CCITT
STUDY GROUP XV WP XV/1
Experts Group for ATM Video Coding
Paris 23-24 May 1991

Document # AVC-56

English Version

SOURCE: AT&T Bell Laboratories
TITLE: Constraints on Variable Bit-Rate Video for ATM Networks
PURPOSE: Informational

Constraints on Variable Bit-Rate Video for ATM Networks

Barry G. Haskell and Amy R. Reibman

May 21, 1991

Abstract

We begin by presenting conditions that ensure 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. Finally, we describe how these ideas might be implemented in a system.

1 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 contribution, we examine the constraints imposed on the encoded video bit-rate as a result of encoder and decoder buffering. In particular, we show that while 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, such 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. Since 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.

2 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-rate, 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 a contract 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 the user does not submit too many. A policing function monitors the user output, and either drops packets in excess of the 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 [2,3,4,5,6] 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 [8,9,10]. Therefore, in this contribution 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, 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. This is called source shaping.

3 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. In addition, 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. Since 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 every one-thirtieth of a second), but no new frame has been decoded. Therefore, three things must happen simultaneously: (a) the decoder buffer is empty, (b) the next-frame frame memory is not full, and (c) it is time to display a freshly decoded frame. For this reason, we discretize the problem using the uncoded frame period.

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 see here, the problem becomes more difficult when the channel may transmit a variable bit-rate, for example, when transmitting video across packet (ATM) networks.

3.1 Buffer dynamics

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.,

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

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

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

First, we discretize the problem by defining $E_i (i = 1, 2, \ldots)$ 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 seconds 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.

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

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

$$B^d_0 = \sum_{j=1}^{L} R_j. \tag{10}$$

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, 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 $B_{i-1}^d - E_i$ or $B_i^d - E_{i+1}$.

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

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

i+L 時点でわかる値

$$B_i^d = \sum_{j=i+1}^{i+L} R_j - B_i^e \tag{13}$$

or

$$B_i^d = \sum_{j=i+1}^{i+L} E_j - (\sum_{j=1}^{i+L} E_j - \sum_{j=1}^{i+L} R_j)$$

$$B_i^d = \sum_{j=i+1}^{i+L} E_j - B_{i+L}^e \tag{14}$$

The first (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. The second (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.

3.2 Buffer verification

We now combine equations from the previous subsection 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

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

$$R_i - B^e_{i-1} \leq E_i \leq B^e_{max} + R_i - B^e_{i-1} \tag{16}$$

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 [11]. 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

$$0 \geq -B^e_{i-1} - E_i + R_i \geq -B^e_{max}$$

$$\underset{\text{encoder overflow condition}}{E_i - \left(B^e_{max} - B^e_{i-1}\right)} \leq R_i \leq \underset{\text{encoder underflow condition}}{B^e_{i-1} + E_i} \tag{17}$$

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

$$0 \leq B^d_{i-1} + R_{i+L} - E_i \leq B^d_{max} \tag{18}$$

$$R_{i+L} + B^d_{i-1} \geq E_i \geq R_{i+L} + B^d_{i-1} - B^d_{max} \tag{19}$$

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.

$$E_i - B^d_{i-1} \leq R_{i+L} \leq E_i + \left(B^d_{max} - B^d_{i-1}\right)$$

or, for $i > L$

$$\underset{\text{decoder underflow condition}}{E_{i-L} - B^d_{i-L-1}} \leq R_i \leq \underset{\text{decoder overflow condition}}{E_{i-L} + \left(B^d_{max} - B^d_{i-L-1}\right)} \tag{20}$$

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),

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

$$E_i \leq E_{i-L} + \left(B^e_{max} - B^e_{i-1}\right) + \left(B^d_{max} - B^d_{i-L-1}\right) \qquad (21)$$

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

$$0 \leq \sum_{i=j+1}^{j+L} E_i \leq B^d_{max} + B^e_{max}. \qquad (22)$$

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.

4 Buffer verification for channels with rate constraints

4.1 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 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

$$B^d(0) = B^d_0 = LRT.$$

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

$$B^d_j = B^d_0 - B^e_j \qquad (23)$$

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

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

$$\begin{aligned} &\neq B_0^d + \sum_{i=1}^{j} R_i - \sum_{i=1}^{j} E_i \\ &= B_0^d - B_j^e \end{aligned}$$

Since B_j^e 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 (23), 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 $B_{max}^d = LRT + \delta$ where δ is small.

