Constraints on Variable Bit-Rate Video for ATM Networks

Amy R. Reibman, *Member, IEEE* and Barry G. Haskell, *Fellow, IEEE*

Abstract—**We consider constraints on the encoded bit rate of a video signal that are imposed by a channel and encoder and decoder buffers. We present conditions that ensure that the video encoder and decoder buffers do not overflow or underflow when the channel can transmit a variable bit rate. Using these conditions and a commonly proposed network-user contract, we examine the effect of a network policing function on the allowable variability in the encoded video bit rate. We describe how these ideas might be implemented in a system that controls both the encoded and transmitted bit rates. Finally, we present the performance of video that has been encoded using the derived constraints for the leaky bucket channel.**

I. Introduction

TRADITIONALLY, video has been transmitted using channels that have constant rate. Because most video compression algorithms use variable length codes to improve compression, a buffer at the encoder is necessary to translate the variable rate output by the encoder into the constant-rate channel. A similar buffer is necessary at the decoder to translate the constant channel bit rate into a variable bit rate.

Recently, however, there has been much interest in sending video over broadband integrated services digital networks (B-ISDN). These networks are able to support variable bit rates by partitioning user data into a sequence of "cells" and inputting them to the network asynchronously. For this reason B-ISDN is referred to as an asynchronous transfer mode (ATM) network. ATM networks may allow video to be transmitted on a channel with variable rate.

In this paper, we examine the constraints imposed on the encoded video bit-rate as a result of encoder and decoder buffering. In particular, we show that for a constant-rate channel, it is possible to prevent the decoder buffer from overflowing or underflowing simply by ensuring that the encoder buffer never underflows or overflows, this is no longer the case for a variable-rate channel. Additional constraints must be imposed on the encoding rate, the channel rate, or both.

In addition, we also examine the effect of a proposed ATM network policing function on the encoded video bit rate. In general, network-imposed policing functions have the effect of limiting the bit rate that the network will guarantee to the user. Because video requires that certain information be received, it is therefore necessary for the video system to constrain its bit rate onto the network to ensure that it does not exceed that allowed by the policing function. Hence, we also examine the constraint on the encoded video bit rate imposed by these policing functions.

This paper is organized as follows. In Section II, we describe the ATM networks and the role of network policing functions. In Section III, we present constraints imposed by the encoder and decoder buffers on both the encoded and transmitted bit rates. In Section IV, we examine how channel constraints can further restrict the encoded and transmitted bit-rates. Sections V and VI present a system that jointly controls the encoded and transmitted bit-rates. Section VII presents some examples illustrating the improvements that can result by using a variable-rate channel. Section VIII concludes the paper with a discussion.

II. ATM Networks

ATM networks are often proposed for transmitting video because they can accommodate the bit rate necessary for high-quality video, and because the quality of the video can benefit from the variable bit rate that the ATM network can theoretically provide. As a result, recent research has gone into developing video compression algorithms that have unconstrained bit rates but achieve constant quality [1]. By having the user select a desired quality, these algorithms can provide better compressed video than algorithms designed for a constant-rate channel, even when both algorithms produce the same average rate.

However, if the bit rate of all streams were to vary arbitrarily, the network would be unable to provide guaranteed delivery of all packets. Two solutions to this have been proposed. The first solution is to have the user assign a priority (high or low) to each packet submitted to the network. The high-priority packets are guaranteed by the network; the low-priority packets can be dropped by the network. The second solution (which is still necessary even given the first) is to assume that a contract exists between the network and the user. The network guarantees that the cell loss rate (CLR) for high-priority packets will not exceed an agreed-to value, provided that the user does not submit too many. A policing function monitors the user output and either drops packets in excess of the

Manuscript received December 3, 1991; revised April 6, 1992. Paper was recommended by Associate Editor Yasuhiko Yasuda.

The authors are with AT&T Bell Laboratories, Holmdel, NJ 07733-3030.

IEEE Log Number 9202479.

1051-8215/92$03.00

contract or marks these excess packets as low priority, possibly to be dropped later in the network.

The advantages of priority labeling, both for video [2]–[6] and for the network [7] have been well established. In addition, the effect of a policing function on the network behavior has also been studied [8]–[10]. A discussion of a unified approach to controlling congestion is given in [11]. In this paper we concentrate on examining the effect of the policing functions on the video quality.

For video, the existence of a policing function has a significant effect on the output bit rate because some information is essential to the decoder, e.g., timing data, start-of-picture codes, etc. If this information is not received, the video decoder will be unable to decode anything. Therefore, it is essential to the video user that all high-priority packets are received. This implies that the network should *never* drop high-priority packets or, equivalently, that the network should never change the marking of a high-priority packet to low priority. Therefore, it is essential that the video algorithm control its output bit rate to ensure that the network-imposed policing function does not detect any excess high-priority packets.

III. Video Buffer Verification for General Variable-Rate Channels

In this section, we present conditions necessary to guarantee that the buffers at the video encoder and decoder do not overflow or underflow. These conditions are presented both in terms of a constraint on the encoder rate and a constraint on the channel rate. The channel rate may be variable but is not otherwise constrained.

Clearly, if either buffer overflows, information is lost. Encoder buffer underflow is only a problem if the channel has constant bit rate and cannot be turned off; in this case, something must always be transmitted. Because encoder buffer underflow can always be avoided by sending stuffing bits, it is not considered a problem.

However, the concept of decoder buffer underflow is less intuitive since the decoder is generally capable of removing bits from its buffer faster than bits arrive. The decoder buffer is said to underflow when the decoder must display a new frame (which happens, e.g., every one-thirtieth of a second), but no new frame has been decoded. Therefore, three things must happen simultaneously: 1) the decoder buffer is empty, 2) the next-frame frame memory is not full, and 3) it is time to display a freshly decoded frame. For this reason, we discretize the problem using the uncoded frame period. A smaller period could be used if necessary in systems having relatively small buffer sizes.

For a constant bit-rate channel, it is possible to determine upper bounds on encoder and decoder buffer sizes such that if the encoder's output rate is controlled to ensure no encoder buffer overflow or underflow, then the decoder buffer will also never underflow or overflow. As we will see, the problem becomes more difficult when the channel may transmit a variable bit rate, for example, when transmitting video across packet (ATM) networks.

