{\max} = L_c R\Delta t$.

(**) Note that the *no-underflow* constraint can always be met by adding filler bits.

It is important to note that : *the longer the delay, the looser the additional constraints.* For $L = 1$ ($B_{\max} = R\Delta t$) all frames have to use no more than the channel rate, *ie* for all i we have $r_i \leq R\Delta t$. Conversely, in the limit case of non real-time transmission, *ie* if $L = N$, the only applicable constraint is that of using a total bit budget of less than $NR\Delta t$. To summarize, we can state that :

Fact 1. *For a given source and* CBR *channel, distortion can be decreased by increasing the end-to-end delay in the system. The best* R-D *performance that can be achieved with channel rate R is obtained when $L = N$,* ie *there is only a constraint on the total bit rate budget.*

II.2. Comparison of CBR and constrained VBR.

We examine now the constraints imposed by a leaky bucket (LB) policing function on the source bit rate. A more detailed analysis of the constraints can again be found in [15]. We choose LB constraints for their simplicity and because they are the most widely considered constraints for VBR video.

A LB can be described as follows [17]. Each packet generated by the source has to receive a token in order to be transmitted. Assume packets have a size of P bits and that tokens are generated at a rate of R_t tokens per second. Furthermore, assume that we have a *bucket size* of N_t, *ie* the encoder can store at most N_t tokens. A LB constraint can thus be represented by the two parameters LB (N_t, R_t), or equivalently LB $(N_b, R_b) = LB(N_t P, R_t P)$ where the bucket size and rate are expressed in bit and bit/s respectively. For simplicity, in this section, we will use the latter notation. The leaky bucket policing mechanism requires the source to give up a token for every packet, *ie* every P bits it transmits. Thus, in order for the r_i bits corresponding to i-th frame to be transmitted, enough tokens have to be available. Denote N_i the accumulated number of *token bits* that are stored in the bucket after the i-th frame has been transmitted. We have that :

$$N_i = \min(N_b, N_{i-1} + R_b\Delta t - r_i), \tag{6}$$

since no more than N_b bits worth of tokens can be accumulated, and assume that the bucket is initially full ($N_0 = N_b$). The encoder can only transmit a packet if a token is available so that the constraint on the rate can be written as :

$$N_i \geq 0 \quad \forall i. \tag{7}$$

Now, denote $N_i' = N_b - N_i$, a counter of the bits transmitted in excess of the rate parameter of the LB. We can write the new constraint on N_i' as :

$$N_i' \leq N_b \quad \forall i, \tag{8}$$

and

$$\begin{aligned} N_i' &= N_b - \min(N_b, N_{i-1} + R_b\Delta t - r_i) \\ &= \max(0, N_{i-1}' + r_{ix(i)} - R_b\Delta t). \end{aligned} \tag{9}$$

We can see that we arrive with (8) and (9) at the same set of constraints from (2) when we have $N_b = B_{\max}$ and $R_b = R$. If the initial states are also the same (*ie* bucket full and buffer empty, respectively) then the two sets of constraints are equivalent. Therefore, the same set of techniques that were proposed for the buffer control problem in [20] can be used for the optimal bit allocation problem under LB constraints. However, there is one very significant difference between the two cases. In the CBR case, L represented a physical delay in the transmission system; in the VBR scheme with equivalent constraints on the bit rate $N_b = L$, no such delay need exist since the channel may be able to accept as many bits as required from each frame. Note how in Figure 1 the constraints in (b) and (c) are equivalent but the bitstream in (b) does not have to be buffered at the encoder and the end-to-end delay could be as short as one frame interval, assuming the network does not introduce additional delay. We can thus state (refer to [15]) :

Fact 2. *An advantage of a* VBR *environment (under a* LB *policing) with respect to an equivalent* CBR *environment is that, for the same number of bits used, the* VBR *system can reach the same level of distortion operating with shorter physical end-to-end delay.*

This advantage of VBR is important because it may allow to achieve quality levels which in a CBR environment would require end-to-end delays that might be unacceptable for some applications. We are assuming that the maximum rate per frame R_{link} that the *user loop* can handle is greater than the frame rates produced by the source ($r_i \leq R_{link}$). Although this implies that some of the capacity in the user loop is *wasted*, it is also true that transmission resources are cheaper in the user loop because distances are smaller [13]. Thus the end-to-end delay will be determined by the amount of delay introduced by the network. In an ideal situation statistical multiplexing will allow this delay to be short : the available bandwitdh within the network will be much larger than that required by each individual source and will be shared by a number of different sources. Of course, if buffering is needed or if capacity has to be deterministically assigned to each source then the VBR gain will not be as significant.

The key question in the VBR environment is whether, through statistical multiplexing, the network can provide this service with either shorter end-to-end delay or, alternatively, with less resources allocated to the transmission. Indeed, in the limit case where the network can only allocate a CBR connection of rate R it will have to buffer the source bit stream, so that both CBR and VBR cases produce the same delay : the only difference now lies in where the buffering is performed, the encoder/decoder or the network.

II.3. VBR vs. CBR : network aspects.

As was noted in the introduction, ATM networks allow various types of transmission modes to be used, including CBR and VBR. From a network perspective, CBR connections have the advantage of being easy to schedule since the required capacity is known *a priori*, but they tie down the resources for the duration of the transmission. On the other hand, VBR connections are said to provide greater flexibility because the network can dynamically re-allocate the transmission capacity to achieve a more efficient use of the available resources. Therefore, the question we can ask is : for the same transmission capacity, can VBR connections enable an increase in the number of users? We try to clarify the significance of the multiplexing gain and study the conditions that the source bit rate has to fulfill in order for this gain to exist. Note that we take a high level view of the allocation of resources. Although the final network performance depends not only on the total number of bits that are used but also on other factors such as specific queueing disciplines, we point out conditions on the overall bit rate that have to be met in order for statistical multiplexing gain to be achieved.

Consider a bit rate sequence $\mathcal{R} = \{r_i\}_{i=1}^N$, where R_i are the bits used by each of the N frames, which we want to characterize in terms of the resources required to transmit it. Consider the following two operators,

$$(10) \quad \mathcal{B}(\mathcal{R}, r) = \{B(r_i, r)\}_{i=1}^N \quad \text{where } B(r_i, r) = \max(B(r_{i-1}, r) + r_i - r, 0), \quad \forall i,$$

and

$$(11) \quad \mathcal{UB}(\mathcal{R}, r) = \{UB(r_i, r)\}_{i=1}^N \quad \text{where } UB(r_i, r) = UB(r_{i-1}, r) + r_i - r, \quad \forall i,$$

$\mathcal{B}(\mathcal{R}, r)$ represents the states of occupancy of a buffer filling up at rate $\mathcal{R}$ and emptying at rate r bits per frame. $\mathcal{UB}(\mathcal{R}, r)$ represents the occupancy of a virtual buffer that is allowed to underflow (and hence may have negative occupancy). The total number of filler bits that would have to be used in order to avoid underflow is thus :

$$(12) \qquad \mathcal{U}(\mathcal{R}, r) = \{B(R_i, r) - UB(R_i, r)\}_{i=1}^N,$$

ie $B(R_i, r) - UB(R_i, r)$ total filler bits have been used up to frame i. We also define :

$$(13) \qquad B_{\max}(\mathcal{R}, r) = \max_i (B(R_i, r)),$$

the maximum buffer size that is reached when transmitting sequence $\mathcal{R}$ at a bit rate r. If $\mathcal{R}$ is to be transmitted using CBR at rate r then buffers of size $B_{\max}$ will be needed and the end-to-end delay will be $B_{\max}(\mathcal{R}, r)/r$ frames.

