

Chapter 3

Rate Constraints for Packet Video Transmission Based on Joint Source/Network Criteria ¹

Contents

3.1	Introduction	75
3.2	Comparison of VBR and CBR video transmission	79
3.3	Greedy versus Non-Greedy coding	86
3.4	Single and double leaky buckets	97
3.5	Conclusions	102

3.1 Introduction

Variable bit rate (VBR) transmission of video through packet networks represents a departure from traditional problems in both the networking and coding fields. Two major advantages are often cited for VBR transmission: (a) constant video quality due to removal of buffer constraints; (b) more efficient use of network bandwidth through statistical multiplexing.

¹Parts of this chapter represent joint work with Mark W. Garrett. For related publications see [70, 76]

Recent implementations of packet video transmission have been reported for video over local area networks [29] or video multicasting over the Internet [62, 10]. 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 was not changed, the information might sometimes, e.g. 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 offered by ATM networks [5]. Integration of video within ATM networks is an active area of research where one of the problems encountered so far has been the lack of interaction between the network and source coding fields.

Analyses of network performance have tended to assume that a video source could be characterized by a more or less elaborate probabilistic model [63, 98, 27, 109, 2, 3], while work on encoding schemes for packet video coders has tended to see the network as a black box, as determined by the source policing interface [3]. The announced performance gains due to multiplexing could then be achieved provided that the sources behave according to the chosen model. This type of analysis 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 [36] and, more importantly, (ii) the models do not take into account that for a given network constraint the source will be using some kind of rate control. Fig. 3-1 illustrates the idea of “self-regulating coders” [92, 91]. Typically, models tend to characterize sources operating with a “constant quantizer” mode (i.e. using the same codebook for every frame as in Fig. 3-1(a)). However, the rate control needed in order to avoid violating the transmission constraints agreed upon by network and user as part of the contract negotiated at

Figure 3-1: 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.

Clearly, much of the research and standards work related to packet video thus far treats the coding and network sides of the problem separately. However, if we simply define an interface, such as the so-called ATM policing function, as a strict agreement between the coder and the network, and allow each side to do greedy one-sided optimization, then the potential advantage of variable rate transmission

will be lost. In this chapter we identify the advantages of VBR video transmission, show how the codec design problem changes under the new constraints of variable rate transmission, and propose that to realize the advantages of ATM, we must understand the joint design and optimization of the video coder and network traffic management for video services. (Once we understand the joint problem, then it is useful to segment the design into modules with clean interfaces.)

Traditional video coding is greedy in that the performance criterion which is maximized is the average signal-to-noise ratio (SNR) or perceptual quality. 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 (but 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.

The outline of the chapter is as follows. In Section 3.2 we first compare CBR and VBR transmission from a source coding point of view and show how VBR is advantageous, even when both methods assign exactly the same bit rate for each frame. We then look at the network side of the problem and show that expectations of statistical multiplexing gain are based on the assumption that VBR connections will not fully utilize the maximum admissible capacity, as determined by a policing function. In Section 3.3 we show that the greedy source coding traditionally used in CBR is no longer appropriate for VBR environments. We motivate that non-

greedy strategies provide the same quality for the most difficult scenes while freeing up transmission resources for other users during the easy scenes. Alternatively, in Section 3.4 we show that the danger posed by greedy source coding can be limited by resorting to schemes where the policing function constraints the bit rate at several time scales, e.g. using multiple leaky buckets.

3.2 Comparison of VBR and CBR video transmission

In this section we approach the comparison of CBR and VBR transmission by looking at both source coding and network transmission aspects. Since, VBR transmission of video has been said to provide advantages both in terms of video quality and network efficiency, our aim is to clarify under what conditions these goals can be achieved.

3.2.1 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, i.e. channel rate fixed, one can reduce the distortion by increasing the buffer size or, equivalently, the total delay. (See Section 2.5.1 and Fig. 2-14)

Assume that the video encoder and decoder are connected through a CBR channel with rate R . The encoder outputs the coded bitstream to a buffer of size B_{max} , while the decoder retrieves the bits to be decoded from a buffer of identical size. If R_i is the bit rate 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 [91] that the appropriate buffer size to be used is directly related to the end-to-end delay in the system (see also Section 2.2.2). 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:

$$B_{max} = L \cdot R. \quad (3.1)$$

For the buffer constraints to be met the bit rate generated by the source has to be such that²:

$$0 \leq \sum_{k=1}^i R_k - R \leq R \cdot L, \quad \forall i. \quad (3.2)$$

A detailed analysis of the constraints on buffering and delay can be found in [91].