A. Buffer Dynamics

Examples of the encoder and decoder buffer dynamics are shown in Figs. 2 and 3, respectively. We define $E(t)$ to be the number of bits (or bytes or packets) output by the encoder at time t. The channel bit rate $R(t)$ is variable. $B^e(t)$ and $B^d(t)$ are the instantaneous fullnesses of the encoder and decoder buffers, respectively. Each buffer has a maximum size, $B^e_{\max}$ and $B^d_{\max}$, that cannot be exceeded. Given $B^e_{\max}$, the encoder is designed to ensure its buffer never overflows, i.e.,

$$0 \le B^e(t) \le B^e_{\max} \ \forall t. \tag{1}$$

Here we examine conditions on the buffers and the channel to ensure the decoder buffer never overflows or underflows, i.e.,

$$0 \le B^d(t) \le B^d_{\max} \ \forall t. \tag{2}$$

First, we discretize the problem by defining E_i $(i = 1, 2, \cdots)$ to be the number of bits in the interval $[(i-1)T, iT)$, where T is the duration of one uncoded frame as output from the camera or fed to the display. Therefore,

$$E_i = \int_{(i-1)T}^{iT} E(t)\, dt. \tag{3}$$

Similarly, let R_i be the number of bits that are transmitted during the ith frame period:

$$R_i = \int_{(i-1)T}^{iT} R(t)\, dt. \tag{4}$$

The encoder buffer receives bits at rate $E(t)$ and outputs bits at rate $R(t)$. Therefore, assuming empty buffers prior to startup at time $t = 0$

$$B^e(t) = \int_0^t [E(s) - R(s)]\, ds \tag{5}$$

and the encoder buffer fullness after encoding frame i is

$$B^e_i = B^e(iT) = \int_0^{iT} [E(s) - R(s)]\, ds. \tag{6}$$

This can be written explicitly as

$$B^e_i = \sum_{j=1}^{i} E_j - \sum_{j=1}^{i} R_j \tag{7}$$

or recursively as

$$B^e_i = B^e_{i-1} + E_i - R_i. \tag{8}$$

After the decoder begins to receive data, it waits LT s before starting to decode. We assume for clarity that L is an integer, although this is not necessary. At the decoder, we define a new time index τ, which is zero when decoding starts.

$$t = \tau + LT + \text{channel.delay}. \tag{9}$$

The encoder can calculate the initial fullness of the decoder buffer $B^d(0)$ (when $\tau = 0$) if L is predetermined or

sent explicitly as a decoder parameter. It is given by

$$B_0^d = \sum_{j=1}^{L} R_j. \tag{10}$$

The decoder buffer fullness at time $\tau = iT$ is then given by

$$B_i^d = B_{i-1}^d + R_{L+i} - E_i \tag{11}$$

$$B_i^d = B_0^d + \sum_{j=1}^{i} R_{L+j} - \sum_{j=1}^{i} E_j. \tag{12}$$

For $(i-1)T < \tau < iT$, the decoder buffer fullness varies, depending on the channel rate $R(t)$ and the rate at which the decoder extracts data from its buffer. In general in this interval, the decoder buffer fullness could rise up to the larger of $B_{i-1}^d + E_{i-1}$ or $B_i^d + E_i$, or fall to the smaller of $B_{i-1}^d - E_i$ of $B_i^d - E_{i+1}$.

There are two useful expressions for B_i^d when the channel has variable rate, each derived using (12) and (10).

$$\begin{aligned} B_i^d &= \sum_{j=1}^{L} R_j + \sum_{j=L+1}^{L+i} R_j - \sum_{j=1}^{i} E_j \\ &= \sum_{j=i+1}^{i+L} R_j - \left(\sum_{j=1}^{i} E_j - \sum_{j=1}^{i} R_j \right) \\ &= \sum_{j=i+1}^{i+L} R_j - B_i^e \end{aligned} \tag{13}$$

or

$$\begin{aligned} B_i^d &= \sum_{j=i+1}^{i+L} E_j - \left(\sum_{j=1}^{i+L} E_j - \sum_{j=1}^{i+L} R_j \right) \\ &= \sum_{j=i+1}^{i+L} E_j - B_{i+L}^e. \end{aligned} \tag{14}$$

Equation (13) expresses B_i^d as a function of the cumulative channel rates over the last L frames and the encoder buffer fullness L frames ago, when frame i was encoded. Equation (14) expresses it as a function of the cumulative encoder rates over the last L frames and the encoder buffer fullness now, or when frame $i+L$ is encoded. This is an expression that the encoder can compute directly from its observations.

B. Buffer Verification

We now combine equations from Section III-A with (1) and (2), to obtain conditions necessary to prevent encoder and decoder buffer underflow and overflow using a general variable-rate channel. To prevent encoder buffer overflow and underflow, from (1) and (8) we have

$$0 \le B_{i-1}^e + E_i - R_i \le B_{\max}^e \tag{15}$$

$$R_i - B_{i-1}^e \le E_i \le R_i + B_{\max}^e - B_{i-1}^e \tag{16}$$

which is a constraint on the number of bits per coded frame for a given channel rate. For example, when the channel has a constant rate, the encoder prevents its buffer from overflowing or underflowing by varying the quality of coding [12]. If the encoder sees that its buffer is approaching fullness, it reduces the bit rate being input to the buffer by reducing the quality of coding, using a coarser quantizer on the data. Conversely, if encoder buffer underflow threatens, the encoder can generate more input data, either by increasing the quality of coding or by outputting stuffing data that are consistent with the coding syntax.

Alternatively, to achieve constant picture quality, we can instead let the number of bits per frame E_i be unconstrained, and force the channel rate R_i to accommodate. Rewriting (15), we get

$$0 \ge -B_{i-1}^e - E_i + R_i \ge -B_{\max}^e$$

$$E_i - (B_{\max}^e - B_{i-1}^e) \le R_i \le B_{i-1}^e + E_i \tag{17}$$

encoder overflow condition — encoder underflow condition

Therefore, encoder buffer overflow and underflow can be prevented by constraining either the encoded bit rate per frame period (16), or the transmitted bit rate per frame period (17).

To prevent decoder buffer overflow and underflow, we combine (2) and (11) to obtain

$$0 \le B_{i-1}^d + R_{i+L} - E_i \le B_{\max}^d \tag{18}$$

$$R_{i+L} + B_{i-1}^d - B_{\max}^d \le E_i \le R_{i+L} + B_{i-1}^d \tag{19}$$

which is a constraint on the encoder bit rate for a given channel rate.

Alternatively, we can again allow the number of bits per frame to be unconstrained and examine the constraint on the channel rate R_i.

$$E_i - B_{i-1}^d \le R_{i+L} \le E_i + (B_{\max}^d - B_{i-1}^d)$$

or, for $i > L$,

$$E_{i-L} - B_{i-L-1}^d \le R_i \le E_{i-L} + (B_{\max}^d - B_{i-L-1}^d).$$

decoder underflow condition — decoder overflow condition

(20)

This provides a restriction on the channel rate R_i that depends on the encoder activity L frames ago.

Even if the channel rate is completely controllable, a restriction still exists on E_i, the number of bits used to encode frame i. This constraint is necessary to prevent simultaneous overflow of both buffers. (Note that simultaneous underflow of both buffers is not a problem. The

upper bound of (17) is always greater than the lower bound of (20).)

It can be seen either by combining the lower bound of (17) with the upper bound of (20),

$$E_i - (B^e_{\max} - B^e_{i-1}) \le R_i \le E_{i-L} + (B^d_{\max} - B^d_{i-L-1})$$

$$E_i \le E_{i-L} + (B^e_{\max} - B^e_{i-1}) + (B^d_{\max} - B^d_{i-L-1}) \quad (21)$$

or by noting that because the delay is L, the system must store L frames worth of data,

$$0 \le \sum_{j=i-L+1}^{i} E_j \le B^d_{\max} + B^e_{\max}. \quad (22)$$

These bounds arise because of the finite memory of the video system. The system can store no more than $B^d_{\max} + B^e_{\max}$ bits at any given time, but it must store L frames of data always. Therefore, these L frames cannot be coded with too many bits. In the case of equality for either (21) or (22), both buffers are completely full at the end of frame i.

