CCITT
STUDY GROUP XV WP XV/1
Experts Group for ATM Video Coding
Singapore, 6 January 1992

Document # AVC- 202

English Version

SOURCE: AT&T Bell Laboratories
TITLE: Selection of Traffic Descriptors and the Impact of Buffering
PURPOSE: Informational

Selection of Traffic Descriptors and the Impact of Buffering

Amy Reibman
AT&T Bell Laboratories

January 2, 1992

1 Introduction

In this contribution, we clarify the distinction between a traffic descriptor (TD) and the Usage Parameter Control (UPC) that provides the network policing. Next, we present algorithms to select parameters for TD's (sliding window and leaky bucket) such that for a given VBR sequence, the UPC will find no violations. We demonstrate that an additional peak rate constraint is advantageous, and show that the leaky bucket is a better choice than the sliding window. In addition, we show that buffering in the video system can reduce the necessary parameters, especially for small window sizes. Results are presented using several video teleconferencing sources each having a duration of 5 or 10 minutes.

The system we consider is as in AVC-56 [1], and shown in Figure 1. At the encoder and decoder, encoded bits are buffered. The number of encoded bits per frame is E_i, and the number of transmitted bits per frame period, which is selected by the rate control device, is R_i. The rate control device ensures that the coded bit-stream can be transmitted without violating any constraints. The decoder contains one frame memory to store the frame currently being decoded, which is the next frame to be displayed.

All arguments in this contribution are presented using a one-layer codec; however, they extend easily to the high-priority output of a two-layer codec. In addition, while we describe the problem using the frame period as the discrete time unit, smaller sampling intervals are possible.

2 VBR Video and Network Policing

Variable bit-rate (VBR) video is expected to be advantageous both for the network and for the user. The network should, through statistical multiplexing, be able to carry

more VBR video calls than constant bit-rate (CBR) video calls. In addition, the user is expected to obtain better quality VBR video than CBR video, even when both systems have the same average rate.

However, if all streams have completely unconstrained bit-rate, the network could have too much traffic to transport. Hence, some form of service contract between the network and the user is necessary. In the service contract, the network would agree to transport all (high-priority) traffic that the user submits, provided the user does not submit traffic that exceeds a pre-agreed quantity. The user understands two things from the service contract. First, the network *will* transport (with agreed-to cell-loss rate (CLR)) any traffic the user submits not in excess of the agreement. Second, if excess traffic *is* submitted, it may not be transmitted. Because, for video, some information is vital to the decoder and should not be dropped, the video terminal *must not* submit excess traffic as high priority.

The notion of the service contract is tightly connected with the traffic descriptor (TD) and the Usage Parameter Control (UPC). The traffic descriptor is the means by which both the video terminal and the network can characterize the quantity of traffic submitted. Commonly proposed TD's are the leaky bucket and sliding window, which are described below. The TD parameters announced by the video terminal will be used by the network for call admission.

The UPC is the control mechanism that is typically referred to as the "network policing function". It is the means by which the network is able to meet the guaranteed quality-of-service (QOS) of the non-excessive traffic. The UPC consists not only of a TD, which is used to monitor the incoming traffic, but also the control action that is applied to the excessive traffic. The control action could either be to immediately drop excessive traffic, or to mark excessive traffic as low-priority.

As far as the video system is concerned, it is immaterial which control action the UPC takes for excessive traffic. Neither the immediate loss of a high-priority packet nor a probable later loss of a high-priority packet can be allowed to happen. The video system must therefore ensure that all high-priority traffic submitted to the network will not exceed the TD used by the UPC. It is the goal of this contribution to identify useful parameter values for two commonly proposed TD for conferencing video.

3 Video

The video used in these examples was filmed at an actual meeting. The output of a CCD camera was digitized and recorded on a D1 tape machine, in order to create repeatable digital source material. The CCIR601 format video was converted into CIF (240 lines) using the MPEG-SM filter. The video was coded using a one-layer, H.261-compatible codec. The codec uses exhaustive motion estimation with ± 15 search range, a constant

quantizer step-size of 8, and intra/inter/MC decisions for each macroblock as in RM8. The first frame is coded intraframe, and all remaining frames are coded predictively. Within each frame, 3 macroblocks are transmitted intraframe.