The buffer control problem, i.e. the problem of choosing the bit rates of each of the frames such that the conditions of (3.2) are met, has been studied in the literature. Here we propose to use the optimal solution presented in Chapter 2. 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 to be solved is that of, given N frames and M available quantizers, finding a mapping x from the set of frames to the set of quantizers such that

$$\min\left(\sum_{i=1}^N d_{ix(i)}\right), \quad (3.3)$$

subject to:

$$0 \leq \sum_{i=1}^N (r_{ix(i)} - R) \leq B_{max}, \quad \forall i = 1, \dots, N, \quad (3.4)$$

where 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 Chapter 2.

²Note that the “no-underflow” constraint can always be met by adding filler bits

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

Fact 3.1 *For a given source and CBR channel rate, one can decrease the distortion by increasing the transmission delay in the system. The best performance that can be achieved with channel rate R is obtained when $L = N$, i.e. there is only a constraint on the total bit rate budget.*

3.2.2 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 [91].

A LB can be described as follows [90]. Each packet generated by the source has to receive a token in order to be transmitted. For simplicity, assume that packets have size R bits and that tokens are generated at a rate of 1 token per frame interval. Furthermore, assume that we have a “bucket size” of N_b , i.e. the encoder can store at most N_b tokens. A LB constraint can thus be represented by the two parameters $LB(N_b, R)$. The leaky bucket policing mechanism requires the source to give up a token for every R bits it transmits. Thus, using our previous notation, in order for the R_i bits corresponding to the i -th frame to be transmitted, enough tokens have to be available. The constraint can be written as:

$$0 \leq \sum_{k=1}^i R - R_i \leq R \cdot N_b. \quad (3.5)$$

We can readily see that we arrive at the same set of constraints from (3.2) when

we have $N_b = L$. If the initial states are also the same (i.e. 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 Chapter 2 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 Fig. 3-1 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 [91]):

Fact 3.2 *The 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 transmission delay.*

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 [36]. Furthermore, we are assuming that the network is able to transmit all the source rate without increasing the delay, i.e. without requiring internal buffering.

Note that the advantage of using VBR is significant, since to reach the same quality levels in a CBR environment would require end-to-end delays that might be unacceptable for some applications. However, the key question in the VBR environment is how the network is going to provide the transmission resources. 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.

3.2.3 VBR vs. CBR: Network aspects

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.

3.2.3.1 Resource allocation within the network

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,

$$\mathcal{B}(\mathcal{R}, r) = \{B(R_i, r)\}_{i=1}^N \quad \text{where} \quad B(R_i, r) = \max(B(R_{i-1}) + R_i - r, 0), \quad \forall i, \quad (3.6)$$

and

$$\mathcal{UB}(\mathcal{R}, r) = \{UB(R_i, r)\}_{i=1}^N \quad \text{where} \quad UB(R_i, r) = UB(R_{i-1}) + R_i - r, \quad \forall i. \quad (3.7)$$

$\mathcal{B}(\mathcal{R}, r)$ 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:

$$\mathcal{U}(\mathcal{R}, r) = \{B(R_i, r) - UB(R_i, r)\}_{i=1}^N \quad (3.8)$$

We also define:

$$B_{max}(\mathcal{R}, r) = \max_i(B(R_i, r)), \quad (3.9)$$

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 3.2.2, since the 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

$$B_{max}(\mathcal{R}, R_b) \leq N_b \cdot R_b. \quad (3.10)$$

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 $B_{max}(\mathcal{R}, R_b) < N_b \cdot R_b$ 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 $N_b \cdot R_b$ (i.e. 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 3.3 *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. 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 $N_b \cdot R_b$ 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, indeed that is the main advantage of statistical multiplexing. However when several sources are considered simultaneously and are sharing a link of known capacity we can use this fact. 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 the constraint that the *combined* source, $\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2$, has to comply with is that defined by $LB(k, r)$. 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, i.e. \mathcal{R}_1 is admissible with $LB(k, r_1)$ and \mathcal{R}_2 is admissible with $LB(k, r_2)$. Then there is some statistical multiplexing gain (SMG) if $r < r_1 + r_2$. We can now state:

Fact 3.4 *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, i.e. 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.

The next two sections will be devoted, respectively, to showing that greedy coding (i.e. 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, the effect of greedy coding strategies.