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

$$0 \leq B_j^d = LRT - B_j^e,$$

or, provided $B_j^e \leq LRT$. Therefore, if we choose $B_{max}^e = LRT$, 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 doesn't overflow or underflow simply by ensuring that the encoder buffer doesn't 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

$$B_j^e = \sum_{k=1}^{j} E_k - jRT \leq LRT.$$

Reorganizing,

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

so that

$$L \geq \sum_{k=1}^{j} E_k/RT - j, \forall j, \tag{24}$$

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, (24) also indicates the trade-off between the allowable decoder delay and the image quality.

Insight into how (24) involves the variability in the number of bits per coded frame can be seen by examining the two extremes of variability. First, suppose all frames have the same number of bits, $E_i = RT$. Then, $L \geq 0$, 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 has 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.

4.2 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

$$N_i = N_{i-1} + R_i - \bar{R} = \sum_{j=1}^{i} R_j - i\bar{R}. \tag{25}$$

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

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

or

$$R_i \leq N_{max} - N_{i-1} + \bar{R} = N_{max} - \sum_{j=1}^{i-1} R_j + i\bar{R}. \tag{26}$$

Equation (26) defines the leaky-bucket constraint on the rate that is input to the network.

Combining (26) with (17), which constrains the rate to prevent encoder buffer underflow and overflow, we have

$$\begin{aligned}
E_i &\leq N_{max} + B^e_{max} - \sum_{j=1}^{i-1} R_j + i\bar{R} - B^e_{i-1} \\
&\leq N_{max} + B^e_{max} + i\bar{R} - \sum_{j=1}^{i-1} R_j - (\sum_{j=1}^{i-1} E_j - \sum_{j=1}^{i-1} R_j) \\
&\leq N_{max} + B^e_{max} + i\bar{R} - \sum_{j=1}^{i-1} E_j \\
E_i &\leq B^E_{max} + \bar{R} - B^E_{i-1}
\end{aligned} \tag{27}$$

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 of smaller size B^e_{max} from overflowing *or* underflowing with constant rate $\bar{R}$, 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 (26) above. As a result, we may not actually be able to obtain the 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 $L = N_{max}/\bar{R}$, and $N_{max} = B^e_{max} = B^d_{max}$. However, recall that we will certainly be paying for both N_{max} and $\bar{R}$.

5 Coding System with Rate Control and Channel Constraint

We describe a means whereby the number of encoded bits for each video frame and the number of bits transmitted across the variable rate channel are selected jointly. This is necessary for two reasons, both of which arise because the channel no longer has constant rate, but can transmit a variable rate.

First, the selection of the encoder bitrate is controlled not only by the encoder buffer fullness (as traditionally done), but also by the decoder buffer fullness. In video coding systems with a constant rate channel, it was unnecessary to be concerned about the decoder buffer fullness since the decoder buffer fullness tracked the encoder buffer fullness, $B^d_i = LTR - B^e_i$.

Second, in the traditional systems the number of bits transmitted across the channel was selected to be as large as possible. The channel rate was constrained to be less than a peak value, so the best video quality could be obtained by transmitting at this peak bitrate value. However, the channel under consideration in this implementation may not be constrained only by a peak bitrate.

An example of a possible constraint is that defined by a leaky bucket algorithm [8]. As a result, it may be desirable to transmit at a rate less than the maximum allowable rate. Thus, the rate control device described in this implementation is original, because it may select a transmission rate less than the maximum allowed by the channel. The actual transmission rate selection process is governed not only on the basis of the channel constraint, but also on the decoder FIFO buffer fullness and other considerations.

5.1 Block Diagram

FIG. 1 describes a system incorporating these concepts. In FIG. 1, a video signal is applied to the video encoder 10. The video encoder 10 produces an encoded video bit-stream that is stored in the encoder FIFO buffer 20 before being transmitted via channel interface 80 to the variable rate channel 30. After being transmitted across the variable rate channel 30, the video bit-stream is stored in the decoder FIFO buffer 40. The bit-stream from the decoder FIFO buffer 40 is input to the video decoder 50. The video decoder 50 outputs a video signal. The delay from encoder buffer input to decoder buffer output, exclusive of channel delay, is exactly LT seconds, where T is the time period of one uncoded video frame. L is not necessarily an integer, although for simplicity, we will describe the implementation here as if it were.