Note also that, as was pointed out in Section II.2, since CBR and LB -constrained VBR have identical bit rate constraints we can use $\mathcal{B}$ to examine the *admissibility* of a sequence under several LB constraints. For instance, $\mathcal{R}$ is admissible under $LB(N_b, R_b)$ if :

$$(14) \qquad B_{\max}(\mathcal{R}, R_b) \leq N_b.$$

We note that for a given sequence $\mathcal{R}$ there are many LB constraints for which $\mathcal{R}$ is admissible. Also, $\mathcal{U}$ and

$B_{\max}$ can give us a measure of how loosely these constraints are met. For instance, if $U_N(\mathcal{R}, R_b) > 0$ then there were some *unused bits* since filler bits would have been needed in a CBR transmission. Similarly, if $B_{\max}(\mathcal{R}, R_b) > N_b$ there was some spare buffer capacity. In a CBR transmission we would typically have a rate $\mathcal{R}$ such that $U_i(R_b) = 0$ and $B_{\max}(\mathcal{R}, R_b)$ close to N_b (*ie* the buffer control algorithm would tend to (a) increase the source rate as needed to prevent underflow and (b) use the full buffer capacity to smooth out rate variations so that the buffer will be almost full at times).

We can now point out some facts about the network allocation.

Fact 3. *Given a sequence $\mathcal{R}$, each set of* LB *parameters for which the sequence is admissible represents a possible combination of network resources that would guarantee delivery of the sequence, at least under some queueing disciplines such as packet-by-packet generalized processor sharing* (PGPS) *[23]. If* LB(N_b, R_b) *is such a set of parameters, then the sequence could be transmitted over a channel of constant rate R_b, provided that buffers of size N_b exist within the network or at the encoder/decoder.*

Obviously the network need not allocate the bandwidth in a deterministic way to each of the sources. However, when several sources are considered simultaneously and are sharing a link of known capacity, Fact 3 can be used. More precisely, assume that two sources, $\mathcal{R}_1$ and $\mathcal{R}_2$, are sharing a link of rate r and that they have both a delay requirement so that the information corresponding to frame i cannot arrive after time $i + k$. Therefore, as seen in Section II.2, the constraint that the *combined* source, $\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2$, has to comply with is defined by LB(kr, r). Assume that r_1 and r_2 will be the required bit rates per frame for the sources to be transmitted independently with the same delay constraint of k-frames, *ie* $\mathcal{R}_1$ is admissible with LB(kr_1, r_1) and $\mathcal{R}_2$ is admissible with LB(kr_2, r_2). Then there is some statistical multiplexing gain (SMG) if $r < r_1 + r_2$. We can now state :

Fact 4. *A necessary condition for* SMG *to exist is that $\mathcal{U}_1(\mathcal{R}_1, r_1)$ and $\mathcal{U}_2(\mathcal{R}_2, r_2)$ are both strictly positive,* ie *that both sources underflow when transmitted at rates r_1 and r_2 respectively.*

Although the existence of underflow does not guarantee the occurrence of SMG (it also depends on when the underflow occurs) it is clear that the more the two sources underflow the larger the potential SMG can be. Note that an alternative way of expressing the SMG is that, if the link had bit rate $r_1 + r_2$, then we would have $B_{\max}(\mathcal{R}_1, r_1) > B_{\max}(\mathcal{R}_1 + \mathcal{R}_2, r_1 + r_2)$ and $B_{\max}(\mathcal{R}_2, r_2) > B_{\max}(\mathcal{R}_1 + \mathcal{R}_2, r_1 + r_2)$. In words, if the two sources shared the link, the end-to-end delay that each one experiences would be smaller.

To summarize, while VBR transmission allows statistical multiplexing to occur *it does not guarantee it.* Several VBR sources, which comply with various LB constraints, can *only* produce SMG if they do not meet strictly the constraint at all times! In [23] it is noted that *greedy* sources, *ie* those that use the maximum rate allowed by the LB constraints, also produce the worst case behavior by generating the highest delay in the system. From the above discussion multiplexing two greedy sources would not result in any gain. It is important to note that many of the results that have been presented in the literature on estimating SMG are obtained for non-greedy sources.

The next two sections will be devoted, respectively, to showing that greedy coding (*ie* generating bit rate sequences that are admissible under the agreed constraints but have nearly no underflow, $\mathcal{U}(\mathcal{R}, r)$ small) is not fully justified from a source coding point of view, and to present ways in which the network can minimize, through policing constraints, the effect of greedy coding strategies.

III. GREEDY VERSUS NON-GREEDY CODING

So far we have seen how the differences between CBR and VBR (*) come from the delay requirements, and how the multiplexing gain to be expected depends on whether the sources are greedy in their use of transmission resources. Here we indicate that some of the coding ideas based on traditional CBR coding will tend to produce greedy VBR sources. We show examples of how this can affect the overall system performance and motivate that non-greedy coding can be used to attain better multiplexing gains while losing relatively little video quality.

III.1. Coding design.

Traditionally, a constant bit rate (CBR) codec is designed by choosing a very complex test scene which should be coded at an acceptable quality level. The coding algorithms are chosen and fine-tuned using the test scene. Given that this scene (or several such test cases) give good results, then all simpler scenes will also yield good results with the given resources. With packet video, there is now a reward for not using all possible resources in every scene, since a large $\mathcal{U}(\mathcal{R}, r)$ will favor SMG. This changes the way the coder chooses the best level of quantization, but only for certain types of scenes.

(*) We emphasize that we are still referring to CBR/VBR *transmission.*

III.1.1. Types of scenes.

We can classify video scenes into four important types, as sketched in Figure 2. The first is the *test* scene. As with CBR codec design, this is a moderately difficult scene which is expected to have good quality at a specified rate. This scene identifies the acceptable target rate and SNR for the coder. The process of tuning algorithms to reduce artifacts and performing careful subjective studies on the test sequence remains unchanged for a VBR design.