In this section, we have considered the case where the channel delay is constant. To accommodate the variable channel delay expected in an ATM network, the largest expected channel delay should be used in (9). In addition, the decoder buffer should be large enough to contain the additional bits that may arrive with shorter delay. Thus if the minimum channel delay is Δ and the maximum channel delay is $\Delta + \delta$, (9) becomes

$$t = \tau + LT + \Delta + \delta \quad (23)$$

and the decoder buffer constraint of (18) becomes

$$0 \le B^d_{i-1} + R_{i+L} - E_i \le B^d_{\max} - \max_i \int_{iT}^{iT+\delta} R(s)\, ds. \quad (24)$$

IV. Buffer Verification for Channels with Rate Constraints

A. Fixed-Rate Channel

If the channel has a fixed bit rate, then the buffer verification problem simplifies. In particular, it is possible to guarantee that the decoder buffer never overflows or underflows, provided that the encoder buffer never overflows or underflows.

For the constant-rate channel let $R_i = RT$ be the number of bits transmitted during one uncoded frame period of duration T. The initial fullness of the decoder buffer when decoding starts is

$$B^d(0) = B^d_0 = LRT. \quad (25)$$

The key to simplification is in (12), which reduces to

$$B^d_j = B^d_0 - B^e_j \quad (26)$$

when the channel has constant rate. Note that this equation is not true for a variable-rate channel since, in that case,

$$B^d_j = B^d_0 + \sum_{i=1}^{j} R_{L+i} - \sum_{i=1}^{j} E_i$$

$$\ne B^d_0 + \sum_{i=1}^{j} R_i - \sum_{i=1}^{j} E_i$$

$$= B^d_0 - B^e_j. \quad (27)$$

Because B^e_j is always positive, the decoder buffer is never as full at the end of a frame as it was before decoding started. Therefore, to prevent decoder buffer overflow, using (26), the decoder buffer size can be chosen solely to ensure that it can handle the initial buffer fullness, plus the number of bits for one frame. In most cases, the decoder is much faster than the channel rate, so we can choose $B^d_{\max} = LRT + \delta$ where δ is small.

In addition, we know that the decoder buffer will never underflow, provided that

$$0 \le B^d_j = LRT - B^e_j, \quad (28)$$

or, provided that $B^e_j \le LRT$. Therefore, if we choose $B^e_{\max} = LRT$, and ensure that the encoder buffer never overflows, the decoder buffer will never underflow. Herein lies the simplicity of the constant-rate channel: it is possible to ensure that the decoder buffer does not overflow or underflow simply by ensuring that the encoder buffer does not overflow or underflow.

We now discuss the choice of the decoder delay L and indicate how the delay enables a variable encoder bit rate, even though the channel has a fixed rate. The encoder buffer fullness can be written as

$$B^e_j = \sum_{k=1}^{j} E_k - jRT \le LRT. \quad (29)$$

Reorganizing,

$$\sum_{k=1}^{j} E_k \le (L + j)RT$$

so that

$$L \ge \sum_{k=1}^{j} E_k/RT - j, \ \forall j \quad (30)$$

which indicates the trade-off between the necessary decoder delay and the variability in the number of encoded bits per frame. Because a variable number of bits per frame can provide better image quality, (30) also indicates the trade-off between the allowable decoder delay and the image quality.

Insight into how (30) involves the variability in the number of bits per coded frame can be seen by examining the two extremes of variability. First, suppose that all frames have the same number of bits $E_i = RT$. Then, $L \ge 0$, and no decoder delay is necessary. At the other

extreme, suppose all the transmitted bits were for frame 1; then $L \geq E_1/RT - 1$. In this case, the decoder must wait until (most of) the data for the first frame have been received.

Therefore, the constant-rate channel provides the simplicity of ensuring no decoder buffer overflow or underflow by monitoring encoder buffer underflow or overflow. In addition, even though the channel has constant rate, with the use of a delay, it is possible to obtain some variability in the number of bits per encoded frame.

B. Leaky-Bucket Channel

We show that for the channel whose rate is controlled by a leaky-bucket policing function, the conditions on the encoder bit rate are somewhat weaker than those for a fixed-rate channel. Therefore, we can get some additional flexibility on the encoder bit rate.

When the network implements a leaky-bucket policing function, it keeps a counter indicating the fullness of an imaginary buffer inside the network. The input to the imaginary buffer (henceforth called the "bucket" here) is R_i bits for frame period i. The output rate of the bucket is $\bar{R}$ bits per frame period. The bucket size is $N_{\max}$. Hence, the instantaneous bucket fullness is

$$N_i = \max\{0, N_{i-1} + R_i - \bar{R}\}. \tag{31}$$

If the bucket never underflows, N_i can be written as

$$N_i = \sum_{j=1}^{i} R_j - i\bar{R}. \tag{32}$$

However, (32) actually provides only a lower bound on the bucket fullness since the actual bucket fullness may be larger if bucket underflow has occurred.

To ensure that the policing mechanism does not mark high-priority packets as droppable, rate R_i must be such that the bucket never overflows, i.e.,

$$N_i \leq N_{\max} \quad \forall i$$

or

$$R_i \leq N_{\max} - N_{i-1} + \bar{R} \tag{33}$$

Equation (33) defines the leaky-bucket constraint on the rate that is input to the network. Even if the bucket does underflow, the rate can also be upper bounded by

$$R_i \leq N_{\max} - \sum_{j=1}^{i-1} R_j + i\bar{R}. \tag{34}$$

Combining (34) with (17), which constrains the rate to prevent encoder buffer underflow and overflow, we have a necessary condition on the encoded rate:

$$\begin{aligned} E_i &\leq N_{\max} + B^e_{\max} - \sum_{j=1}^{i-1} R_j + i\bar{R} - B^e_{i-1} \\ &\leq N_{\max} + B^e_{\max} + i\bar{R} - \sum_{j=1}^{i-1} R_j - \left(\sum_{j=1}^{i-1} E_j - \sum_{j=1}^{i-1} R_j\right) \\ &\leq N_{\max} + B^e_{\max} + i\bar{R} - \sum_{j=1}^{i-1} E_j \\ E_i &\leq B^E_{\max} + \bar{R} - B^E_{i-1} \end{aligned} \tag{35}$$

where $B^E_{\max} = N_{\max} + B^e_{\max}$ is the size of a virtual encoder buffer and

$$B^E_i = \sum_{j=1}^{i} E_j - i\bar{R}$$

is the fullness of the virtual encoder buffer at time i.