Three sequences are used here. Sequence A is 10 minutes long and consists of a person listening and interspersing occasional comments and questions. Sequences B and C are both 5 minutes long, and each contain one active participant. In sequence B, the subject is constantly moving, while in sequence C, the subject moves only occasionally.

In all the sequences, intervals with high activity do not necessarily imply that the subject is speaking. Often, a high activity region corresponds to a period in which the subject is silent, and a low activity region corresponds to when they speak.

4 Traffic descriptor parameter selection

The sliding window and leaky bucket were compared in [2] using hypothetical video traffic. Here, we compare them using actual video conferencing data.

4.1 Sliding window

The sliding window can be described by two parameters, the time duration S_{win}, and the maximum number of bits (or bytes or packets) that can be transmitted in that time window, W_{max}. An alternate description could include the average bit-rate, W_{max}/S_{win}. We measure S_{win} in frame periods for simplicity; in general, the window duration need not be a multiple of the frame period. Mathematically, the constraint on the channel rate that is imposed by the sliding window is

$$\sum_{j=k+1}^{k+S_{win}} R_j \leq W_{max}, \tag{1}$$

for all k.

For a given sequence of R_i, it is a simple matter to determine the size of the sliding window parameters necessary to pass a given bit-stream without violation by computing the maximum number of bits in any window for each window length of interest [3]. If we assume $R_i = E_i$, then the results are shown in Figures 2–4 for 3 teleconferencing sequences, where the nominal average rate (W_{max}/S_{win}) is plotted against the window size S_{win}.

In each figure, the minimum value on the y-axis is the actual average rate of the sequence. The grid-lines on the y-axis each correspond to an increment of the actual average rate. The window size must generally be quite large before the nominal average rate needed for the window approaches even twice the actual average rate.

The nominal average rate does not necessarily decrease monotonically as the window size increases. Figure 5 shows an example in which the nominal average rate increases

as the window size increases from 5 to 6. Therefore, using a larger window size may actually decrease the transmission efficiency.

4.2 Leaky bucket

The leaky bucket can be considered as an imaginary FIFO buffer of size N_{max} with constant drain rate $\bar{R}$ bits per frame period. Since R_i bits arrive in frame period i, the instantaneous bucket fullness is

$$N_i = \max\{0, N_{i-1} + R_i - \bar{R}\}. \tag{2}$$

To avoid violation, we choose N_{max} such that the bucket never overflows: $N_i \leq N_{max} \ \forall i$.

The result of this is also shown in Figures 2–4, again, assuming $R_i = E_i$. The size of the bucket is $N_{max}/\bar{R}$, which is the number of frame periods it takes to empty a full bucket. Here, we plot the drain rate $\bar{R}$ as a function of one plus the size, to provide a comparison to the sliding window. The drain rate, which is the nominal average rate, decreases monotonically as the bucket size increases.

4.3 Comparison

For a size of one, both traffic descriptors are identical. The nominal average rate is the peak rate of the sequence. The sliding window has a duration of one frame, and there is no leaky bucket. However, for larger sizes, it is difficult to compare these traffic descriptors because the size parameters are not directly equivalent. Both TD's use an integrator, but the sliding window has a finite-memory integrator, while the leaky bucket could have an infinite-memory integrator if the bucket never empties.

To compare these two TD's for larger size parameters, we examine the worst-case traffic that could be described by these TD because the network will use this for call admission. The worst-case traffic alternates between the peak allowed by the TD and zero, with an average rate equal to the TD's nominal average rate. (For both TD, the peak rate is the size times the nominal average rate.) Unfortunately, the peak increases much more quickly than the average rate decreases, so the worst-case traffic deteriorates quickly. Therefore, there may be little advantage to using a size greater than 1.

However, if we separately negotiate a peak rate and use the leaky bucket or sliding window to describe the average rate, then we obtain a more accurate traffic description. (The peak rate may also be constrained by the physical connection to the ATM network.) Again, the worst-case traffic alternates between the peak and zero. If both TD's have the same peak and average, then we can compare them using the maximum allowable duration of the peak, which depends directly on the size parameter. As the duration at the peak rate increases, the network queue lengths increase and packet losses become more probable.