3.3 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.

3.3.1 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 for $\mathcal{U}(\mathcal{R}, r)$ large will favor SMG.

Figure 3-2: Rate and distortion behavior for a sequence containing four types of scenes: Test, Normal, Easy, Difficult. Note that the scales are not important: we just try to qualitatively illustrate the typical behavior.

3.3.1.1 Types of scenes

We can classify video scenes into four important types, as sketched in Fig. 3-2. 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.

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, i.e. 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, $E(\text{SNR})$, 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 $E(\text{SNR})$ 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 type are 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. The distinction between coding for a target SNR rather than an absolutely maximized average SNR does come into play for the scenes where frames of different complexity are mixed, and the coder should avoid increasing the SNR for the easier frames beyond the target threshold.

3.3.1.2 Coding criteria: constant-Q, target-R and target-SNR

To illustrate and further understand the idea of coding for a *target* SNR, we coded a five-minute sequence (7200 frames) from the movie “Star Wars” using monochrome JPEG coding [1] 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, $\text{SNR}(t)$ we obtain valuable information about the four scene types described above, including relative frequency, time correlation structure etc.

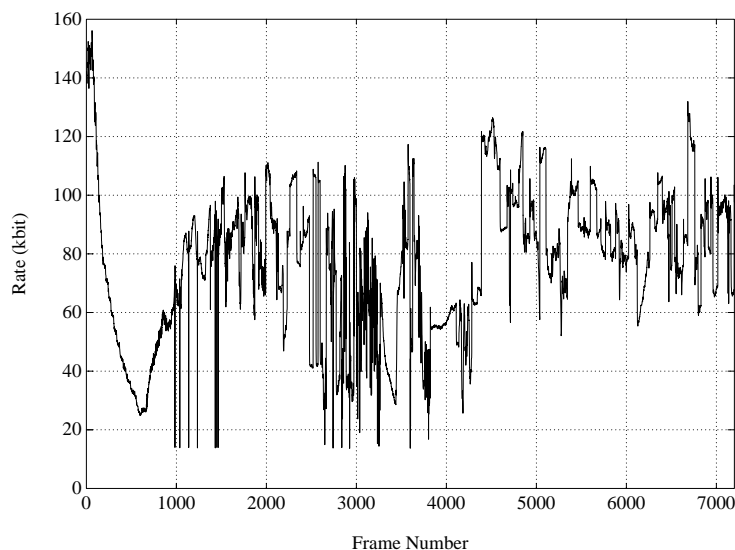


Figure 3-3: Rate time series with constant Quantizer of 0.4. Peak/mean rate = $156118.0/76482.6 = 2.04$.

Figures 3-3 - 3-8 show $R(t)$ and $\text{SNR}(t)$ for three rules governing the choice of quantizer for each frame. The first system (*constant-Q*, Figs. 3-3 and 3-4) 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*, Figs. 3-5 and 3-6) 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

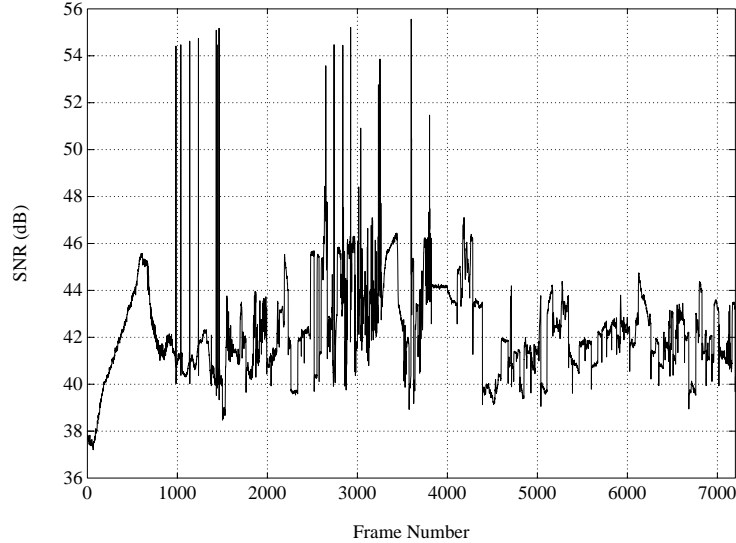


Figure 3-4: SNR time series with constant Quantizer of 0.4. Peak/mean dist = $16345.8/5545.4 = 2.95$.