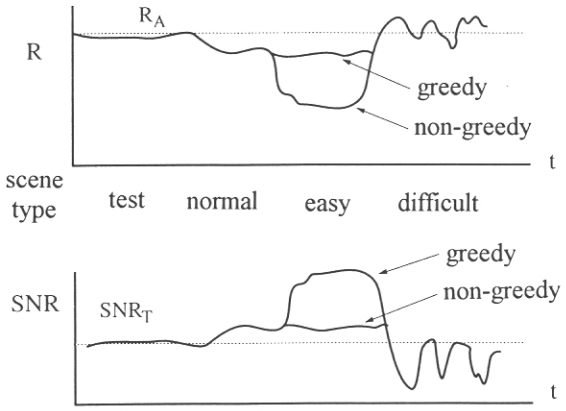

FIG. 2. — Rate and distortion behavior for a sequence containing four types of scenes : test, normal, easy, difficult. Note that the scales are not important, as the illustration is qualitative only. R_A represents the rate allocated in the network, and therefore the target coding rate for test scenes. SNR_T represents the target quality level.

Mesures du débit et de la distorsion pour une séquence contenant quatre types de scènes : test, normale, facile, difficile. L'échelle utilisée dans le graphe n'a pas d'importance ; on cherche à illustrer le comportement typique. R_A représente le débit réservé et donc l'objectif débit pour les scènes test. SNR_T représente l'objectif de qualité.

The second scene class we will denote as *normal* frames. These are comparable to the test scene, or a bit less complex, and basically require the full bandwidth resource. They result in a good quality level, *ie* a quality completely acceptable to the viewer for long periods.

The third type are the *easy* scenes, which are substantially simpler than the test sequence. For these scenes, there is an important difference between the CBR and VBR designs. A greedy CBR approach will use the available peak bandwidth and yield an SNR which is much higher than the target established by the test scenes. The optimization criterion is usually taken as the expected signal to noise ratio, E (SNR), which will indicate quality improvement even after the distortion drops below the level where the viewer becomes insensitive (or indifferent) to further picture refinement. Clearly, a greedy allocation of bandwidth, keeps the rate consistently very close to the peak, and allows no statistical multiplexing gain (SMG). A traditional approach using E (SNR) as a performance measure will lead to the conclusion that VBR has no advantage over CBR coding. An incorrect but common belief among codec designers, that excess bits can always be put to good use, is probably due to using only short, difficult test scenes, where resources are always scarce.

The fourth scene type consists of the *difficult* scenes. These should be very rare, because they are more difficult than the test scene and result in a noticeable degradation at the allocated rate. The techniques of buffer control, bit allocation etc., devised to minimize the perceived distortion under the rate constraint are as necessary for VBR coding as for CBR. It may be possible to exceed the allocated rate momentarily through a policing mechanism such as the leaky bucket. However this is completely equivalent to a larger smoothing buffer, and the known techniques still apply (see Section II.2).

III.1.2. Coding criteria : constant-Q, target-R and target-SNR.

To evaluate the idea of coding for a *target* SNR, rather than maximizing the overall SNR, we coded a five-minute sequence (7200 frames) from the movie *Star Wars* using monochrome JPEG coding [24] with 6 different quantization scales (0.1, 0.4, 0.8, 1.2, 1.6, 2.5). From the time series of rate $R(t)$, and quality, SNR(t) we can directly observe and identify the four scene types described above.

Figures 3–8 show $R(t)$ and SNR(t) for three rules governing the choice of quantizer for each frame. The first system (*constant-Q*, Fig. 3 and 4) has a constantly fixed quantizer level (0.4). This has often been cited as an easy way to generate VBR video, and is sometimes mistaken to be *constant quality*. As can be seen, over a long scene quality is not constant. The second case (*target-R*, Fig. 5 and 6) has a target rate, which makes it essentially like a simple CBR coding where there is no buffering between frames. We use the finest quantizer for which $R \leq R_{target}$. The final case (*target-*SNR, Fig. 7 and 8) has a target SNR, which yields consistent quality as closely as possible given the available quantizers. For each frame, we use the coarsest quantizer for which SNR $\geq$ SNR$_{target}$.

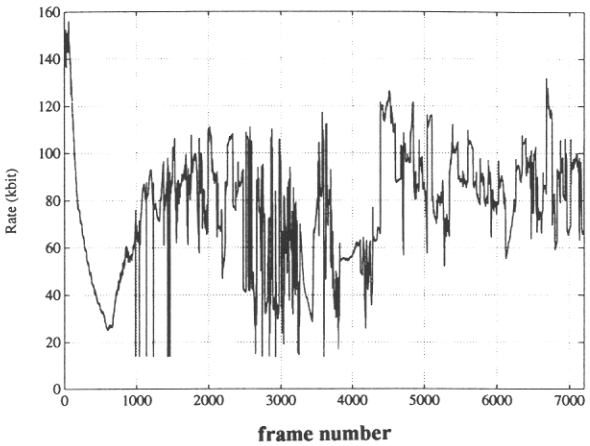

FIG. 3. — Rate time series with constant quantizer of 0.4, peak/mean rate = 156118,0/76482,6 = 2.04.

Série temporelle du débit avec quantification constante à 0,4. Rapport maximal à moyenne = 156118.0/76482.6 = 2,04.

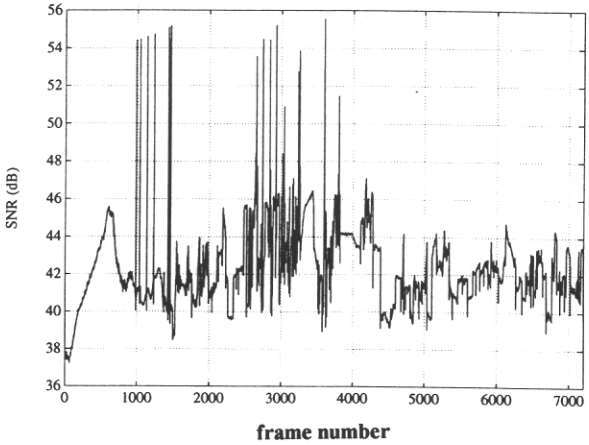

FIG. 4. — SNR time series with constant quantizer of 0.4, peak/mean dist = 16345.8/5545.4 = 2.95.

Série temporelle de la SNR avec quantification constante à 0,4, rapport maximal à moyenne en distorsion = 16345,8/5545,4 = 2,95.

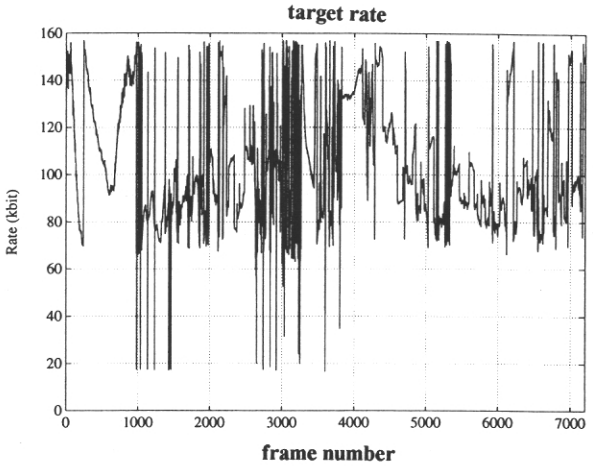

FIG. 5. — Rate time series with target rate of 157 000 bit/frame, peak/mean rate = 156994.0/105110.8 = 1.49.