The encoder FIFO buffer 20 and the decoder FIFO buffer 40 have finite sizes, B^e_{max} and B^d_{max}, which are known apriori. In addition, the value of the delay L is also know apriori.

The video encoder 10 encodes the video signal using any method that allows the number of bits produced by the video encoder 10 to be controlled (see for example [11]). A range indicating the number of bits that can be produced by video encoder 10 is provided by the encoder rate control device 70. The video encoder 10 produces a bit-stream that contains E_i number of bits in one frame period. This number of bits is within the bound given by the encoder rate control device 70. The index i designates the frame number from some arbitrary starting point. The E_i bits are input to an encoder FIFO buffer 20 and stored until they are transmitted.

The channel rate control device 60 takes as input the actual number of bits output in each frame period by the video encoder 10. It estimates channel rates $R_i, \ldots R_{i+L-1}$, describing the number of bits that will be transmitted across the channel 30 in the following L frame periods. Channel rate control 60 then sends the estimated value of R_i to channel 30 as R_{req} on line 63. If the channel grants the request, then the actual channel rate R_i returned on line 67 is equal to the estimated value. Otherwise, $R_i < R_{req}$.

Under instruction from channel rate control 60, channel interface 80 conveys bit-stream CODEIM to channel 30 at rate R_i for the period during which it pertains. If $R_i < R_{req}$, on occasion and under control of channel rate control 60 some information might have to be discarded from bit-stream CODEIM, so that CODETR, the output of channel interface 80, might not always be exactly equal to CODEIM. In some systems, channel 30 itself might perform this discarding of information, so that its output may not equal its input.

Such discarding of information is to be regarded as an emergency measure to be undertaken only in case of severe and unpredictable congestion of the channel. Under more predictable constraints on channel transmission rate, it is the express purpose of this invention to avoid discarding of information. Instead, data control is exerted in a relatively harmless manner by the video encoder.

Also, on occasion, encoder buffer 20 may become empty, whereupon channel interface 80 immediately terminates transmission. In most cases, this will cause a reduction of R_i.

The selection of the channel rates by the channel rate control device 60 is governed by the need to prevent the encoder FIFO buffer 20 AND the decoder FIFO buffer 40 from overflowing or underflowing. It is also governed by a possibly time-varying bit-rate constraint on the channel 30. Additional factors that can be considered in the selection of the channel rate are the cost of transmitting each bit, and the fact that for some channel constraints (for example, the leaky bucket [8]), the channel rate could be conserved now to produce a less stringent channel rate constraint later. This might be desirable if a frame that is encoded using intraframe techniques will need to be transmitted soon.

The encoder rate control device 70 computes a bound on the number of bits that the video encoder 10 may produce without overflowing or underflowing either the encoder FIFO buffer 20 or the decoder FIFO buffer 40. It takes as input the actual number of bits E_i output in each frame period by the video encoder 10. It also takes as input the channel rate values that are selected by the channel rate control device 60. The bound output by the encoder rate control device 70 is computed to guarantee that neither the encoder FIFO buffer 20 nor the decoder FIFO buffer 40 overflow or underflow.

We now describe how channel rate control 60 and encoder rate control 70 operate.

5.2 Flow Chart

The principles of buffer control for variable rate channels with leaky bucket rate constraints are now illustrated by means of the flow chart in Fig. 2.

Step 801 initializes variables at time $t = 0$ prior to encoding frame $i = 1$. B_i^e is the encoder buffer fullness at time $t = iT$, B_i^d is the decoder buffer fullness at time $\tau = iT$, and N_i is the leaky bucket fullness at time $t = iT$. All are initialized to zero (0).