Therefore, the encoder output bit rate E_i must be constrained (by the compression algorithm) to ensure that a virtual encoder buffer of size $B^E_{\max}$ does not overflow, assuming a constant output rate of $\bar{R}$ bits per frame. Because this constraint is less strict than preventing an actual encoder buffer with the same drain rate but smaller size $B^e_{\max}$ from overflowing or underflowing, we have a potential advantage over a channel with constant rate.

However, this is not the only constraint. In fact, preventing decoder buffer overflow can impose a stronger constraint. In particular, the right side of the decoder rate constraint (20) may actually be more strict than the leaky-bucket rate constraint (33). As a result, we may not actually be able to obtain the full flexibility in the encoder bit rate equivalent to using a virtual encoder buffer of a larger size.

It is possible, however, to reduce the actual delay at the decoder without sacrificing the flexibility in the encoded bit rate. Theoretically, we can obtain the same flexibility in the encoded bit rate that is available with a constant-rate channel and decoder delay L when we use a leaky-bucket channel with zero delay, provided that $L = N_{\max}/\bar{R}$ and $N_{\max} = B^e_{\max} = B^d_{\max}$. However, recall that we will certainly be paying for both $N_{\max}$ and $\bar{R}$.

V. Coding System with Joint Channel and Encoder Rate Control

In this section, we describe a method whereby the number of encoded bits for each video frame and the number of bits transmitted across the variable rate channel are selected jointly. The necessity arises as described previously: with a variable bit-rate channel, the decoder buffer imposes a constraint on the transmitted bit rate that is different than that imposed by the encoder buffer. This method also provides the flexibility of having channel bit rates that are less than the maximum allowed by the channel, which may be desirable when the channel is not constrained solely by its peak rate.

A. System Description

Fig. 1 describes a system incorporating these concepts. In Fig. 1, a video signal is applied to the video encoder. The video encoder produces an encoded video bit stream

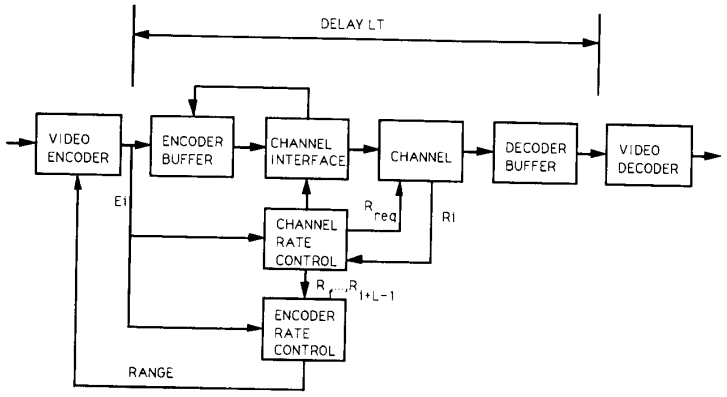

Fig. 1. System.

that is stored in the encoder buffer before being transmitted via the channel interface to the variable-rate channel. After being transmitted across the variable-rate channel, the video bit stream is stored in the decoder buffer. The bit stream from the decoder buffer is input to the video decoder, which outputs a video signal. The delay from encoder buffer input to decoder buffer output, exclusive of channel delay, is exactly LT s. The value of the delay L is known a priori, as are the encoder and decoder buffer sizes $B^e_{\max}$ and $B^d_{\max}$.

The video encoder can encode the video signal using any method that allows the number of bits that are produced to be controlled (see, e.g., [12]). A range indicating the number of bits that can be produced is provided by the encoder rate control device. The video encoder produces a bit stream that contains E_i number of bits in one frame period, which is within the range given by the encoder rate control device. These bits are input to the encoder buffer and stored until they are transmitted.

The channel rate control device takes as input the actual number of bits output in each frame period by the video encoder. It computes estimated channel rates R_i, $\cdots, R_{i+L-1}$, describing the number of bits that will be transmitted across the channel in the following L frame periods. These rates are chosen to prevent encoder and decoder buffer overflow and underflow and to conform to the channel constraint. The channel rate control device sends the estimated value of R_i to channel as R_{req}. We assume here that the channel grants the request, in which case $R_i = R_{\text{req}}$. (If the request is not granted, the channel rate control device can selectively discard information from the bit stream. However, such information discarding is an emergency measure only since our express purpose is to avoid such discarding.) If the encoder buffer empties, the channel interface unit immediately terminates transmission. In most cases, this will cause a reduction of R_i.

The encoder rate control device computes a bound on the number of bits that the video encoder may produce without overflowing or underflowing either the encoder or decoder buffers. It takes as input the actual number of bits E_i output in each frame period by the encoder. It also takes as input the channel rate values that are selected by the channel rate control device. The bound output by the encoder rate control device is computed to guarantee that neither the encoder nor decoder buffers overflow or underflow.

B. Encoder and Channel Rate-Control Devices

We describe here the joint operation of the encoder and channel rate-control devices. To simplify the discussion, we assume that the channel allows transmission at the requested rate. This is not an unreasonable assumption because the channel rate-control device is selecting estimated channel rates to conform to the channel constraints negotiated between the channel and the video system.

Joint operation of the encoder and channel rate-control devices:

1) Initialize buffer fullness variables prior to encoding frame $i = 1$: $B^e_i = B^d_i = 0$. Initialize leaky bucket fullness N_i.
2) Estimate the future channel rates, future leaky-bucket fullnesses, and future decoder-buffer fullnesses for the next L frames. For the channel rates, we utilize inequalities (20) and (33), where for $k \leq 0$, $E_k = 0$. Leaky-bucket and decoder-buffer fullnesses are given, respectively, by (31) and (12). Rewriting them, we get for $j = i, i+1, \cdots, i+L-1$

$$\underbrace{E_{j-L} - B^d_{j-L-1} \leq R_j}_{\text{decoder underflow condition}} \leq \underbrace{E_{j-L} + \left(B^d_{\max} - B^d_{j-L-1}\right)}_{\text{decoder overflow condition}} \tag{36}$$

$$R_j \leq N_{\max} - N_{j-1} + \bar{R} \tag{37}$$

$$N_j = \max\left\{0, N_{j-1} + R_j - \bar{R}\right\} \tag{38}$$

$$B^d_{j-L} = B^d_{j-L-1} + R_j - E_{j-L}. \tag{39}$$

Several selection methods for the estimated rates are discussed in Section VI. These methods may

ideally consider the fact that a frame with a large number of bits has just occurred or is imminent. They may also consider the cost of transmitting at a given rate. When $i \le L$, no frames are being decoded and the decoder buffer is only filling. In general, the sum of $R_1, \cdots, R_L$ should be chosen to exceed the expected encoded bit rate of the first few frames in order to avoid decoder buffer underflow.

3) Calculate an upper bound on R_{i+L} using the leaky-bucket constraint (37):

$$R_{i+L} \le R_{i+L}^{ub} = N_{\max} - N_{i+L-1} + \bar{R}. \quad (40)$$

4) Calculate an upper bound on E_i using constraints on encoder buffer overflow from inequality (16) and decoder buffer underflow from inequality (19).

$$E_i \le B_{\max}^e + R_i - B_{i-1}^e \quad (41)$$

$$E_i \le R_{i+L}^{ub} + B_{i-1}^d. \quad (42)$$

The minimum of these two upper bounds on E_i is output by the encoder rate control device to the video encoder.