In general, the maximum duration of the peak for the sliding window is longer than that of the leaky bucket. For example, for sequence C, we would declare the peak to be 78004 bits per frame. For a nominal average rate of three times the actual average ($\bar{R}$ = 35112 bits per frame), the leaky bucket size would be 2 frames and the sliding window size would be 8 frames. Then, the worst case traffic for the leaky bucket would have a rate of 78004 for 1.6 frames to fill the bucket and have zero rate for 2 frames to empty the bucket. The worst case traffic for the sliding window would have a rate of 78004 for 3.6 frames and have zero rate for 4.4 frames to produce an average of 35112 bits per frame over 8 frames. Because the duration of the peak is shorter for the leaky bucket, the leaky bucket is a better choice. A call described by the leaky bucket is more likely to be accepted into the network, which will, therefore, be utilized more efficiently.

5 Buffering

Next, we examine the ability of buffering in the video system to reduce the necessary TD parameters. We present bounds on the cumulative bit-rate output to the network based on buffering arguments, and rationalize a particular choice of smoothed rate. Then we indicate the reduction in the TD parameters.

We assume time is discretized at the frame level; although smaller intervals could be used if desired. The encoder outputs bits into the encoder buffer, E_i bits in frame period i. R_i bits are transmitted onto the channel during frame period i, and we assume these are received by the decoder buffer after some transmission delay. After waiting L frames, the decoder begins to decode frame i. (We treat L as an integer here, although it need not be.) The instantaneous encoder buffer fullness after coding frame i is B_i^e. Similarly, the instantaneous decoder buffer fullness after decoding frame i is B_i^d. (This implies the encoder and decoder buffers are using different time indexes.) Both the encoder and decoder buffer fullnesses must be constrained to prevent overflow and underflow. The encoder and decoder buffer sizes are B_{max}^e and B_{max}^d.

In AVC-56 [1], we presented bounds on the transmitted bit-rate such that neither the encoder nor decoder buffers overflow or underflow. We rewrite these here as bounds on the cumulative rate transmitted across the channel.

$$\sum_{j=1}^{i} E_j - B_{max}^e \leq \sum_{j=1}^{i} R_j \leq \sum_{j=1}^{i} E_j \tag{3}$$

$$\sum_{j=1}^{i-L} E_j \leq \sum_{j=1}^{i} R_j \leq \sum_{j=1}^{i-L} E_j + B_{max}^d, \tag{4}$$

when $i > L$. The left side of equation (3) prevents encoder buffer overflow, while the right side prevents encoder buffer underflow. Similarly, the left side of equation (4) prevents decoder buffer underflow, and the right side prevents decoder buffer overflow. Upper and

lower bounds on the cumulative transmitted rate can be found by taking the maximum of the left sides, and the minimum of the right sides.

$$D_i \leq \sum_{j=1}^{i} R_j \leq U_i, \tag{5}$$

where

$$D_i = \max\{\sum_{j=1}^{i} E_j - B^e_{max}, \sum_{j=1}^{i-L} E_j\} \tag{6}$$

and

$$U_i = \min\{\sum_{j=1}^{i} E_j, \sum_{j=1}^{i-L} E_j + B^d_{max}\}. \tag{7}$$

Any sequence of rates $\{R_i\}$ that satisfy equation (5) is valid from a buffering standpoint. Choosing one R_i may affect the range of possible choices for other R_j.

For a given video sequence, delay, and buffer sizes, the easiest way to see the effect of these bounds visually is to plot $U_i - i\tilde{R}$, and $D_i - i\tilde{R}$ as a function of the time index i, where $\tilde{R}$ is some convenient choice of rate. An example is shown in Figure 6 for the first 200 frames of sequence A, where $\tilde{R}$ is chosen to be the average rate of the 200 frame subsequence. The bounds are shown as dashed lines. The delay is $L = 3$ frames, and the buffer sizes are large enough that the constraints on the rate are imposed solely by the delay. The actual rate is indicated by the slope of the path; a positive slope corresponds to a rate greater than $\tilde{R}$, while a negative slope corresponds to a rate less than $\tilde{R}$. A zero rate corresponds to a slope of $-\tilde{R}$.