$R \leq R_{Target}$. The final case (*target-SNR*, Figs. 3-7 and 3-8) 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 $SNR \geq SNR_{Target}$.

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 (i.e. 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, and its rate is somewhat more bursty than the constant-Q system, the peak is the same and the

mean is lower.

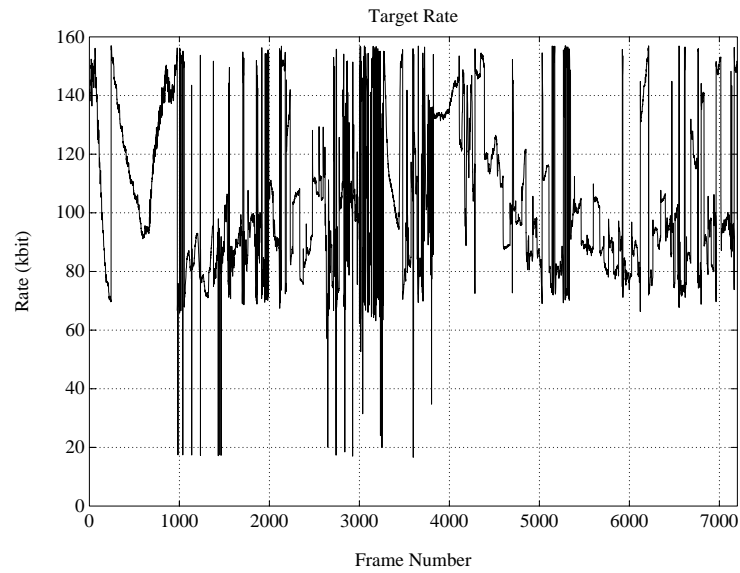


Figure 3-5: Rate time series with target Rate of 157000 b/frame. Peak/mean rate = $156994.0/105110.8 = 1.49$.

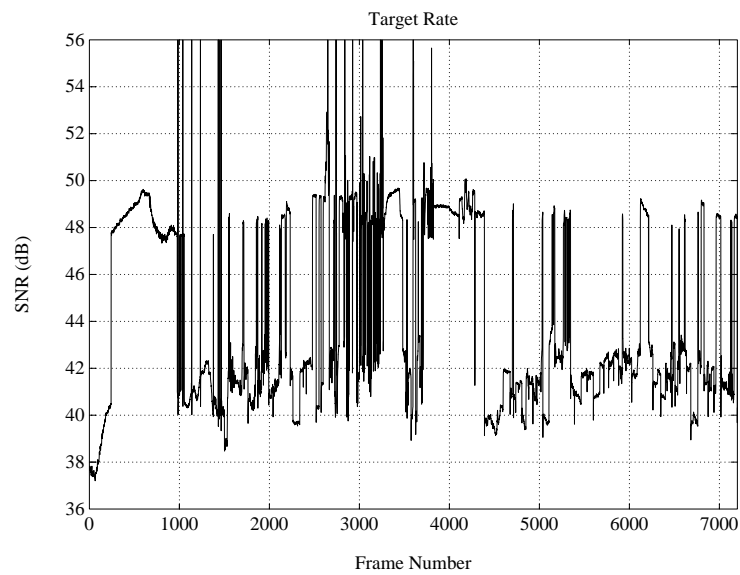


Figure 3-6: SNR time series with target Rate of 157000 b/frame. Peak/mean dist = $16345.8/4614.1 = 3.54$.

Since the target-R system peak to mean ratio is 1.5, we conclude that 33% of

Strategy	Peak rate	Mean rate	Peak/mean rate	Peak/mean distortion
constant-Q	156kbit	76.4kbit	2.04	2.95
target-R	156kbit	105kbit	1.49	3.54
target-SNR	156kbit	46kbit	3.39	1.27

Table 3.1: 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 156kbit.

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 anyway.) 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 (e.g. 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 or many sources share a pipe. See Table 3.3.1.2 for a summary of the results.

In this movie, there are three or four scenes more difficult than our test scene [36] (i.e. 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. A large smoothing buffer (beyond one frame) will not improve the examples shown because the peak allocated rate is always sufficient (this is true for CBR coding as well). Where it is not, however, we can use the buffer to re-allocate rate across the several buffered frames. If an *easy* frame occurs within a *difficult* scene, we should still not optimize it below the target quality threshold. This will yield more bits to use on the critical frames.