5) Encode frame i to get E_i bits.
6) Using the actual value of E_i, recompute R_i, the actual number of bits transmitted this frame period. (This may be necessary if the encoder buffer would underflow, thus making the actual R_i less than that estimated.)

$$R_i = B_{i-1}^e + E_i. \quad (43)$$

7) Use actual values of E_i and R_i to compute actual values of B_i^e, N_i, and B_{i-L}^d using (8), (38), and (39), respectively.
8) Increment i, and go to step 2).

VI. Rate-Control Strategies

In this section, we describe an encoder rate control algorithm, and two channel rate-control algorithms for the leaky bucket. Other channel constraints could have been used instead. In the encoder rate-control algorithm, the quantizer step size used by the encoder is chosen to ensure not only that the encoder buffer does not overflow but also that the decoder buffer does not underflow when the corresponding data is decoded. In the channel rate-control algorithms, we select the channel rate R_i based on the channel constraints as well as the decoder buffer fullness.

Voeten *et al.* [13] present an encoder rate-control algorithm that considers the gabarit channel constraint and the encoder buffer. Bits are submitted to the channel as fast as the channel constraint allows. The decoder buffer is not considered.

A. Encoder Rate Control

To control the encoder rate in order to ensure no encoder or decoder buffer overflow or underflow, we select the quantizer step size to be used by the encoder. Our selection of quantizer step size is modified from the Reference Model 8 (RM8) simulation encoder [12].

In the RM8 simulation encoder, the quantizer step size is selected based solely on the fullness of the encoder buffer. With the encoder buffer size chosen to be $B_{\max}^e = \mathrm{P} * 6400$, the RM8 buffer control selects

$$Q = 2 * \mathrm{INT}(32 * B^e(t)/B_{\max}^e) + 2 \quad (44)$$

where INT( ) denotes truncation to a fraction without rounding.

We make two modifications to the RM8 encoder rate control algorithm. The first modification is introduced to prevent the decoder buffer from underflowing when the frame currently being encoded is finally decoded. We rewrite the constraint of (42) as

$$E_i \le R_{i+L}^{ub} + \left[B_{\max}^d - (B_{\max}^d - B_{i-1}^d) \right] \quad (45)$$

and, comparing this to the encoder buffer overflow constraint (41), we set

$$Q = 2 * \mathrm{INT}(32 * \max\left\{ B^e(t)/B_{\max}^e, \left[B_{\max}^d - B^d(t + LT) \right)/B_{\max}^d \right]\} + 2. \quad (46)$$

Note that the value of the decoder buffer fullness is a prediction of what we expect the decoder buffer fullness to be when the current frame is decoded.

If the channel rate is constant, (46) does not select a different quantizer than RM8. However, if the channel rate is variable, the quantization control in (46) is necessary to prevent the current frame from being encoded with more bits than the system can transmit before this frame is to be decoded.

However, an additional modification must be made to the quantization strategy to enable the leaky bucket to empty when scene activity is low. If we start with a full leaky bucket and choose Q as in (46), the leaky bucket would never empty and we would always transmit at the average channel rate. As with RM8, the quantizer step size can decrease arbitrarily to increase the number of encoded bits per frame and keep the encoder buffer from underflowing. However, if we can enable the leaky bucket to empty, the channel rate can subsequently be larger than average, and the leaky-bucket channel can allow better performance than a peak-rate channel. Thus, a second modification to the RM8 quantizer step size is necessary to obtain some advantages from a variable bit-rate channel.

Rather than encoding fairly still parts of the sequence with progressively smaller quantizer step sizes, we assume the user has preselected a minimum quantizer step size together with the resultant maximum quality. Therefore, if a scene is still, it will be encoded with quantizer $Q = Q_{\min}$, and its average encoded bit rate will be less than $\bar{R}$.

Thus, we choose the quantizer step size to be

$$Q = \max\left\{Q_{\min}, 2 * \mathrm{INT}\left(32 * \max\left\{B^e(t)/B^e_{\max}, \left[B^d_{\max} - B^d(t+LT)\right]/B^d_{\max}\right\}\right) + 2\right\}.$$

By selecting a minimum step size, the user sets an upper bound on the quality that can be received. (A given quantizer step size does not ensure a given image quality; however the two are closely related.) Although the user makes a small sacrifice in still image quality by choosing, say $Q_{\min} > 4$, such a choice may yield overall better quality.

B. Leaky-Bucket Channel Rate Control

We compare two rate control algorithms for the leaky bucket. Both use the basic procedure of Section V-B; however, they differ in the selection of R_i. The first algorithm is greedy, always choosing the maximum rate allowed by both the channel and the decoder-buffer fullness. The second algorithm is conservative, selecting a channel rate to gradually fill the decoder buffer if the leaky bucket is not full.

Greedy Leaky-Bucket Rate-Control Algorithm (GLB): For the greedy algorithm (GLB), we choose the maximum rate that both the channel and the decoder buffer will allow. Therefore,

$$R_i = \min\left\{B^e_i, E_{i-L} + B^d_{\max} - B^d_{i-L-1}, N_{\max} - N_{i-1} + \bar{R}\right\}. \quad (47)$$

The first constraint prevents the encoder buffer from underflowing, the second constraint prevents the decoder buffer from overflowing, and the third constraint prevents the leaky bucket from overflowing. Equation (47) is also used to obtain the estimated rates.

Considering only the encoder buffer fullness, the greedy leaky-bucket algorithm appears optimal. Because we transmit at the maximum rate allowed by both the network and the decoder buffer, the encoder buffer is kept as empty as possible, providing the most room to store newly encoded data. If we were to transmit at less than the maximum rate, then the bits that would remain in the encoder buffer would still need to be transmitted later. However, this algorithm may actually suffer in performance because it fills the bucket as fast as possible. The gain in performance provided by the leaky bucket could be of longer duration if the leaky bucket filled more slowly.

Conservative Leaky-Bucket Rate-Control Algorithm (CLB): The second rate-control algorithm for the leaky bucket is more conservative. The rate is chosen to either fill the leaky bucket, to fill the decoder buffer (if this will take fewer than $\bar{R}$ b), or to take L frames to fill the decoder buffer. The estimated rate R_i is assigned as follows:

$$tmp = E_{i-L} + B^d_{\max} - B^d_{i-L-1};$$
$$\text{if}(tmp > \bar{R})\, tmp = \bar{R} + \left(E_{i-L} + B^d_{\max} - B^d_{i-L-1} - \bar{R}\right)/L;$$
$$R_i = \min\left(tmp, N_{\max} - N_i + \bar{R}\right).$$

Because the rate is smaller than the maximum, we will see that we can extend the duration of the improvement provided by the leaky bucket, although we may limit the magnitude of the improvement.