We note that, for this sequence, it is impossible to transmit at a constant rate without overflowing or underflowing one of the buffers, since no straight line passes through these bounds. However, paths through these bounds that correspond to variable rate transmission do exist. An example is shown by the solid line.

Now, we would like to determine the extent to which buffering can smooth the transmitted data to reduce the traffic descriptor parameters. If smoother data is transmitted onto the network, the network should perform better.

We argue in the following that a good choice of cumulative rate, in order to smooth the instantaneous rate transmitted onto the network, is given by the path through these bounds that has the shortest length. This provides a solution that is not causal, so it can not be implemented. However, it does use the buffers to their maximal extent, and as such, it will provide a measure of the extent to which buffering can reduce the size of the TD parameters. The shortest path is independent of the choice of $\tilde{R}$, and it can be found by applying the algorithm described in [4].

Clearly, the path with the shortest length will minimize the maximum rate, since it will have the smallest possible slope. In addition, the shortest path provides a rate that is locally optimal in the following sense. Any perturbation that decreases the maximum

slope in some local region will increase the maximum slope in a larger region that contains the local region.

The shortest path does not necessarily minimize the average rate for a given sliding window size, especially for large windows. However, for the sequences shown, the shortest path does minimize the average rate for windows up to 30 frames long. In addition, this one path minimizes the leaky bucket size for all drain rates. Visually, the bit-rate resulting from the shortest path is quite smooth, as shown in Figure 7. The actual coded bit-rate is quite noisy, while the smoothed rate is constant for large time intervals.

The presence of buffering in the video system significantly reduces the peak rate. In fact, for all these sequences, the window size for the buffered data is constrained not by the intraframe peaks, but by the bit-rate during high activity intervals.

In addition, the presence of buffering can significantly reduce the nominal average rate for a given sliding window size, as seen in Figures 2–4. This is particularly true for larger nominal average rates and smaller window sizes. However, the presence of buffering reduces the leaky bucket parameters significantly only for very small bucket sizes.

6 Conclusion

We have presented the traffic descriptor parameters for video teleconferencing that are necessary to avoid violation at the UPC. Using a peak rate constraint in addition to a leaky bucket or sliding window improves the traffic description to the network. Furthermore, the leaky bucket characterizes the video traffic to the network more accurately than the sliding window does. Therefore, the network can admit more calls so it can utilize the total capacity more efficiently. We also showed that buffering can significantly reduce the necessary sliding window parameters, especially for small window sizes. However, buffering reduces the leaky bucket parameters only slightly.

References

[1] AVC-56, "Constraints on Variable Bit-Rate Video for ATM Networks", AT&T Bell Labs, Paris, 23–24 May 1991.

[2] AVC-128, "Comparison between sliding window and leaky bucket as a UPC mechanism", Japan, Japan, 6 November 1991.

[3] AVC-49, "Some observations on variable bit rate coded video signals", Japan, Paris, 23–24 May 1991.

[4] D.T. Lee and F. P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers", *Networks*, volume 14, pages 393–410, 1984.