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 (e.g. MPEG), the smoothing buffer averages bandwidth across frames coded with two or three different algorithms. To make a fair evaluation of rate and quality for a non-greedy algorithm, we would have to explicitly take into account the rate allocation algorithm which attempts to optimally trade off resources among the buffered frames (see next Section). (Even in our example, we ignore the possibility of changing quantization level within the frame.) 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.

3.3.2 Coder-Network Interface

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.

3.3.2.1 Greedy coding and network policing

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 (e.g. 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 3.2.2, 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 (i.e. with relatively scarce resources), we choose $R = 60\text{ kbit/frame}$ and target SNR = 41 dB . The examples of figures 3-9 and 3-10 compare non-greedy and greedy buffer control strategies. The greedy buffer control is the optimal bit allocation of Chapter 2 which maximizes the average SNR for the given rate (here given by the LB leak rate), and the constraint that the buffer (given by the LB bucket size) does not overflow. As was seen in Section 3.2.2 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). This would be a simple implementation of the thresholded MSE idea described in Section 1.4.1 where we assume $MSE = 0$ for quantizers such that $SNR > 41\text{ dB}$. The basic idea of non-greedy coding has been proposed before in rate control for ATM transmission of video [57, 91]. The novelty here is that we express the constraint in terms of distortion, rather than as a lower bound in the quantizer stepsize as in [57, 91] so that 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. 3-9(a)) while the buffer is almost never in underflow (see Fig. 3-10(a)). By contrast, the non-greedy version produces much more consistent quality (see Fig. 3-9(b)), while using less network resources as shown by the frequent occurrence of buffer underflow (see Fig. 3-10(b)). For the full 5 minute segment (with buffer size

of 120kbit), the mean buffer output rate is 59kbit/frame for the greedy optimization, and only 46kbit/frame for the non-greedy case (i.e. a 33% reduction in average rate). The SMG is necessarily less in this case than in the previous target-SNR example since we have chosen an operating point with higher target SNR and a lower rate constraint. This results in more *difficult* and less *easy* scenes. Note (see Fig. 3-9) 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 difficult scenes*.

3.3.2.2 Network issues

To encourage users to adopt a non-greedy coding algorithm requires 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 can 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; i.e. as more sources are combined, the bandwidth allocated for each source R_a , decreases with N . The function $R_a(N)$ depends on the nature of the source [36]. So to define the network policing algorithm we must have an understanding of both the coder and network traffic control tasks.

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. The difference between allocated and mean bandwidth may be recovered statistically by the network.

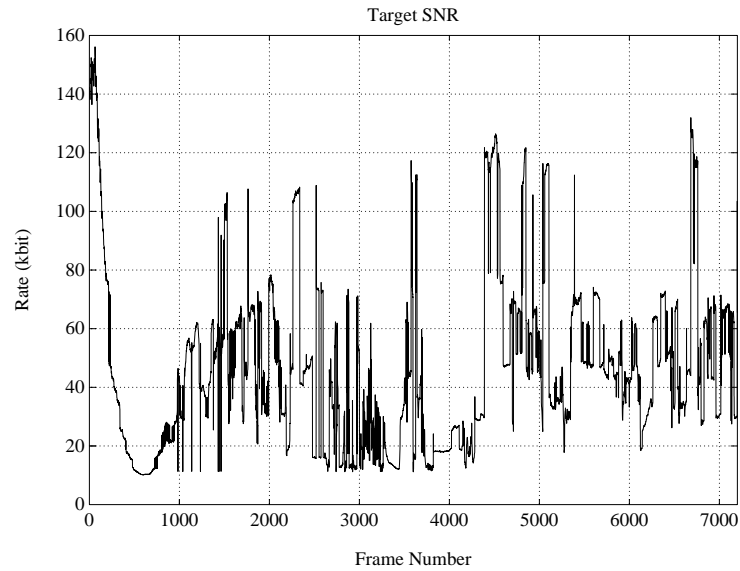


Figure 3-7: Rate time series with target SNR of 36.8 dB. Peak/mean rate = $156118.0/46070.8 = 3.39$.