VII. Simulation Examples

In this section, we numerically compare the video quality of the rate-control strategies described in Section VI to the video quality produced using a channel with a peak rate constraint. We examine the results for the table tennis and ferris wheel CIF sequences. The table tennis sequence contains several scenes with varying average bit rates, whereas the Ferris wheel sequence contains one scene with two periods in which a wheel is rotating in front of the camera and an interim period in which the wheels are rotating perpendicular to the camera near the edges of the image.

A. Simulation Approach

We simulate using the basic compression algorithm of RM8. The only modification is in the quantizer step-size selection (described in Section VI) and the buffer regulation algorithm. For each frame period, we estimate the current rate R_i and an upper bound on the future rate R^{ub}_{i+L}. We assume that both the current and future rates are divided as evenly as possible across the frame. They may not be exactly evenly divided if the encoder buffer started empty and the encoded bit rate at the beginning of the frame is small. The interval between recomputing buffer variables is 11 macroblocks. In addition to recomputing the encoder-buffer fullness, we also recompute the estimated future decoder-buffer fullness and the current leaky-bucket fullness.

For our examples, intraframe coding was used periodically every 60 frames, as well as when a scene change occurs. We use $P = 58$ to select the nominal average rate of $P * 64$ kb/s second, or 3.712 Mb/s. As a result, the nominal average number of transmitted bits per frame period is $\bar{R} = 123733$ b.

Of the six variables (the delay, the encoder- and decoder-buffer sizes, the bucket size, the nominal average channel rate, and the minimum quantizer step size), we examine the effect of changing the minimum quantizer step-size $Q_{\min}$ and the leaky bucket size $N_{\max}$. Default parameters are those specified by RM8: $B^e_{\max} = B^d_{\max} = LT\bar{R}$, where $L = 3$. We assume the network bucket is full initially, i.e., $N_0 = N_{\max}$.

B. Simulation Results

Quantizer Step-Size: We begin by illustrating the effect of a judicious choice of $Q_{\min}$ using the Ferris wheel sequence. Fig. 4(a) shows the peak signal-to-noise ratio

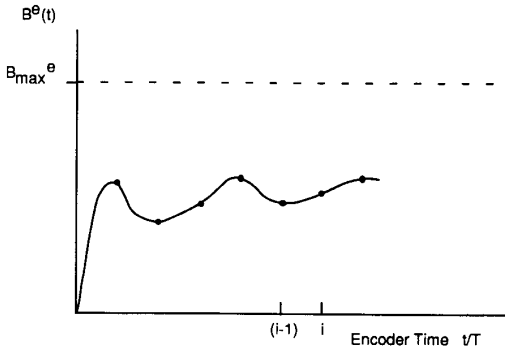

Fig. 2. Encoder buffer dynamics.

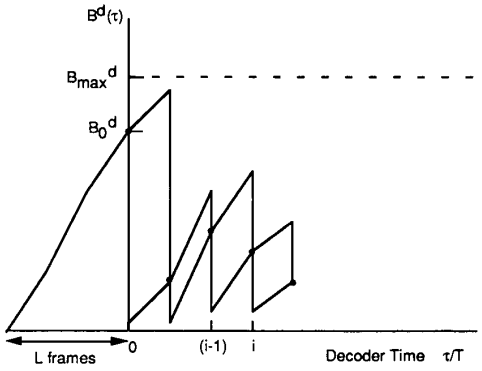

Fig. 3. Decoder buffer dynamics.

(PSNR = $10\log_{10}$[(mean squared error)/255*255] as a function of time when the minimum quantizer step sizes are $Q_{\min} = 8$ and 10, with a leaky-bucket size of $N_{\max} = 3\bar{R}$ using the GLB rate control algorithm. When $Q_{\min} = 8$, the quality varies greatly as a function of image content. However, when $Q_{\min} = 10$, the image quality is fairly constant from frame 60 onward. If the viewer cannot distinguish between images with a PSNR greater than 37, then the larger PSNR for frames 68–114 when $Q_{\min} = 8$ is effectively wasted.

The associated channel rates for each case are shown in Fig. 4(b). When $Q_{\min} = 8$, the number of bits transmitted by the channel during each frame period is constant, $R_i = \bar{R}\ \forall i$. When $Q_{\min} = 10$, we have a variable-channel bit rate, although it is constrained by the leaky bucket. The actual average channel rate when $Q_{\min} = 8$ is 3.712 Mb/s, whereas it is 3.651 Mb/s when $Q_{\min} = 10$. Alternatively, if we use $Q_{\min} = 12$ (not shown), the quantizer step size is always $Q_{\min}$. This produces an actual average rate of 3.083 Mb/s that is significantly below the nominal.

Leaky-Bucket Size: Next we examine the effect of changing the leaky bucket size $N_{\max}$. For simplicity, we first examine the two cases $N_{\max} = 0$ and $N_{\max} = 3\bar{R}$. When $N_{\max} = 0$, the channel rate can be no larger than $\bar{R}$; however, it can be less if the encoder buffer is empty.

The PSNR for the table tennis and ferris wheel sequences are shown in Fig. 5(a) and (b), respectively, for $N_{\max} = 0$ and for $N_{\max} = 3\bar{R}$ with the greedy leaky-bucket

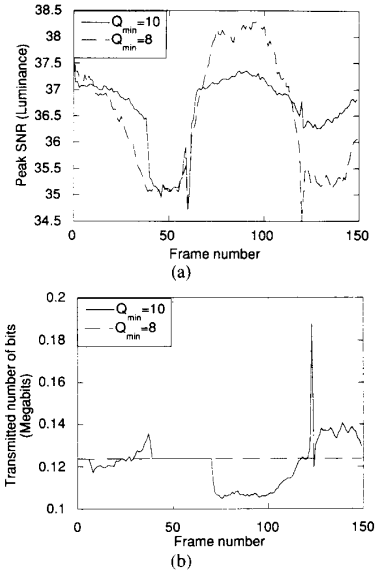

Fig. 4. Effect of $Q_{\min}$. (a) Peak SNR. (b) Transmitted number of bits per frame.

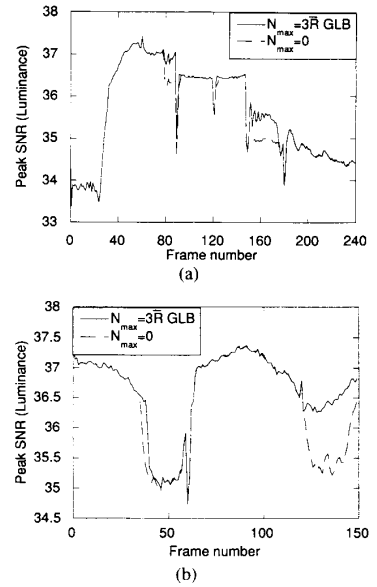

Fig. 5. Effect of leaky bucket. (a) Table tennis. (b) Ferris wheel.