Step 802 estimates channel rates, leaky bucket sizes and decoder buffer sizes for the next L video frames. For channel rates we utilize inequalities (20) and (26), where for $k \leq 0$, $E_k = 0$. Leaky bucket and decoder buffer sizes are given, respectively, by (25) and (12). Rewriting them, we get for $j = i, i+1, \ldots i+L-1$

$$\underbrace{E_{j-L} - B^d_{j-L-1}}_{\text{decoder underflow condition}} \leq R_j \leq \underbrace{E_{j-L} + \left(B^d_{max} - B^d_{j-L-1}\right)}_{\text{decoder overflow condition}} \tag{28}$$

$$R_j \leq N_{max} - N_{j-1} + \bar{R} \tag{29}$$

$$N_j = N_{j-1} + R_j - \bar{R} \tag{30}$$

$$B^d_{j-L} = B^d_{j-L-1} + R_j - E_{j-L} \tag{31}$$

In most cases, for $j < i + L - 1$ we can simply use previous estimates, in which case we need only make evaluations for $j = i + L - 1$. However, as we shall see below, B_{i-L-1} can sometimes change, which may make re-evaluation of all estimates desirable.

In general, a value of $R_j \geq 0$ that is halfway between the lower bound of (28) and whichever upper bound of (28) and (29) is smaller is a conservative choice. However, if we know that a frame with a large number of bits is imminent or has just occurred, we may wish to choose larger values. If we wish to conserve bits as much as possible, as on a video disk, for example, we may wish to choose smaller values.

For $i \leq L$, no frames are being decoded, and the decoder buffer is only filling. In general, the sum of $R_1, \ldots R_L$ should be chosen to exceed the expected encoded bit rate of the first few frames in order to avoid decoder buffer underflow.

Step 803 simply sets R_{req} to the estimated value for R_i and requests that rate from channel 30.

Step 804 determines whether or not the request is granted. If it is not,

Step 805 determines what value of R_i is granted by the channel, and

Step 806 evaluates inequality (28) with $j = i$ to see if decoder buffer underflow would occur with this allowable value of R_i. If decoder buffer underflow would occur, then in

Step 807, channel rate control 60 causes channel interface 80 to discard data in a selective fashion, trying to minimize deleterious effects on the picture, in order to reduce the value of E_{i-L}, and thus satisfy inequality (28) and avoid catastrophic decoder buffer underflow.

Step 808 re-computes estimates of channel rates, leaky bucket sizes and decoder buffer sizes using the smaller value of R_i supplied by channel 30 on line 67. Step 808 is identical to step 802.

Step 809 verifies the value of $R_i = R_{req}$.

Step 810 calculates an upper bound on R_{i+L} using the leaky bucket constraint (29)

$$R_{i+L} \leq UBR_{i+L} = N_{max} - N_{i+L-1} + \bar{R} \tag{32}$$

Step 811 calculates an upper bound on E_i using constraints on encoder buffer overflow from inequality (16) and decoder buffer underflow from inequality (19).

$$E_i \leq B^e_{max} + R_i - B^e_{i-1} \tag{33}$$

$$E_i \leq UBR_{i+L} + B^d_{i-1} \tag{34}$$

The minimum of these two upper bounds on E_i is output by encoder rate control 70 as the signal RANGE to the video encoder 10.

Step 812 commences video encoding of frame i according to the bound on E_i and transmission at rate R_i.

Step 813 finishes video coding of frame i and transmission. It then inputs the actual value of E_i to channel rate control 60 and encoder rate control 70.

Step 814 uses the actual value of E_i to evaluate inequality (16) to see if encoder buffer underflow occurred during the coding of frame i. If it did, channel rate control 60 and channel interface 80 terminate transmission early, and

Step 815 calculates the new lower value of R_i as

$$R_i = B^e_{i-1} + E_i \tag{35}$$

(36)

Step 816 uses actual values of E_i and R_i to compute actual values of B^e_i, N_i and B^d_{i-L} using equations (8), (30) and (31), respectively.

Step 817 increments i in preparation for encoding the next frame. If at this point, $R_i = R_{req}$ then all values previously estimated in step 802 are still valid, and for the next frame, only the latest time interval need be considered. If $R_i \neq R_{req}$, then B^d_{i-L} has changed, and all values previously estimated in step 802 must be recomputed.