Série temporelle du débit avec un objectif de débit de 157 000 bit/trame, rapport maximal à moyenne = 156994,0/105110,8 = 1,49.

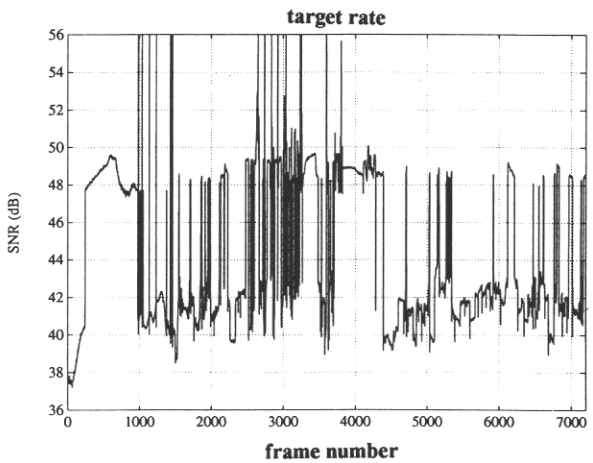

FIG. 6. — SNR time series with target rate of 157 000 bit/frame, peak/mean dist = 16345.8/4614.1 = 3.54.

Série temporelle de la SNR avec un objectif de débit de 157 000 bit/trame, rapport maximal à moyenne en distorsion = 16345,8/4614,1 = 3,54.

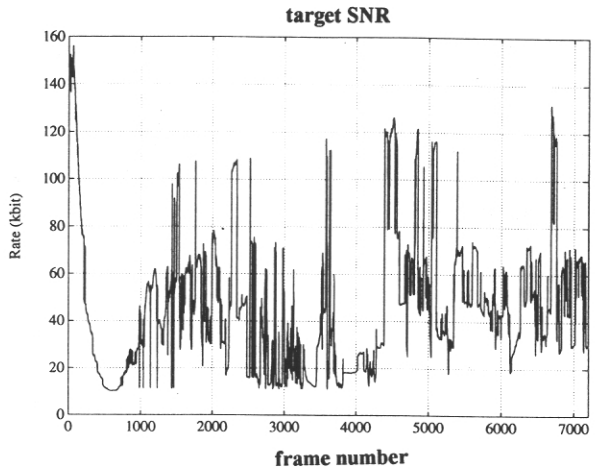

FIG. 7. — Rate time series with target SNR of 36.8 dB, peak/mean rate = 156118.0/46070.8 = 3.39.

Série temporelle du débit avec un objectif de SNR de 36.8 dB, rapport maximal à moyenne = 156118,0/46070,8 = 3,39.

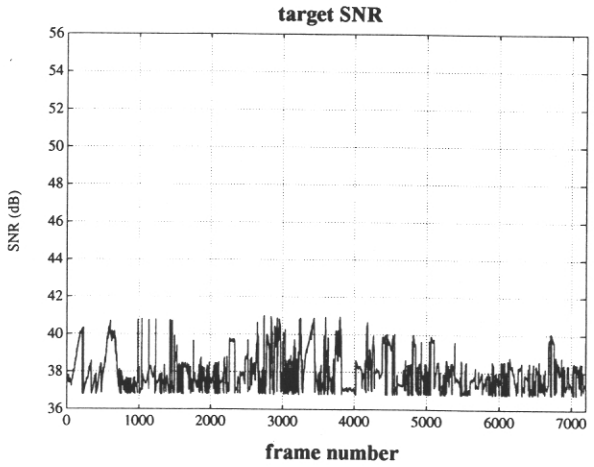

FIG. 8. — SNR time series with target SNR of 36.8 dB, peak/mean dist = 17998.4/14120.2 = 1.27.

Série temporelle de la SNR avec un objectif de SNR de 36,8 dB, rapport maximal à moyenne en distorsion = 17998,4/14120,2 = 1,27.

The first sixty frames include a sequence of text near the beginning of the movie, and represent the worst case of the 5-minute series. We use this as the *test scene*, to determine the tradeoff between R and SNR. The most interesting and striking feature of these three schemes is that the (worst case) rates and distortions for this test scene are practically identical. In this sense, they are equivalent in quality.

The constant-Q system makes a good reference, since it indicates the natural frame complexity. Observing the target-R system in comparison, it is clear that many *easy* frames have their rate boosted to the allocated (*ie* CBR) level and their distortion is lowered far below the required level of the test scene. The target-SNR system, in contrast, keeps the distortion very close to a constant level, while its rate is somewhat more bursty than the constant-Q system; the peak is the same and the mean is lower.

Since the target-R system peak to mean ratio is 1.5, we conclude that 33% of the bandwidth used by a CBR packet video service can be made available by simply allowing the smoothing buffer to underflow. (Timing recovery — which is the only benefit of padding to avoid underflow — can be done explicitly through side information. This is necessary in the presence of cell loss in any case.) Another 38% can be recovered by choosing quantizers by a target-SNR rule instead of a target-R rule. The target-SNR trace shows that only the remaining 29% of the bandwidth is utilized for necessary video information. The network bandwidth allocation (*eg* the leaky bucket rate parameter), though, still has to be set at the peak rate for the test scene. The *recovered* bandwidth is only available through statistical multiplexing in the parts of the network where several sources share a link. See Table I for a summary of the results. In terms of the notation introduced in the previous section it can be seen that of the three rate sequences, $\mathcal{R}_R$, $\mathcal{R}_Q$, $\mathcal{R}_{\text{SNR}}$, the largest is $\mathcal{U}(\mathcal{R}_{\text{SNR}}, 156)$.

In this movie, there are three or four scenes more difficult than our test scene [13] (*ie* they require higher rate or result in higher distortion). The algorithms used to optimize the coding of such scenes under tight resource constraints remain the same for VBR as for CBR codec design. We treat only intra-frame coding here because we can conveniently make a frame-wise choice of $R(t)$ and $\text{SNR}(t)$ from the six choices of Q. For mixed inter/intra-frame coding (*eg* MPEG), the smoothing buffer averages bandwidth across frames coded with two or three different algorithms. The main result, that non-greedy quantization allows substantial statistical multiplexing gain while retaining allocated rate and an upper bound on distortion, is still valid.

III.2. Rate control for non-greedy coding.

In this section we examine the network resource allocation and policing function. The leaky bucket (LB) mechanism alone does nothing to promote non-greedy coding. The network can, however, provide proper mechanisms and incentives to ensure good SMG without precluding consistently good quality video.

In order to allocate resources in the network, there has to be some description of the traffic generated by a source and the performance required of the network. The leaky bucket is a reasonable mechanism for specifying such an agreement, *not* because it specifies the mean rate (*eg* for billing) and the size of substantial peaks which are somehow reliably multiplexed; but because it can be used to specify an allocated rate for the single source (which is close to the peak rate), and a bound on the jitter imposed by network queues, cross traffic etc.