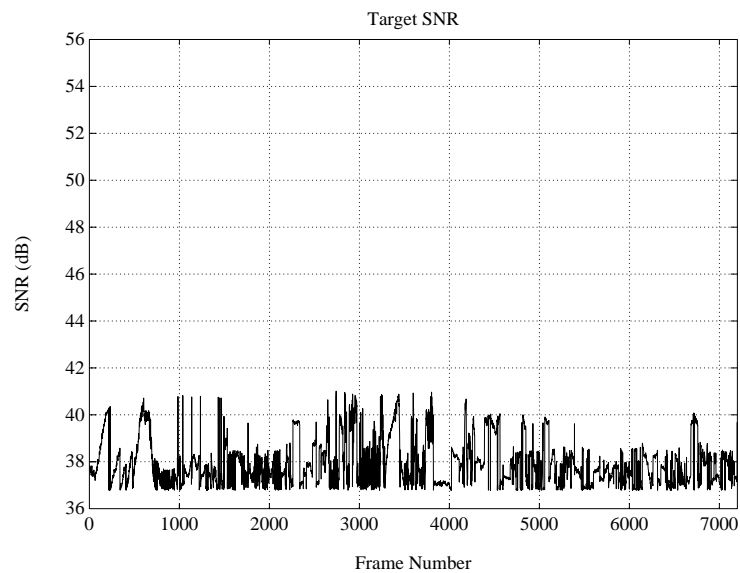


Figure 3-8: SNR time series with target Distortion of 18000. Peak/mean dist = $17998.4/14120.2 = 1.27$.

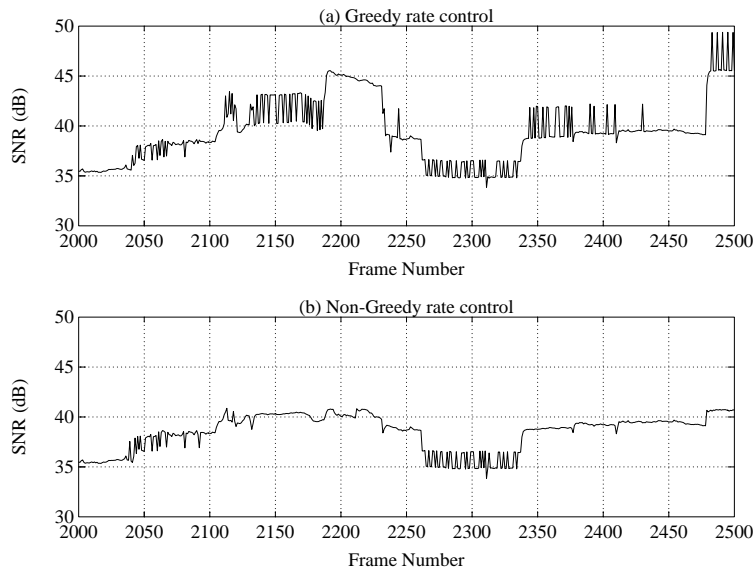


Figure 3-9: SNR trace with (a) greedy and (b) non-greedy rate control. Note that the SNR remains the same for the most difficult scene, but does not exceed the target for easier scenes in the non-greedy case.

3.4 Single and double leaky buckets

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 assume sources behaving in a non-greedy fashion. Since a simple LB may be too loose a constraint we examine other alternatives that will provide a “richer” set of constraints. While other alternatives to LB policing have been proposed [45, 102], we concentrate here on a solution using multiple leaky buckets due to the simplicity of their implementations

We study the trade-off involved in choosing the leaky bucket constraint by looking, for any given LB constraint, at the maximum average rate that can be generated while still abiding by the constraints. We will call these the worst case bursts and they will be measured as the maximum average that can be used over a window of i frames

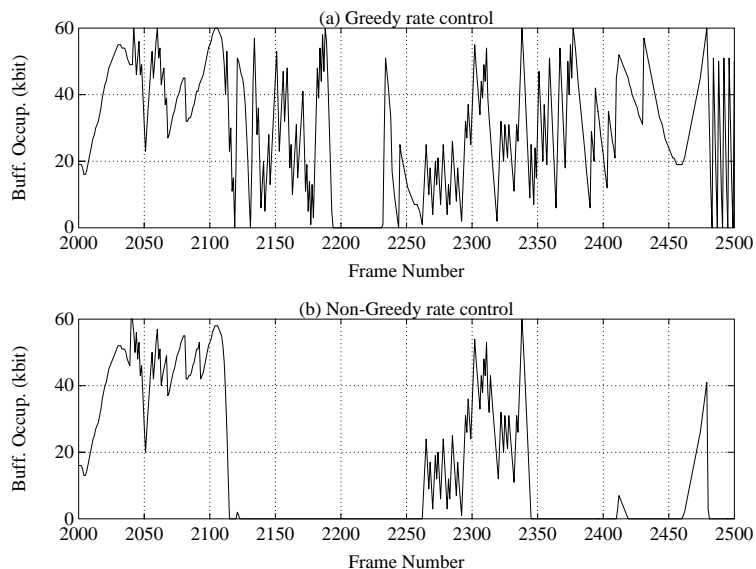


Figure 3-10: Buffer occupancy trace with (a) greedy and (b) non-greedy rate control. Note substantial buffer underflow for easy scenes in non-greedy case.

when a $LB(N_b, R)$ is used.