References

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

[2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE Journal on Selected Areas in Communications*, 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 Journal on Selected Areas in Communications*, 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 Journal on Selected Areas in Communications*, 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 International Workshop on Packet Video*, 1988.

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

[8] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE Journal on Selected Areas in Communications*, 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 Journal on Selected Areas in Communications*, 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 Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 343–350, April 1991.

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

FIG. 1

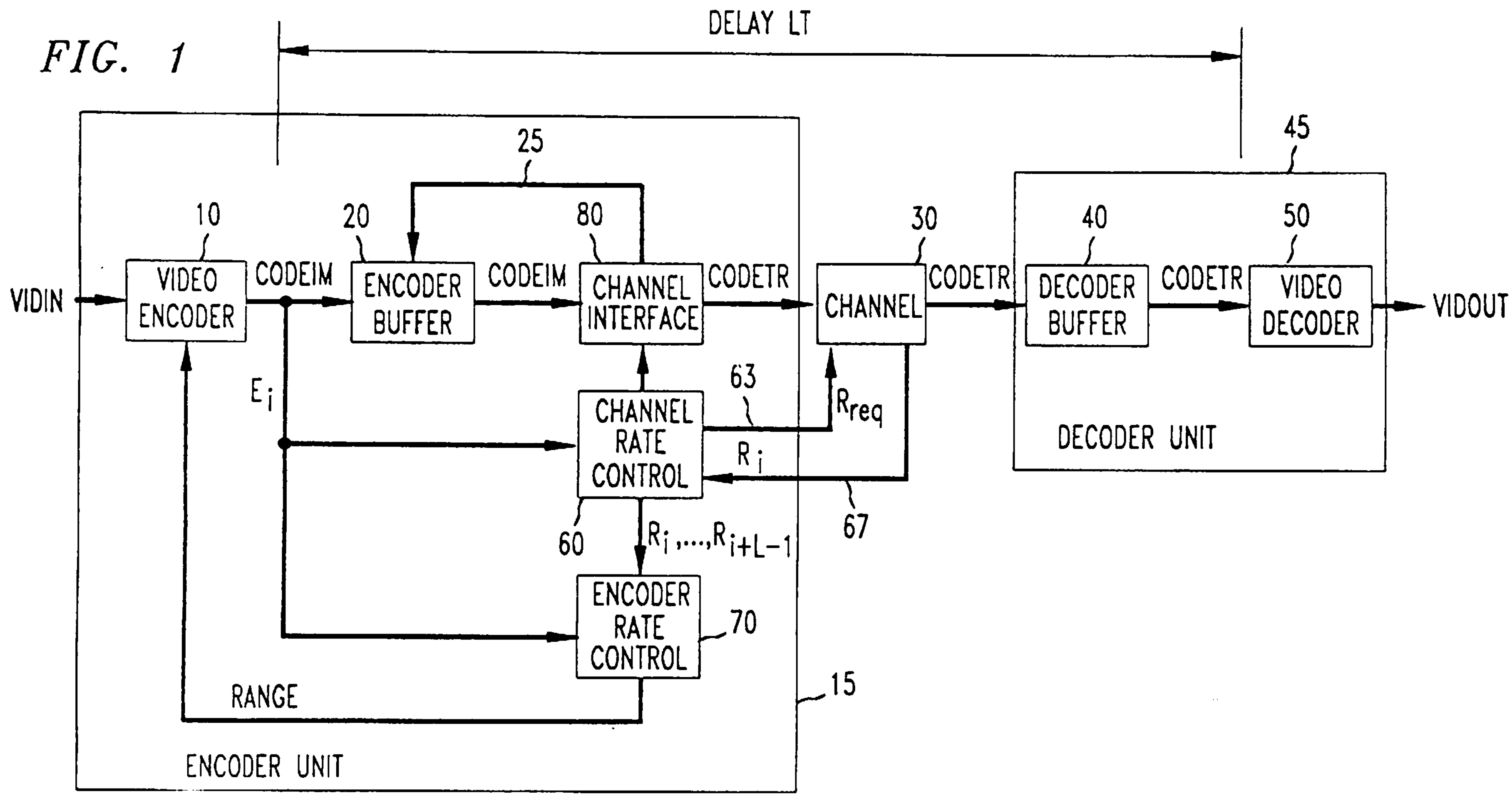
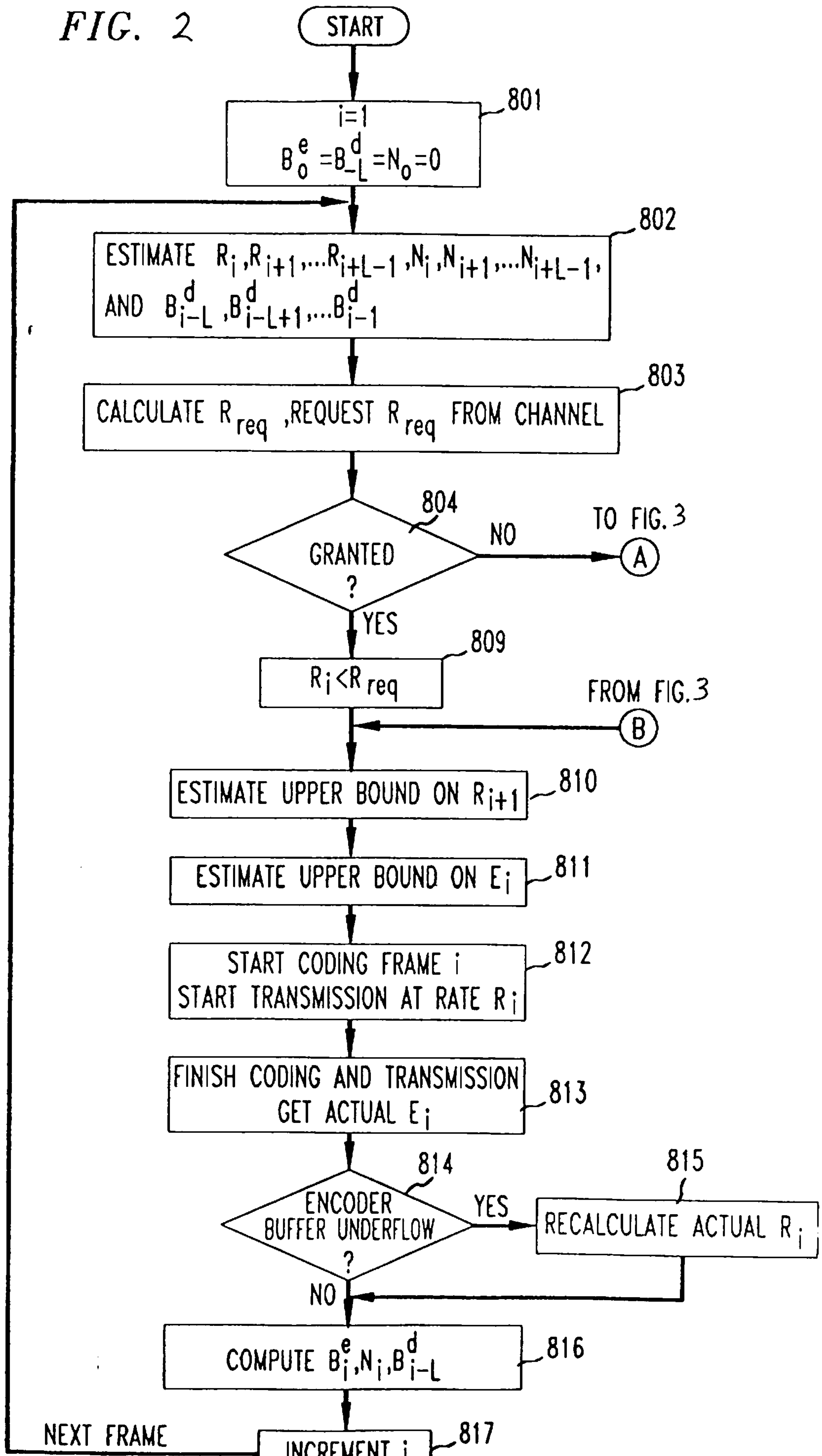
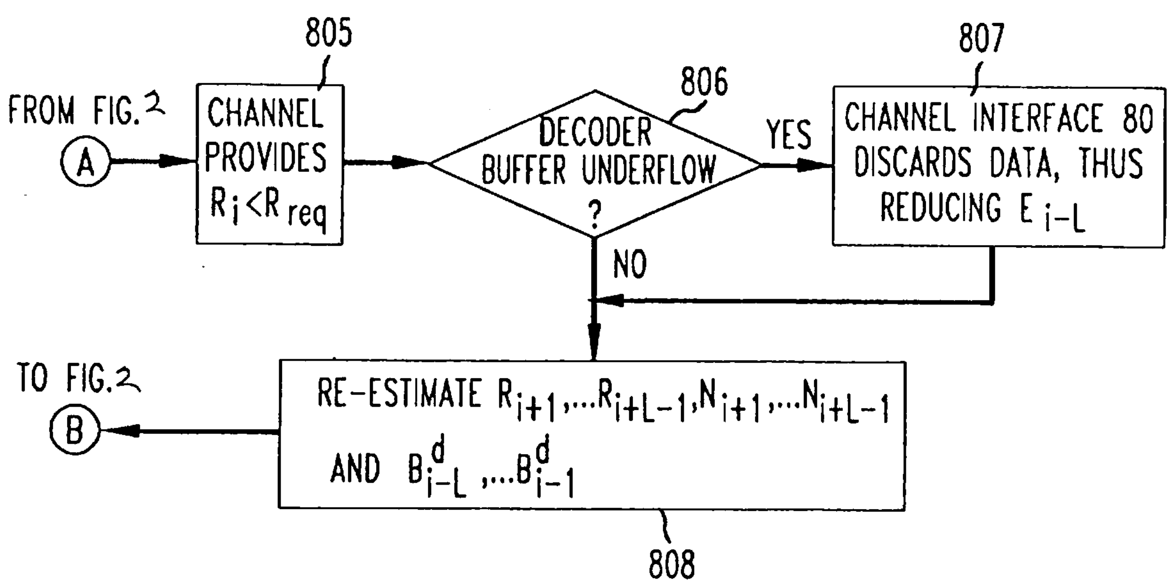