Since the policing mechanism regulates the coder output by dropping the violating cells, it makes sense to incorporate this function into the coder rate control algorithm (*). The leaky bucket however, as was seen in Section II.2, presents the same constraints on the bit rate as buffer constrained CBR environment of same rate (although a LB constraint does not necessarily introduce a delay).

In the previous example the test scene corresponded to the most difficult scene in the sequence. To compare greedy and non-greedy coding for scenes more difficult than the test scene (*ie* with relatively scarce resources), we choose $R = 60$ kbit/frame and target SNR $= 41$ dB. The examples of Figures 9 and 10 compare non-greedy and greedy buffer control strategies. The greedy buffer control is the optimal bit allocation formulated in Section II.1 [20], which maximizes the average SNR for the given rate (here given by the LB rate parameter), and the constraint that the buffer (given by the LB bucket size) does not overflow. As was seen in Section II.2, the same techniques used for buffer-constrained CBR transmission can be used for LB-constrained VBR transmission. The non-greedy version has a simple modification which enforces an upper bound for the SNR per frame (of about 41 dB). The basic idea of non-greedy coding has been proposed before in rate control for ATM transmission of video [25, 15]. The novelty here is that we express the constraint in terms of distortion, rather than as a lower

(*) Incorporating the policing mechanism into the encoder does not require that the policing function state be fedback from the network. If the function is agreed upon then both network and source can implement it in parallel.

TABLE I. — Summary of the results of using the three approaches, constant-Q, target-R and target-SNR. Note that in all three cases the worst case frame is allocated the same rate of 156 kbit/frame.

Résultats obtenus en utilisant les trois méthodes. On remarque que dans les trois cas l'unité qui correspond au pire cas reçoit le même débit de 156 kbit/trame.

strategy	peak rate (kbit/frame)	mean rate (kbit/frame)	peak/mean rate	peak/mean distortion
constant-Q	156	76.4	2.04	2.95
target-R	156	105	1.49	3.54
target-SNR	156	46	3.39	1.27

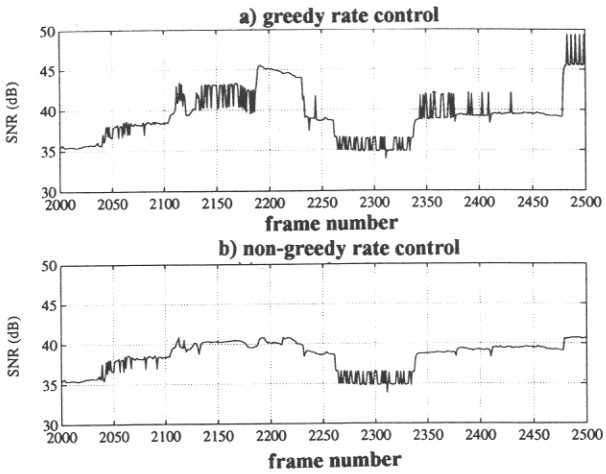

FIG. 9. — SNR trace with (a) greedy and (b) non-greedy rate control. Note that the SNR remains the same for the most difficult scene, but does not exceed the target for easier scenes in the non-greedy case.

Série temporelle de SNR avec les algorithmes de contrôle de débit (a) glouton et (b) non glouton. On remarque que le SNR est maintenu constant pour la scène la plus difficile, mais ne dépasse pas l'objectif pour des scènes plus faciles dans le cas non glouton.

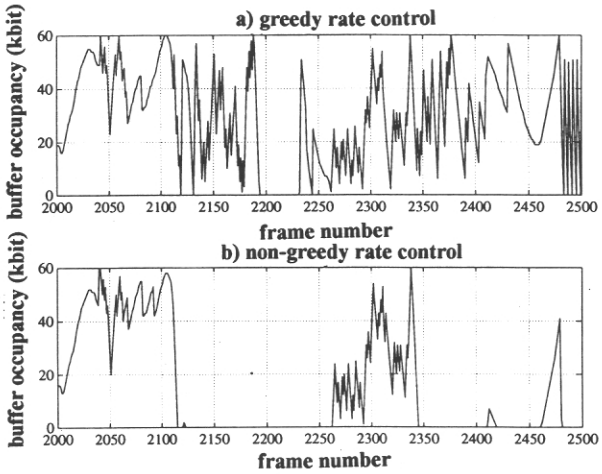

FIG. 10. — Buffer occupancy trace with (a) greedy and (b) non-greedy rate control. Note substantial buffer underflow for easy scenes in non-greedy case.

Série temporelle de l'occupation du tampon avec un contrôle de débit (a) glouton et (b) non-glouton. On remarque que dans le cas non glouton le tampon se vide souvent.

bound in the quantizer stepsize as in [25, 15] so we can easily accommodate the non-greedy requirement within our optimization framework.

For the greedy algorithm, the SNR changes depending on the scene complexity (see Fig. 9a) while the buffer is almost never in underflow (see Fig. 10a). By contrast, the non-greedy version produces much more consistent quality (see Fig. 9b), while using less network resources as shown by the frequent occurrence of buffer underflow (see Fig. 10b). For the full 5 minute segment (with buffer size of 120 kbit), the mean buffer output rate is only 46 kbit/frame for the non-greedy case compared to 59 kbit/frame for the greedy optimization (*ie* a 33% reduction in average rate). The SMG is necessarily less in this case than in the previous target-SNR example because we have chosen an operating point with higher target SNR and a lower rate constraint. This results in more *difficult* scenes and fewer *easy* scenes. Note (see Fig. 9) that the non-greedy algorithm reduces the SNR for those scenes above the target SNR but maintains it for those scenes near or below the target SNR. In other words, the non-greedy version of the algorithm *does not affect the worst case or the difficult scenes.*

To encourage users to adopt a non-greedy coding algorithm may require a different price structure than that for fixed bandwidth circuits. The network reuses some of the (peak) bandwidth allocated to the user, so the benefit could be returned to the user in the form of a lower tariff. Just as other aspects of coding and networking are standardized, so the statistical behavior of video traffic can be agreed upon, monitored and enforced. By definition, it is not possible (as some may wish) to enforce statistical agreements instantaneously. However it is surely possible to design and operate a communications system with statistical traffic enforcement. Statistical multiplexing only occurs where there are several sources sharing a resource. Therefore we should expect this to be reflected in traffic description; *ie* as the number of sources (N) increases, the bandwidth allocated for each source R_a, decreases with N. The function $R_a(N)$ depends on the nature of the source [13].

IV. RATE CONSTRAINTS USING MULTIPLE LEAKY BUCKETS

Our results have shown that a *bounded-rate*, non-greedy VBR coding scheme is practically equivalent to CBR coding in the sense of having the same allocated peak rate and distortion level on the test scene. In the VBR case, the difference between allocated and mean bandwidth may be recovered statistically by the network.