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. As an example, for a $LB(N_b, R)$, a sequence where every frame uses R bits is admissible. Similarly, a sequence that uses $N_b \cdot R$ bits for every N_b -th frame and zero bits for those in between is also admissible. Obviously, these are extreme cases but indicate that sources with varying degrees of “peakiness” 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 $N_b \cdot R$ bits, where N_b and R are the window size and leak rate respectively. Then we have that:

Figure 3-11: Worst case average rate for several window sizes. As an example: the point $(1, R \cdot N)$ indicates that the maximum bit rate that can be used by a single frame is $R \cdot N$. The range of average rates allowed under the constraint is represented the area under the curve.

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 and we want to adjust the LB parameters to make the sequence admissible, i.e. choose N_b , 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_b is chosen then the required bit rate 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 $N_b \cdot R$ bits.
- Conversely, if a small N_b is chosen then the required bit rate R will be higher (in the limit case of $N_b = 1$, as high as the highest frame rate in the sequence) whereas the product $N_b \cdot R$ would be relatively small.

This trade-off has been noted in the literature on policing functions [90] although here we look at it in a deterministic, rather than stochastic, fashion. In [90] the fact that the maximum peak rate increases as the window size N increases was seen as justification to police only the, loosely defined, peak rate, i.e. 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 $N_1 \cdot R_1 > N_2 \cdot R_2$ 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 $N_2 \cdot R_2$.

By choosing two different window sizes as N_1, N_2 we ensure that the two problems mentioned above are not encountered, i.e., referring to Fig. 3-12 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 $R_2 \cdot N_2 < R_1 \cdot N_1$ 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, i.e. when the source uses as many bits as it is allowed to, the bit rate that is used by the source is smaller (either the peak or the long term average) than in the single LB case (see Figs. 3-11 and 3-12). A double LB scheme allows the same peak rate (resp. average rate) but with a smaller long term average (resp. 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 Fig. 3-10 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 Fig 3-13. Here the maximum allowable peaks are small (180kbit/frame) whereas the long term average is 60kbit/frame. Thus the danger is that a greedy source could use continuously 60kbit/frame. Indeed the greedy source of Fig. 3-10(a) would be admissible under these constraints. Conversely one could choose a longer window LB, LB(60,55), see Fig 3-14 where the long term average would be kept lower (55kbit/frame). However the danger here is that a source could be admissible while generating a peak rate of up to 3300kbit for one frame.

When the two LB are combined, see Fig 3-15, 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

Figure 3-12: 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.

LB scheme, the greedy sequence of Fig. 3-10(a) would *not* be admissible, while the non-greedy sequence of Fig. 3-10(b) would be.

3.5 Conclusions

We have examined the problem of rate control for video encoders designed for transmission over packet networks. The main point is to note 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 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. Recent work has also suggested increasing the constraints on

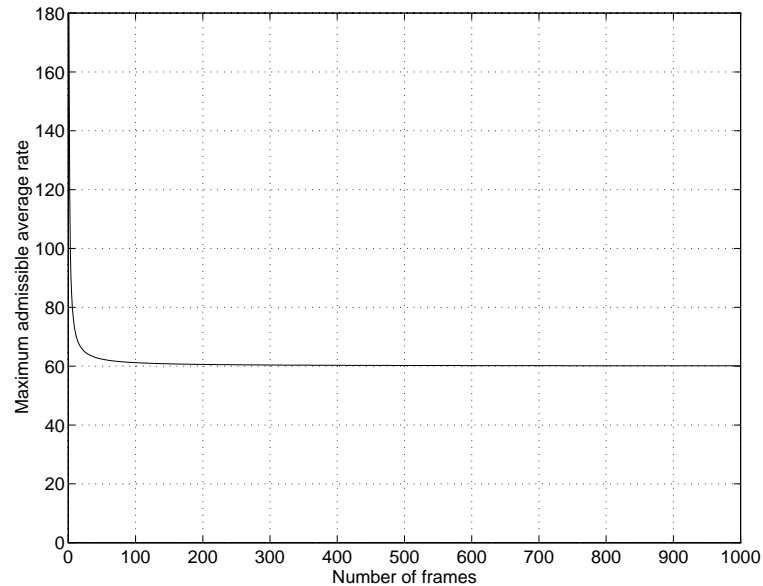


Figure 3-13: Worst case burst curve for a $LB(3,60)$ that has been chosen for the non-greedy source of Fig. 3-10(b). The window is short ($N = 3$) and thus the leak rate has to be large enough to permit the larger frames to be sent. The drawback is that the long term average is 60kbit/frame, while the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