DELAY LT
VIDIN
10
VIDEO ENCODER
CODEIM
20
ENCODER BUFFER
25
CODEIM
80
CHANNEL INTERFACE
CODETR
30
CHANNEL
CODETR
45
40
DECODER BUFFER
CODETR
50
VIDEO DECODER
VIDOUT
DECODER UNIT
E_i
CHANNEL RATE CONTROL
63
R_{req}
R_i
67
60
$R_i,\ldots,R_{i+L-1}$
ENCODER RATE CONTROL
70
RANGE
15
ENCODER UNIT

FIG. 2



START
i=1
$B_0^e = B_{-L}^d = N_0 = 0$
801
ESTIMATE $R_i, R_{i+1}, \ldots R_{i+L-1}, N_i, N_{i+1}, \ldots N_{i+L-1}$, AND $B_{i-L}^d, B_{i-L+1}^d, \ldots B_{i-1}^d$
802
CALCULATE R_{req}, REQUEST R_{req} FROM CHANNEL
803
GRANTED ?
804
NO
TO FIG. 3
A
YES
$R_i < R_{req}$
809
FROM FIG. 3
B
ESTIMATE UPPER BOUND ON R_{i+1}
810
ESTIMATE UPPER BOUND ON E_i
811
START CODING FRAME i
START TRANSMISSION AT RATE R_i
812
FINISH CODING AND TRANSMISSION
GET ACTUAL E_i
813
ENCODER BUFFER UNDERFLOW ?
814
YES
RECALCULATE ACTUAL R_i
815
NO
COMPUTE B_i^e, N_i, B_{i-L}^d
816
NEXT FRAME
INCREMENT i
817

FIG. 3


FROM FIG.2
A
805
CHANNEL PROVIDES $R_i < R_{req}$
806
DECODER BUFFER UNDERFLOW ?
YES
NO
807
CHANNEL INTERFACE 80 DISCARDS DATA, THUS REDUCING E_{i-L}
TO FIG.2
B
RE-ESTIMATE $R_{i+1},\ldots R_{i+L-1}, N_{i+1},\ldots N_{i+L-1}$ AND $B^d_{i-L},\ldots B^d_{i-1}$
808