The previous section has shown how encouraging the use of non-greedy coding techniques can provide appropriate multiplexing gain while maintaining the video quality for the most difficult scenes. Our point of view in this section is to assume that it may not be always realistic to expect sources to behave in a non-greedy fashion. We consider therefore constraints that will outperform simple LB in the worst case scenario of greedy coding. While other alternatives to LB policing having proposed [19, 26], we concentrate here on a solution using multiple leaky buckets due to the simplicity of their implementations and the fact that this mechanism is well understood. As in the rest of the paper we assume that sources have sufficient knowledge of the rate constraints to be able to adjust their own traffic.

IV.1. Worst case bursts under LB constraints.

We define as before a sequence as being admissible when it does not violate a certain policing function. Then for a given single LB constraint the admissible sequences can be very different. Consider LB(N_f, R) where R is expressed in bits per frame as before, but we now use $N_f = N_b/R$ (in frames) as the dimension of the bucket. Thus, a larger N_f means that the LB averages the source bit rate over more frames. A sequence where every frame uses R bits is admissible under LB(N_f, R) and, similarly, a sequence that uses $N_f R$ bits for every N_fth frame and zero bits for those in between is also admissible. Obviously, these are extreme cases but they indicate that sources with varying degrees of *burstiness* can be admissible.

To better understand the trade-offs involved, we define a curve that can describe the *worst case* performance of LB policing mechanism. We plot $R_{\max}(i)$ which we define as the maximum average rate that can be used without violating the policing function over a window of size i. We assume that the bucket is initially full, so that there is credit to transmit $N_f R$ bits. Then we have that :

$$R_{\max}(1) = N_f R, \tag{15}$$

$$R_{\max}(2) = (N_f R + R)/2 = (N_f + 1)R/2, \tag{16}$$

and in general

$$R_{\max}(i) = (N_f R + i - 1)R/i. \tag{17}$$

Obviously, as the window over which we measure the rate grows (i increases) the maximum average rate comes closer to the transmission rate of the LB, R. Such a curve for a given set of LB parameters provides a bound on the rate that a source can use without violating the policing constraint (see Fig. 11 for an example).

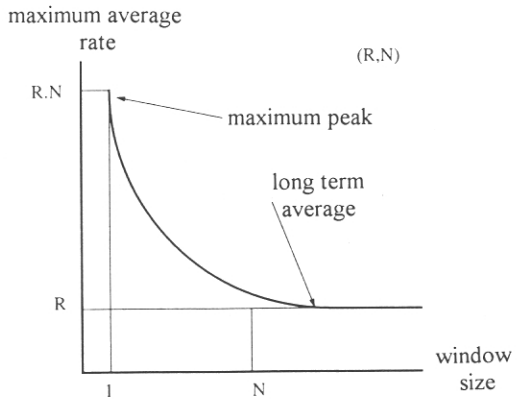

FIG. 11. — Worst case average rate for several window sizes, as an example : the point (1, *RN*) indicates that the maximum bit rate that can be used by a single frame is *RN*. The range of average rates allowed under the constraint is represented by the area under the curve.

Débit moyen dans le pire cas pour plusieurs fenêtres d'observation, par exemple : le point (1, RN) indique que le débit maximal qui peut être utilisé par une seule trame est RN. L'intervalle des débits permis sous la contrainte correspond à l'aire sous la courbe.

The question we are seeking to answer is : how do we choose the leaky bucket parameters so that we minimize the effect on the network performance of worst case bursts? Assume we are given a sequence of bit rate $\mathcal{R}$ and we want to adjust the LB parameters to make the sequence admissible, *ie* choose N_f, R so that the LB constraint is not violated. As mentioned earlier there are many possible choices of LB parameters. We have the following trade-off :

- if a large N_f is chosen then the required bit rate $\mathcal{R}$ can be close to the average rate of the sequence and thus relatively low. However, a source constrained by that LB could send to the network (large) bursts of size up to $N_f R$ bits;
- conversely, if a small N_f is chosen then the required bit rate R will be higher (in the limit case of $N_f = 1$, as high as the highest frame rate in the sequence) whereas the product $N_f R$ would be relatively small.

IV.2. Multiple leaky buckets.

The above trade-off has been noted in the literature on policing functions [17] although here we look at it in a deterministic, rather than stochastic, fashion. In [17] the fact that the maximum peak rate increases as the window size N increases was seen as a justification to police only the, loosely defined, peak rate, *ie* measure the average bit rate over short time windows. Here we propose that combining two or more LB can be an easy way of maintaining a long term monitoring while not risking very sharp peaks in bit rate.

As an example, consider the two leaky bucket case. We now require that, in order to be admissible, every packet generated by the source has to obtain two tokens, one each from each LB, so that *both* constraints have to be met. Assume LB of sizes N_1, N_2 and leak rates R_1, R_2. Then, clearly, if we choose the parameters such that : $N_1 > N_2$, $R_1 < R_2$ and $N_1 R_1 > N_2 R_2$ we can achieve our goal of limiting the peak size. In this example, the maximum constant rate would be R_1, while the maximum peak would be $N_2 R_2$.

By choosing two different window sizes as N_1, N_2 we ensure that the two problems mentioned above are not encountered, *ie*, referring to Figure 12 we have that,

- the maximum admissible constant rate is $R_1 < R_2$ so that the long term average has to be kept relatively small, but
- the maximum instantaneous admissible rate is $R_2 N_2 < R_1 N_1$ so that the amplitude of the peaks is limited.

Our main motivation for adding more constraints is to ensure that in the worst case scenario, *ie* when the source uses as many bits as permissible, the bit rate used by the source is smaller (either the peak or the long term average) than in the single LB case (see Fig. 11 and 12). A double LB scheme allows the same peak rate (resp. average rate) but with a smaller long term average (resp.

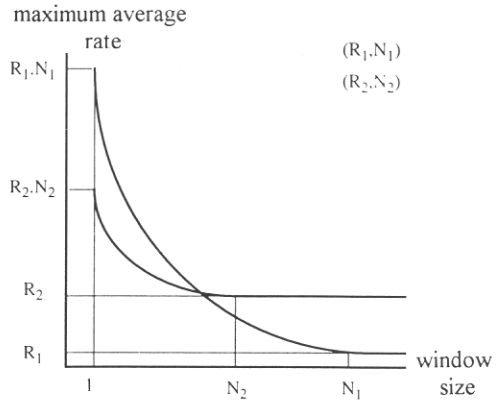

FIG. 12. — Motivation for using a double leaky bucket. The worst case short term behavior is determined by the short bucket, while the long term average is set by the long bucket. As before the range of admissible average rates is represented by the area under whichever curve is closer to the *x*-axis, for a given window size.