the video streams, without resorting to using leaky buckets, in order to achieve reliable network operation with curtailing the potential benefits of VBR video [45]. In relation to the analyses of Chapter 2 we have proposed that rate control algorithms should be subject to additional constraints (i.e. be non-greedy) in a packet video environment.

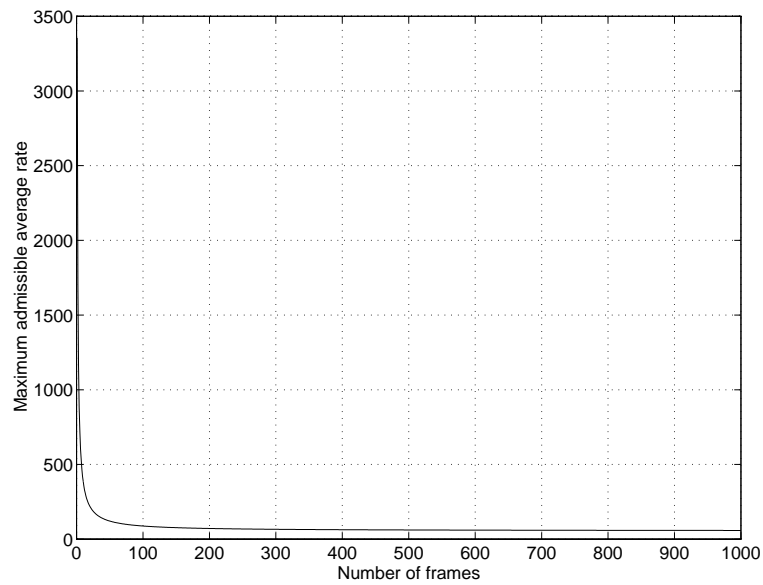


Figure 3-14: Worst case burst curve for a LB(60,55). The non-greedy sequence of Fig. 3-10(b) is also admissible under this LB. Note that the longer window $N_b = 60$ enforces a lower long term average. However, there is the danger that a compliant source may generate bursts of up to 3000 kbit/frame!

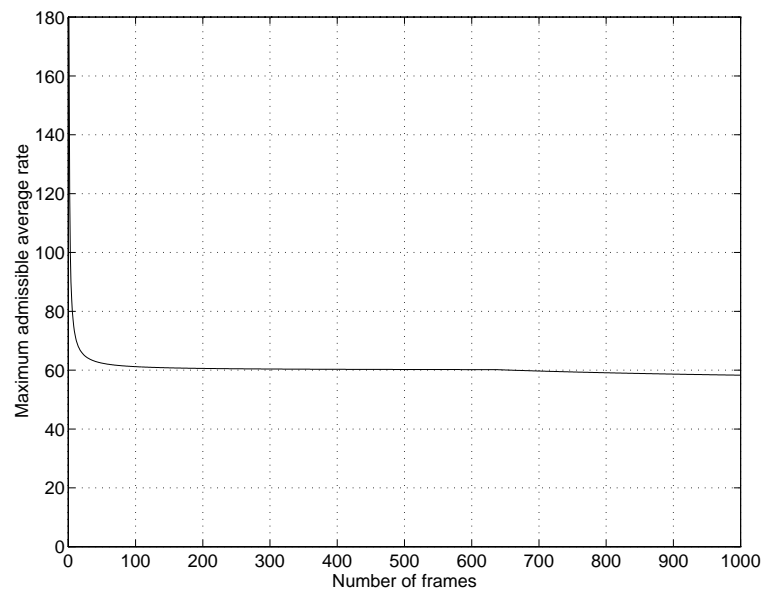


Figure 3-15: Effect of combining two LB's. The resulting worst case burst curve shows both the lower long term average (which tends to 55kbit/frame) and smaller short term bursts (less than 180 kbit/frame).