(GLB) algorithm. For the table tennis sequence, $Q_{\min} = 8$, whereas for the ferris wheel sequence, $Q_{\min} = 10$. For each sequence, the PSNR when $N_{\max} = 3\bar{R}$ is always at least as good as when $N_{\max} = 0$. However, for the table tennis sequence, $N_{\max} = 3\bar{R}$ performs better than $N_{\max} = 0$ between frames 78 and 92 and between frames 151 and 176. For the ferris wheel sequence, $N_{\max} = 3\bar{R}$ performs slightly better than $N_{\max} = 0$ between frames 35 and 40 and performs significantly better from frame 120 onward. During the rest of each sequence, the presence of a nonzero leaky bucket has little effect.

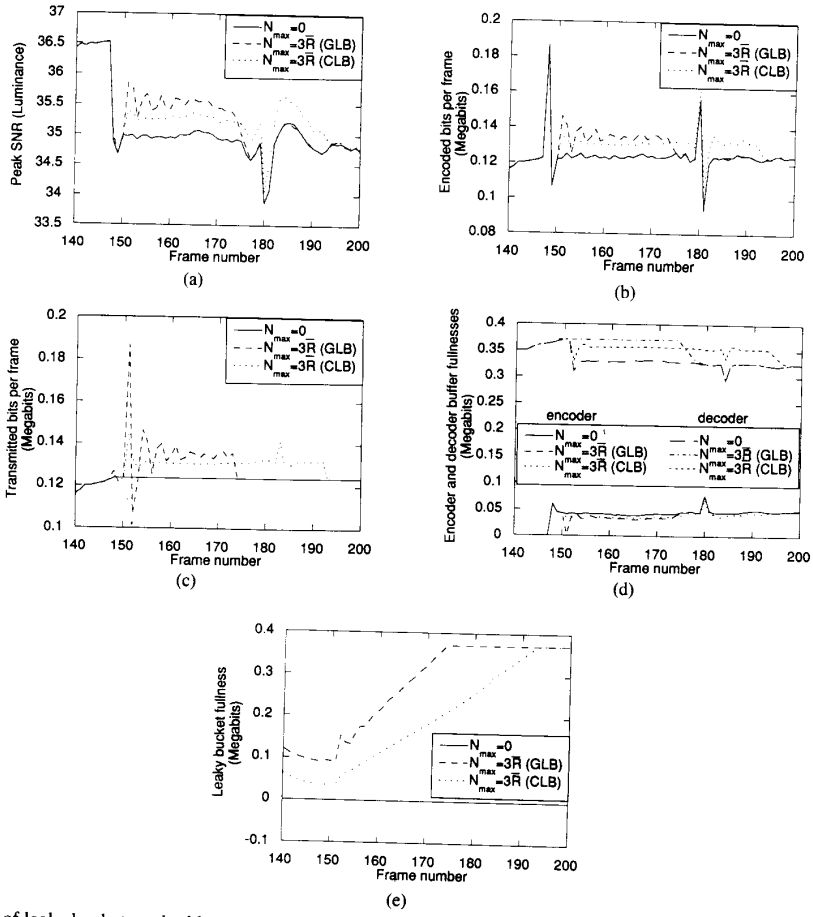

Fig. 6. Effect of leaky bucket, and table tennis. (a) PSNR. (b) Encoded bits per frame E_i. (c) Transmitted bits per frame R_i. (d) Encoder and decoder buffer fullnesses B_i^e and B_i^d. (e) Leaky bucket fullness N_i.

An explanation of the performance improvement of the leaky bucket is best found by simultaneously examining graphs of the PSNR, encoded bit rate E_i, the channel bit rate R_i, the encoder- and decoder-buffer fullnesses B_i^e and B_i^d, and the leaky-bucket fullness N_i. These are shown for the table tennis sequence in Fig. 6, and for the ferris wheel sequence in Fig. 7, expanded to show the regions of interest. The conservative leaky-bucket (CLB) algorithm with $N_{\max} = 3\overline{R}$ is also shown in the same figures.

For both sequences, at the start of the intervals shown, the leaky bucket is not full (Figs. 6(e) and 7(e)). Therefore, when the scene activity increases (whether because of a scene change (table tennis) or just increased activity (ferris wheel)), both algorithms (GLB and CLB) select the transmitted rate (Figs. 6(c) and 7(c)) to be larger than average. This keeps the encoder buffer (Figs 6(d) and 7(d)) emptier, which allows more encoded bits per frame (Figs. 6(b) and 7(b)), producing improved image quality (Figs. 6(a) and 7(a)). For the table tennis sequence, the improvement is approximately 0.5 dB over a range of 25–45 frames, whereas for the ferris wheel sequence, the improvement is over 1 dB for at least 30 frames.

The conservative algorithm (CLB) does not perform as well as the greedy algorithm (GLB) when both are performing better than no leaky bucket. However, the CLB performs better than no leaky bucket over a longer period since the leaky bucket does not fill as quickly.

Next, we examine the effect that increasing the bucket size has on the image quality. Fig. 8(a) and (b) show the PSNR and leaky-bucket fullnesses for $N_{\max} = 0, \overline{R}, 2\overline{R}, 3\overline{R}, 4\overline{R}, 5\overline{R}$ using the GLB for the table tennis sequence. In general, as the bucket size increases, performance does not decrease. That is, for a particular frame, $N_{\max} = 3\overline{R}$ may not perform better than $N_{\max} = 2\overline{R}$; however, as $N_{\max}$ increases, the duration of improvement lengthens. $N_{\max} = \overline{R}$ provides significant improvement over $N_{\max} = 0$ for only 7 frames, whereas $N_{\max} = 5\overline{R}$ outperforms $N_{\max} = 0$ for over 40 frames.

VIII. Discussion and Conclusions

We have presented constraints on the variable transmission rate and the encoded rate of compressed video that is imposed by encoder and decoder buffers. A system is presented that controls both the encoded and transmitted bit rates to satisfy these constraints. This solution is