Motivation pour l'utilisation d'un réservoir de jetons double. Le pire cas à court terme est déterminé par le réservoir le plus court, tandis que le comportement à long terme est défini par le réservoir long ; comme auparavant l'intervalle des débits permis sous la contrainte correspond à l'aire sous la courbe en dessous de celle des deux courbes qui se trouve plus près de l'axe x pour une taille de la fenêtre donnée.

peak rate) than an equivalent single LB scheme. As in the non-greedy coding example, a double LB can be matched so that the peak rate needed for the worst case scene is allowed while the rate used in the *easier* parts of the sequence is limited.

We now show an example using the coding examples of Figure 10 to choose the appropriate parameters for the LB. We consider two separate single LB schemes. First a short window LB, LB(3,60), is chosen, see Figure 13. Here the maximum allowable peaks are small (180 kbit/frame) whereas the long term average is 60 kbit/frame. Thus the danger is that a greedy source could use continuously 60 kbit/frame. Indeed, the greedy source of Figure 10a would be admissible under these constraints. Conversely one could choose a longer window LB, LB(60,55), (see Fig. 14) where the long term average would be kept lower (55 kbit/frame). However the danger here is that a source could be admissible while generating a peak rate of up to 3300 kbit for one frame.

When the two LB are combined, see Figure 15, we observe that the unwanted properties of each of the single LB schemes are avoided. Thus the maximum short term peak is kept small, as is the long term average. Note that, under the double LB scheme, the greedy sequence of Figure 10a would *not* be admissible, while the non-greedy sequence of Figure 10b would be.

The idea of having multiple LB corresponds to monitoring the bit rate at different time scales [2]. The network would tend to be concerned about short term (a few packets) measures of rate, and would find longer term measures (a few frames) impractical since they would not be enforceable. Conversely for video coders, where the bit rate changes widely both within a frame and between successive frames, short term measures are not so meaningful (and can be easily met by internal buffering),

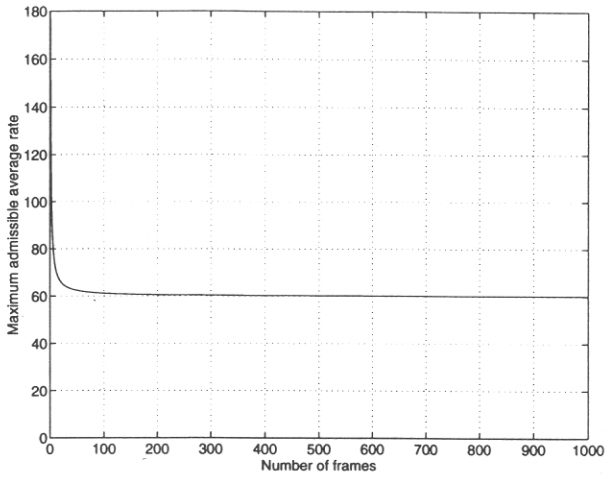

FIG. 13. — Worst case burst curve for a LB(3,60) that has been chosen for the non-greedy source of Figure 10b. The window is short (N = 3) and thus the leak rate has to be large enough to permit the larger frames to be sent. The drawback is that the long term average is 60 kbit/frame, while the actual sequence's average was 46.3 kbit/frame. The greedy sequence of Figure 10a would also be admissible.

Courbe du pire cas pour un réservoir à jetons (3,60) qui a été choisi pour la source non glouton de la Figure 10b ; la fenêtre d'observation est courte (N = 3) et donc le débit du réservoir à jetons doit être élevé pour permettre que l'on puisse transmettre les trames qui utilisent plus de débit ; le désavantage est que la moyenne de débit à long terme est de 60 kbit/trame, tandis que la vraie moyenne de la séquence était 46,3 kbit/trame. La séquence glouton de la figure 10a serait donc admissible aussi.

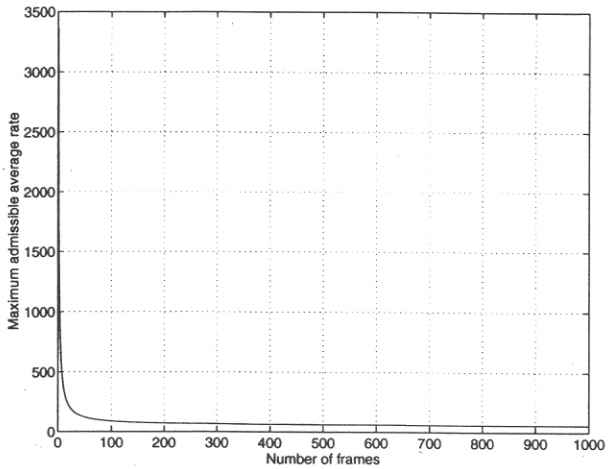

FIG. 14. — Worst case burst curve for a LB(60,55) ; the non-greedy sequence of Figure 10b is also admissible under this LB ; note that the longer window N_f = 60 enforces a lower long term average ; however, there is the danger that a compliant source may generate bursts of up to 3000 kbit/frame !

Courbe du pire cas pour un réservoir à jetons (60,55) ; la séquence non glouton de la figure 10b est aussi admissible avec ce LB ; on remarque qu'une fenêtre plus longue N_f = 60 permet de forcer une moyenne à long terme plus petite ; cependant, il existe le danger qu'une source puisse transmettre un pic de 3 000 kbit/trame tout en vérifiant la contrainte !

while a long term rate constraint with large buffers is well understood. Our approach is thus to maintain both types of constraints as they serve different goals.

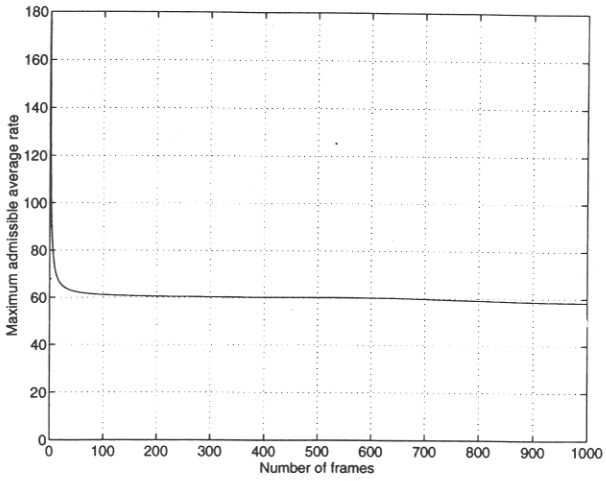

FIG. 15. — Effect of combining two LB's ; the resulting worst case burst curve shows both the lower long term average (which tends to 55 kbit/frame) and smaller short term bursts (less than 180 kbit/frame).

Résultat de la combinaison des deux réservoirs à jetons ; la courbe de pire cas obtenue a simultanément la moyenne à long terme plus petite (qui tend vers 55 kbit/trame) et des pics à court terme plus limités (moins de 180 kbit/trame).