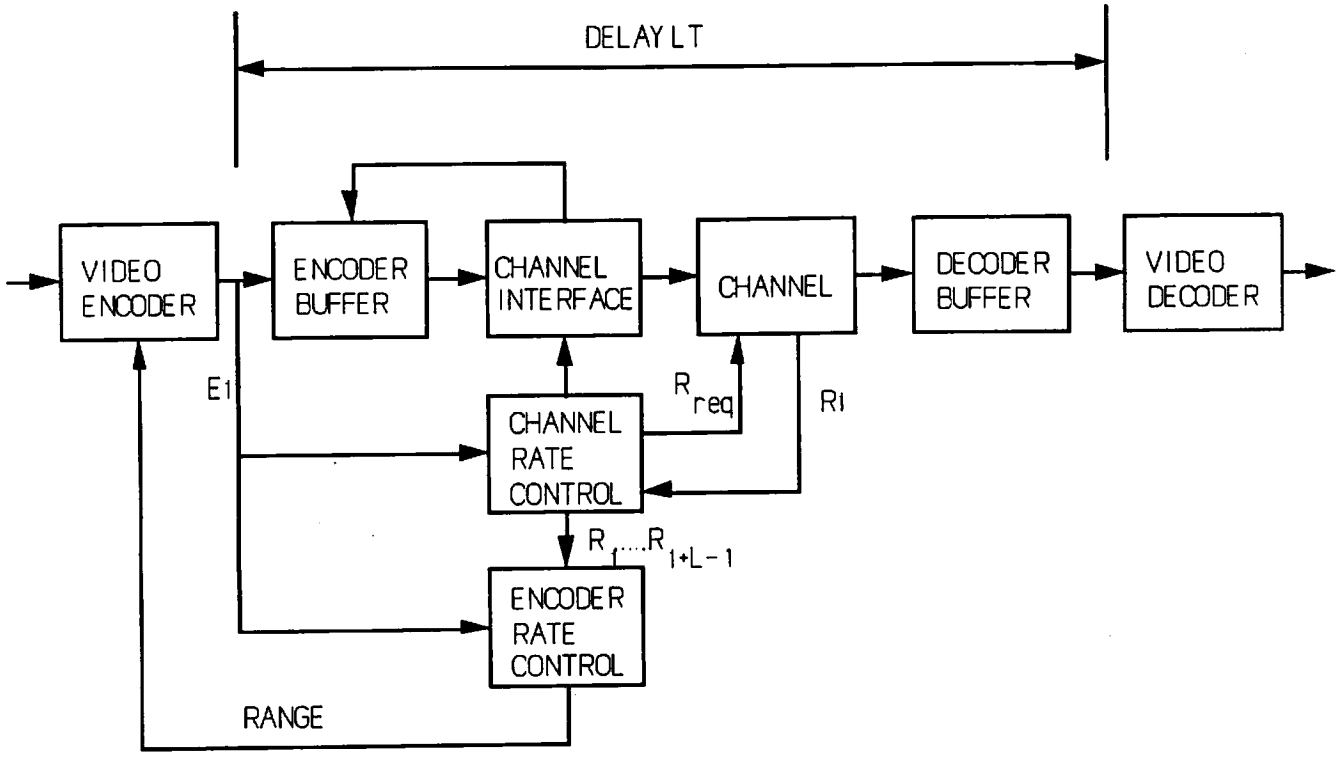

Figure 1: System

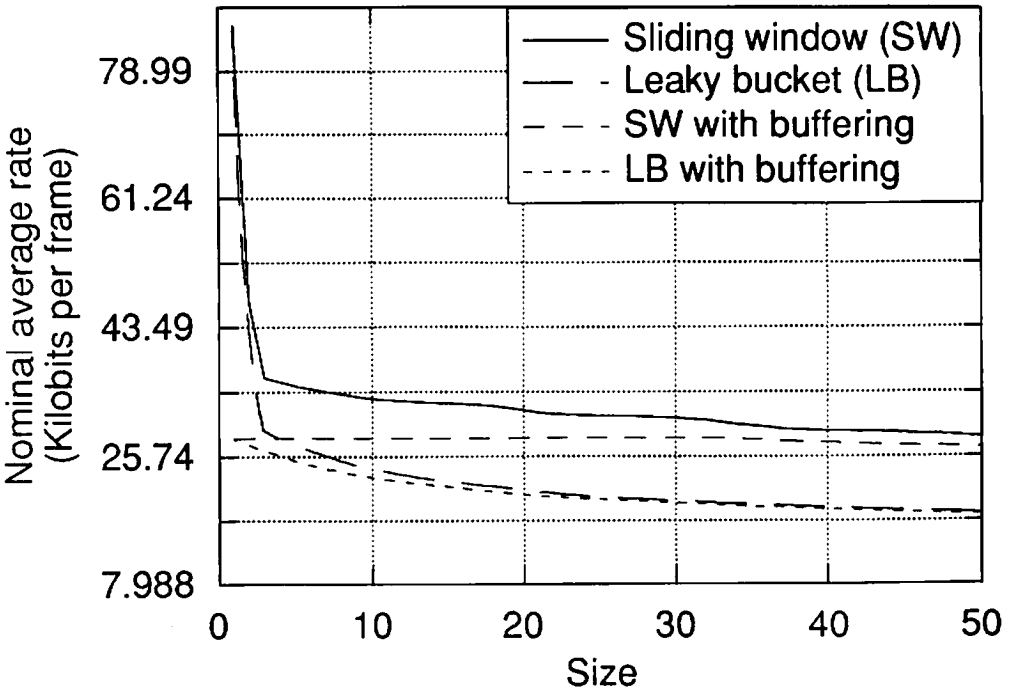

Figure 2: TD parameters, sequence A