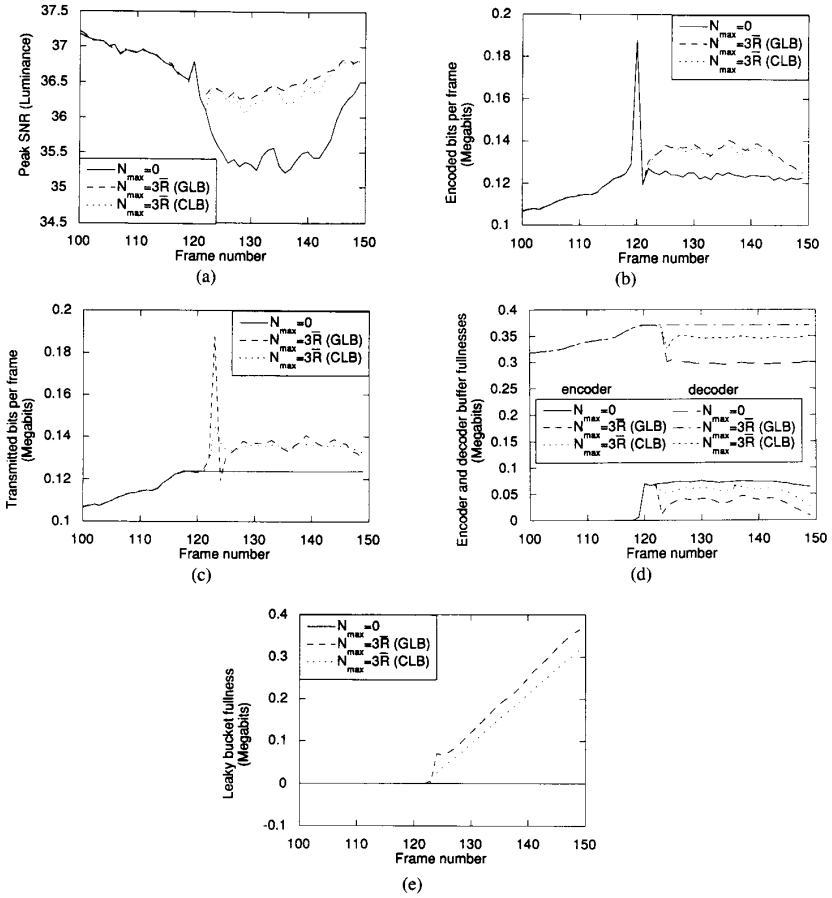

Fig. 7. Effect of leaky bucket, and Ferris wheel. (a) PSNR. (b) Encoded bits per frame E_i. (c) Transmitted bits per frame R_i. (d) Encoder and decoder buffer fullnesses B_i^e and B_i^d. (e) Leaky bucket fullness N_i.

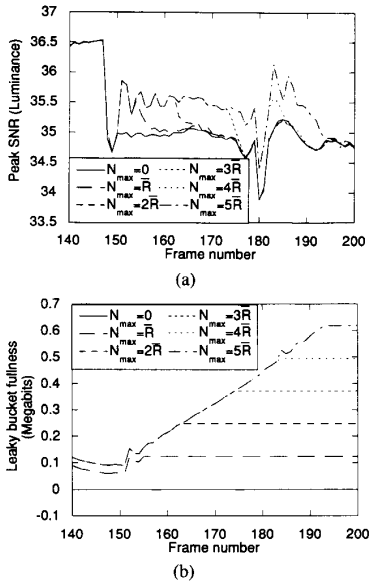

Fig. 8. Effect of leaky-bucket size, table tennis. (a) Peak SNR. (b) Leaky-bucket fullness.

essential if the decoder buffer size is fixed a priori. We presented an encoder rate control algorithm, and two channel rate control algorithms for the leaky bucket channel. The performance of each was illustrated on video sequences.

In general, the leaky bucket will improve image quality in two situations: at an intraframe when the leaky bucket is not full, and when the scene activity increases while the leaky bucket is not full. However, to obtain the improvement, the selection of $Q_{\min}$ is critical. If $Q_{\min}$ is too small, the leaky bucket will be full when the scene activity increases. Alternatively, if $Q_{\min}$ is too large, the quality will be limited by $Q_{\min}$ rather than the encoder buffer fullness.

Also, the magnitude of the quality improvement is often not significant. For the results shown, the magnitude of the improvement is limited by the decoder buffer fullness. More significant improvements may be available if the decoder buffer is not kept nearly as full.

More sophisticated encoder rate control algorithms are possible, including those that use adaptive quantization based on scene content. More sophisticated channel rate control algorithms are also possible. For example, the rate

control algorithm could forecast when an intraframe will be encoded and attempt to empty the encoder buffer in advance. However, the results presented here do indicate the extent to which quality improvement is possible by using a leaky-bucket channel constraint.

Overall, if the user selects a maximum quality they are willing to receive, using a leaky-bucket channel instead of a constant-rate channel may allow them to achieve overall better image quality. However, because of the presence of the channel constraint, it will not be possible to obtain truly constant image quality.

REFERENCES

[1] J. C. Darragh and R. L. Baker, "Fixed distortion subband coding of images for packet-switched networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 789–800, June 1989.

[2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 771–781, June 1989.

[3] A. R. Reibman, "DCT-based embedded coding for packet video," *Image Communication*, June 1991.

[4] G. Karlsson and M. Vetterli, "Packet video and its integration into the network architecture," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 739–751, June 1989.

[5] F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, "Variable bit-rate coding of video signals for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp. 801–806, June 1989.

[6] M. Nomura, T. Fujii, and N. Ohta, "Layered packet-loss protection for variable rate video coding using DCT," in *Second Int. Workshop on Packet Video*, 1988.

[7] G. Ramamurthy and B. Sengupta, "Modeling and analysis of a variable bit rate video multiplexer," in *Proc. 7th ITC Seminar*, 1990.

[8] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 325–334, April 1991.

[9] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the 'leaky bucket' policing mechanism in ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 335–342, April 1991.

[10] L. Dittmann, S. B. Jacobsen, and K. Moth, "Flow enforcement algorithms for ATM networks," *IEEE J. Selected Areas Commun.*, vol. 9, no. 3, pp. 343–350, April 1991.

[11] A. E. Eckberg, Jr., D. T. Luan, and D. M. Lucantoni, "An approach to controlling congestion in ATM networks," *Int. J. Digital and Analog Communication Syst.*, vol. 3, pp. 199–209, 1990.

[12] "Description of reference model 8 (RM8)," Tech. Rep. 525, CCITT SGXV Working Party XV/4, 1989.

[13] B. Voeten, F. V. der Putten, and M. Lamote, "Preventive policing in video codecs for ATM networks," in *Fourth Int. Workshop on Packet Video*, pp. G1.1–G1.6, 1991.

Amy R. Reibman (M'87) was born in Schenectady, NY, on April 17, 1964. She received the B.S., M.S., and Ph.D. degrees in electrical engineering from Duke University, Durham, NC, in 1983, 1984, and 1987, respectively.

From 1988 to 1991, she was an Assistant Professor in the Department of Electrical Engineering at Princeton University, Princeton, NJ. She is currently a Member of the Technical Staff in the Visual Communications Research Department at AT&T Bell Laboratories, Holmdel, NJ. Her research interests include video compression and packet video.

Dr. Reibman is a member of Sigma Xi, Eta Kappa Nu, and Tau Beta Pi.

Barry G. Haskell (F'87) received the B.S., M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1964, 1965, and 1968, respectively.

From 1964 to 1968 he was a Research Assistant in the University of California Electronics Research Laboratory, with one summer being spent at the Lawrence Livermore Laboratory. Since 1968 he has been at AT&T Bell Laboratories, Holmdel, NJ, and is presently Head of the Visual Communications Research Department. He has also taught graduate courses at Rutgers University, New Brunswick, NJ, City College of New York and Columbia University, New York. His research interests include digital transmission and coding of images, videotelephone, satellite television transmission, medial imaging, as well as most other applications of digital image processing. He has published over 35 papers on these subjects and has 20 patents either granted or pending. He is also the coauthor of the book *Digital Pictures —Representation and Compression.*

Dr. Haskell is a member of Phi Beta Kappa and Sigma Xi, the CCITT SG15 Experts Group on Videotelephony, and the ISO WG11 Motion Picture Experts Group (MPEG). He is an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.