V. CONCLUSIONS

We have examined the problem of rate control for video encoders designed for transmission over packet networks. The main point is that if the overall performance is to be improved, techniques different from those used in CBR coding may be required. One approach to reach this goal is to have encoders with rate controllers designed for these specific VBR requirements (non-greedy encoders). We have presented coding results showing how a non-greedy strategy can provide increased SMG while maintaining the same quality as a greedy scheme for the most difficult scenes. We also propose an alternative solution which relies on increasing the constraints of the policing function. An example of this approach involving the use of two leaky buckets has also been presented.

ACKNOWLEDGMENTS

M. W. Garrett would like to thank R. Ansari, A. Fernandez and S. Singhal of Bellcore for useful technical discussions.

Manuscrit reçu le 30 mars 1995.

REFERENCES

[1] ORTEGA (A.), GARRETT (M. W.), VETTERLI (M.). Toward joint optimization of VBR video coding and packet network traffic control. *Proc. of the 5th Packet Video Workshop*, Berlin (March 1993).

[2] ORTEGA (A.), VETTERLI (M.). Multiple leaky buckets for increased statistical multiplexing of ATM video. *Proc. of the 6th Packet Video Workshop*, Portland, OR (Sep. 1994).

[3] ELEFTHERIADIS (A.), PEJHAN (S.), ANASTASSIOU (D.). Algorithms and performance evaluation of the Xphone multimedia communication system. *Proc. of the ACM Multimedia 93 Conf.*, Anaheim, CA (Aug. 1993), pp. 311-320.

[4] MACEDONIA (M.), BRUTZMAN (D.). Bone provides audio and video across the Internet. *Computer* (Apr. 1994), **27,** pp. 30-36.

[5] BOLOT (J.-C.), TURLETTI (T.). A rate control mechanism for packet video in the internet. *Proc. of Infocom'94*, Toronto (June 1994), pp. 1216-1223.

[6] ***. ATM Forum. ATM *user-network interface specification, Version 3.0.* Prentice-Hall (1993).

[7] MAGLARIS (B.), ANASTASSIOU (D.), SEN (P.), KARLSSON (G.), ROBBINS (J.). Performance models of statistical multiplexing in packet video communications. *IEEE Trans. COM* (July 1988), **36,** pp. 834-843.

[8] SEN (P.), MAGLARIS (B.), RIKLI (N.), ANASTASSIOU (D.). Models for packet switching of variable-bit-rate video sources. *IEEE J. SAC* (June 1989), **7,** pp. 865-869.

[9] DOUGLAS (P.), VETTERLI (M.). Statistical analysis of the output rate of two variable bit-rate video coders. *Proc. of the 3rd Packet Video Workshop*, Morristown, NJ (March 1990).

[10] VERBIEST (W.), PINNOO (L.), VOETEN (B.). The impact of the ATM concept on video coding. *IEEE J. SAC* (Dec. 1988), **6,** pp. 1623-1632.

[11] ***. Special issue on packet speech and video. *IEEE J. SAC* (Apr. 1991).

[12] ***. Special issue on packet video. *IEEE Trans. CSVT* (June 1993).

[13] GARRETT (M. W.). Contributions toward real-time services on packet switched networks. *PhD thesis*, Dept. of Electrical Eng., Columbia Univ. (1993).

[14] RIGOLIO (G.), VERRI (L.), FRATTA (L.). Source control and shaping in ATM networks. *In : Globecom'91*, Phoenix (1991).

[15] REIBMAN (A. R.), HASKELL (B. G.). Constraints on variable bit-rate video for ATM networks. *IEEE Trans. CAS* (Dec. 1992), **2,** pp. 361-372.

[16] TURNER (J. S.). New directions in communications (or which way to the information age ?). *IEEE COM Mag.* (Oct. 1986), **24,** pp. 8-15.

[17] RATHGEB (E. P.). Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. SAC* (Apr. 1991), **9,** pp. 325-334.

[18] DARRAGH (J. C.), BAKER (R. L.). Fixed distortion subband coding of images for packet-switched networks. *IEEE J. SAC* (June 1989), **7,** pp. 789-800.

[19] HEEKE (H.). A traffic-control algorithm for ATM networks. *IEEE Trans. CSVT* (June 1993), **3,** pp. 182-189.

[20] ORTEGA (A.), RAMCHANDRAN (K.), VETTERLI (M.). Optimal trellis-based buffered compression and fast approximations. *IEEE Trans. IP* (Jan. 1994), **3,** pp. 26-40.

[21] ZDEPSKY (J.), RAYCHAUDHURI (D.), JOSEPH (K.). Statistically based buffer control policies for constant rate transmission of compressed digital video. *IEEE Trans. COM* (June 1991), **39,** pp. 947-957.

[22] CHEN (C.-T.), WONG (A.). A self-governing rate buffer control strategy for pseudo-constant bit rate video coding. *IEEE Trans. IP* (Jan. 1993), **2,** pp. 50-59.

[23] PAREKH (A. K.), GALLAGER (R. G.). A generalized processor sharing approach to flow control in integrated services networks : the single node case. *IEEE/ACM Trans. N* (June 1993), **1,** pp. 334-357.

[24] ***. JPEG technical specification : Revision (Draft), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII (Aug. 1990).

[25] LEDUC (J.-P.), AGOSTINO (S. d'). Universal VBR videocodecs for ATM networks in the Belgian broadband experiment. *Image Communication* (June 1991), **3,** pp. 157-165.

[26] SKELLY (P.), SCHWARTZ (M.), DIXIT (S.). A histogram-based model for video traffic behavior in an ATM multiplexer. *IEEE/ACM Trans. N* (Aug. 1993), **1,** pp. 446-459.

BIOGRAPHIES

Antonio ORTEGA received the Telecommunications Engineering degree from Universidad Politécnica de Madrid, Madrid, Spain in 1989 and the PhD in Electrical Engineering from Columbia University, New York, in 1994. His PhD work 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 at the Electrical Engineering-Systems Department, University of Southern California. His research interests are in the area of digital image and video communication systems, including adaptive and multiresolution compression techniques, as well as joint source-channel coding for transmission over broadcat channels and packet networks.

Mark W. GARRETT received the BS, MS and PhD degrees in electrical engineering from Columbia University in 1982, 1984 and 1993, respectively. In 1984 he joined the research staff at Bell Communications Research (Bellcore). His work there has included LAN protocols and prototyping, network simulation and analysis, packet voice/video, and traffic management. Dr Garrett designed the first LAN media access control chip to operate at over 100 Mbit/s.

Martin VETTERLI received the Dipl. El-Ing degree from ETH Zürich, Switzerland, in 1981, the MS degree from Stanford University in 1982, and the Doctorat ès Science degree from EPF Lausanne, Switzerland, in 1986. He was a Research Assistant at Stanford and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University in New York where he was last an Associate Professor of Electrical Engineering. Since July 1993, he is on the faculty of the Dept. of EECS at the University of California, Berkeley where he is a Professor of Electrical Engineering. In July 1995, he joined the Swiss Federal Institute of Technology in Lausanne, Switzerland as a Professor in Communication Systems. His research interests include wavelets, multirate signal processing, computational complexity, signal processing for telecommunications and digital video processing and compression.