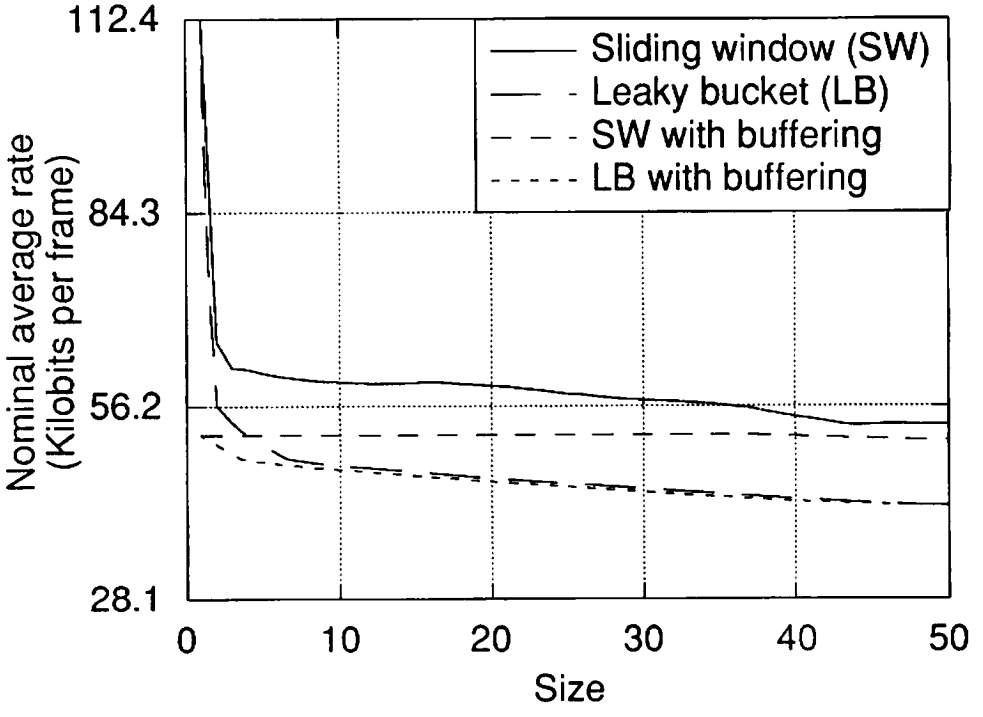

Figure 3: TD parameters, sequence B

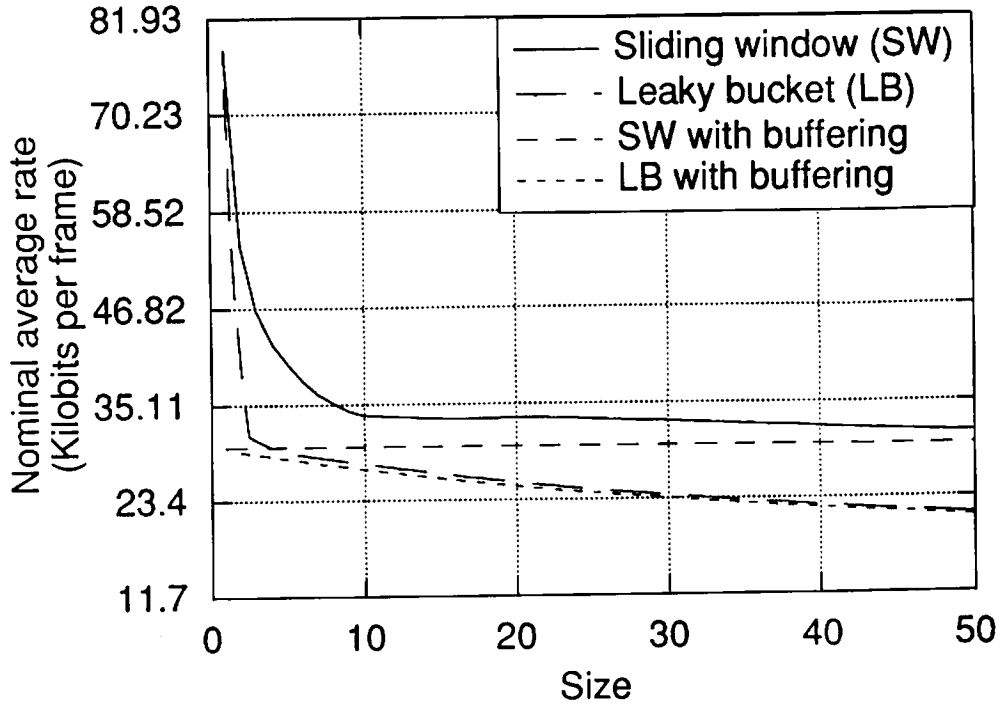

Figure 4: TD parameters, sequence C

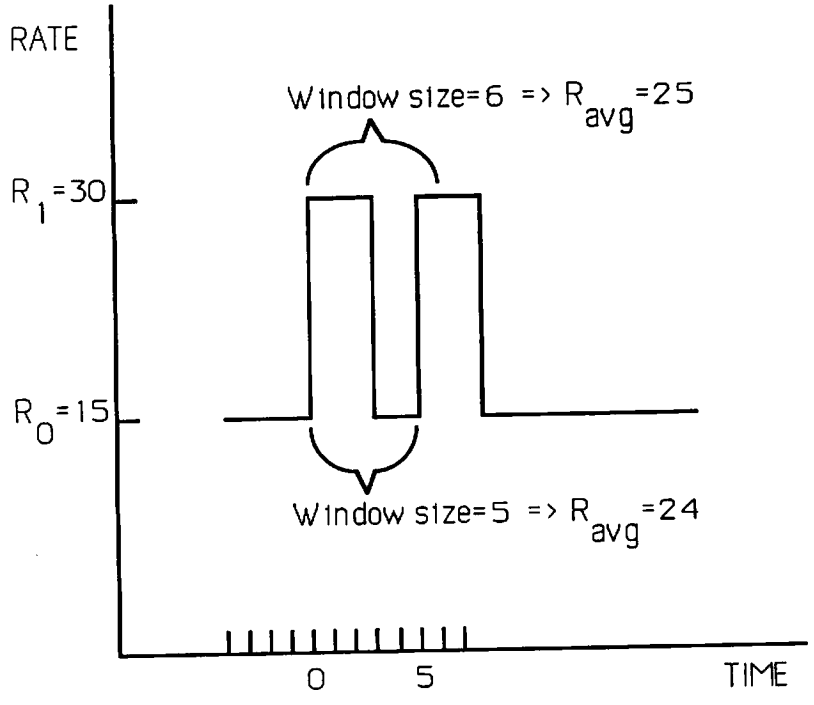

Figure 5: Example of Sliding Window

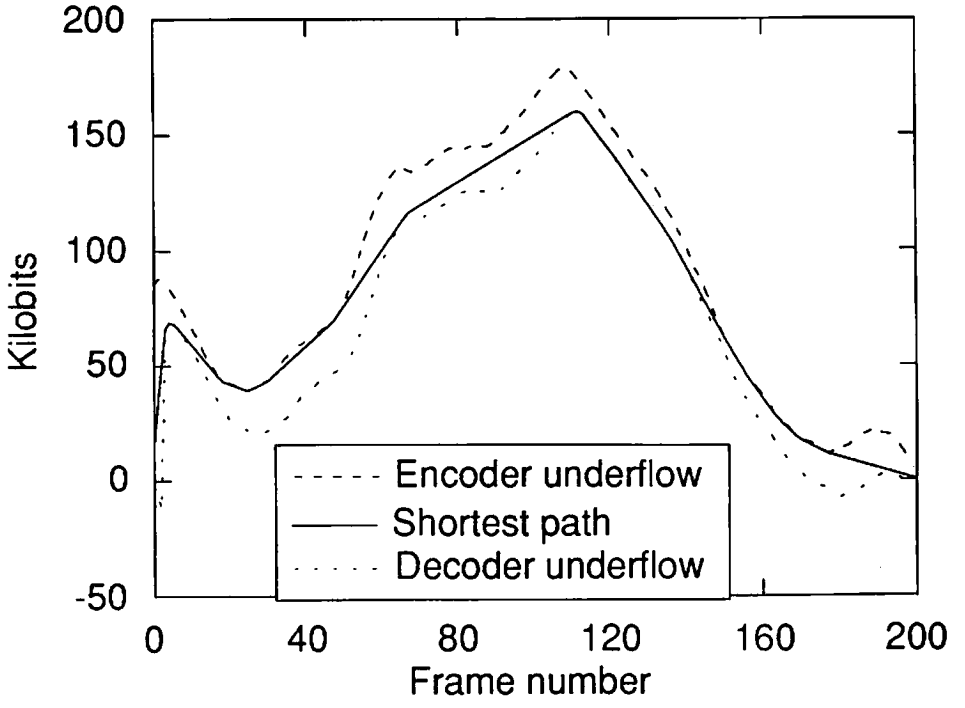

Figure 6: Bounds on cumulative rate imposed by buffering

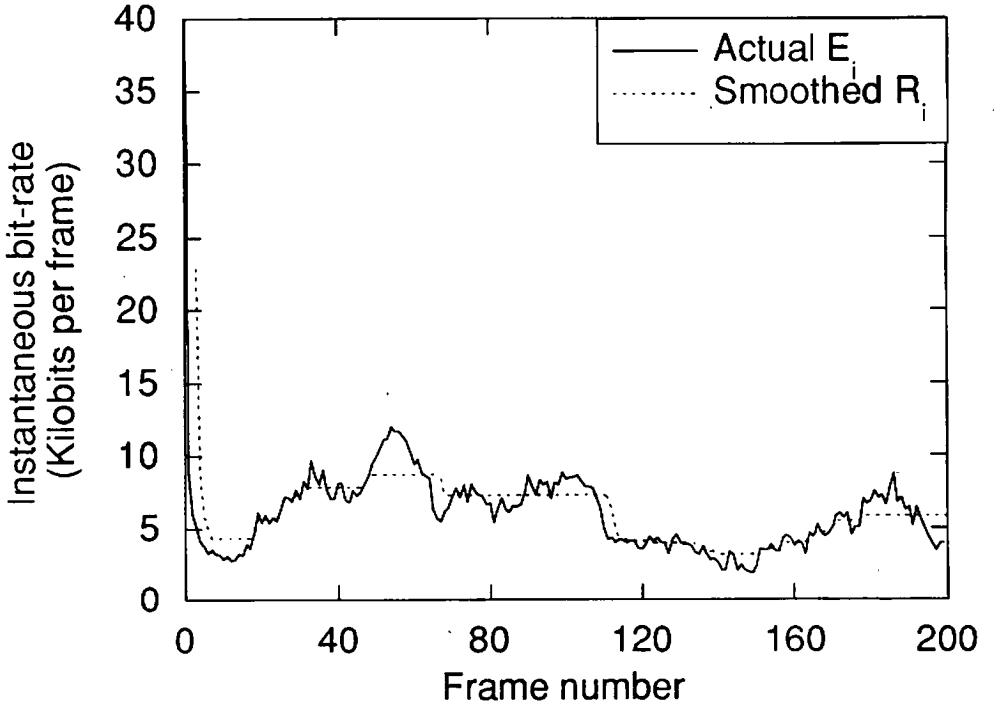

Figure 7: Actual and smoothed rate for sequence A