A FLEXIBLE DECODER BUFFER MODEL FOR JVT VIDEO CODING

Jordi Ribas-Corbera, Philip A. Chou, and Shankar Regunathan
Microsoft Corporation
One Microsoft way, Redmond, WA 98052, USA
{jordir, pachou, shanre}@microsoft.com

ABSTRACT

Video coding standards require that a compliant bit stream be decodable by a hypothetical decoder that is conceptually connected to the output of an encoder and consists of a decoder buffer, a decoder, and a display unit. The encoder must create a bit stream so that the hypothetical decoder buffer does not overflow or underflow. Previous decoder models assume that a given bit stream will be transmitted through a channel of a given constant bit rate and will be decoded (after a given buffering delay) by a device of some given buffer size. Therefore, these models are quite rigid and do not address the requirements of many of today's important video applications such as broadcasting live video or streaming pre-encoded video on demand over network paths with various peak bit rates to devices with various buffer sizes. In this contribution, we present a new hypothetical reference decoder for JVT that is more general and flexible than those defined in prior standards and provides significant additional benefits.

1. INTRODUCTION

In previous hypothetical reference decoders [1, 2], the video bit stream is received at a given constant bit rate (usually the average rate in bits/sec of the stream) and is stored into the decoder buffer until the buffer fullness reaches a desired level. This level is denoted the initial decoder buffer fullness and is directly proportional to the transmission or start-up (buffer) delay. When the initial buffer fullness is reached, the decoder instantaneously removes the bits for the first video frame of the sequence, decodes the bits, and displays the frame. The bits for the following frames are also removed, decoded, and displayed instantaneously at subsequent time intervals.

The hypothetical reference decoder operates at a fixed bit rate, buffer size, and initial delay. However, in many of today's video applications (e.g., video streaming through the Internet) the peak transmission bit rate varies according to the network path and also fluctuates in time according to network conditions [3, chapters 1-2]. In addition, the video bit streams are delivered to a variety of devices with different buffer capabilities and are created for scenarios with different delay requirements [4, chapter 8]. As a result, these applications require a more flexible hypothetical reference decoder that can decode a bit stream at different peak transmission bit rates, and with different buffer sizes and start-up delays.

We propose a new hypothetical reference decoder that operates according to N sets of transmission rate and buffer size parameters for a given bit stream. Each set characterizes what is known as a leaky bucket model [5, 6] and contains three values (R, B, F), where R is the peak transmission bit rate, B is the buffer size, and F is the initial decoder buffer fullness. (F/R is the start-up or initial buffer delay.) An encoder can create a video bit stream that is contained by some desired N leaky buckets, or can simply compute the N sets of parameters after the bit stream has been generated. Our new hypothetical reference decoder intelligently interpolates among the leaky bucket parameters and can operate at any desired peak transmission bit rate, buffer size or delay. To be more concrete, given a desired peak transmission bit rate R', our reference decoder will select the smallest buffer size and delay (according to the available leaky bucket data) that will be able to decode the bit stream without suffering from buffer underflow or overflow. Conversely, for a given buffer size B', the hypothetical decoder will select and operate at the minimum required peak transmission bit rate.

There are multiple benefits of this generalized hypothetical reference decoder. For example, a content provider can create a bit stream once, and a server can deliver it to multiple devices of different capabilities, using a variety of channels having different peak transmission bit rates. Or a server and a terminal can negotiate the best leaky bucket for the given networking conditions, e.g., the one that will produce the lowest start-up (buffer) delay, or the one that will require the lowest peak transmission bit rate for the given buffer size of the device. In Section 4 we quantify these benefits for the standard MPEG test sequences encoded with the test model TML4 (version 4.3) of JVT [7]. We find that in realistic scenarios, the buffer size and the delay for some terminals can be reduced by an order of magnitude, or the peak transmission bit rate can be reduced by a factor of four, or the SNR can increase perhaps by several dB without increasing the average bit rate, except by a few bytes in the stream header.

2. PREVIOUS WORK

We first define a leaky bucket model, since it is the basis of all the hypothetical reference decoders that we will discuss later.

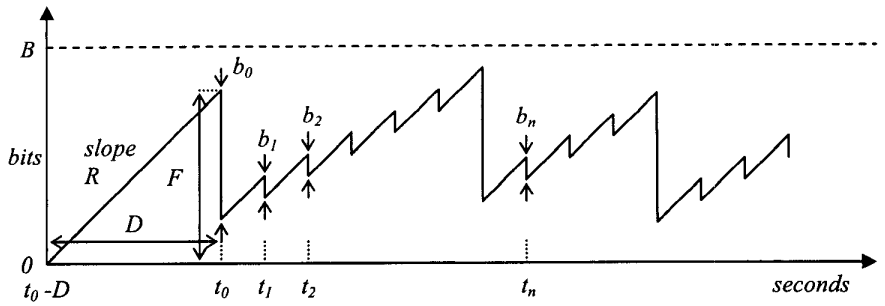

Figure 1. The plot illustrates the decoder buffer fullness when decoding a generic video bit stream that is contained in a leaky bucket having parameters (R, B, F). R is the peak incoming bit rate, B is the buffer size, and F is the initial decoder buffer fullness. $D = F/R$ is the start-up delay. The number of bits for the ith frame is b_i. The coded video frames are removed from the buffer as shown by the drops in buffer fullness.

2.1 Leaky bucket model

A leaky bucket is a model for the state of an encoder or decoder buffer as a function of time. We focus on the decoder buffer without loss of generality, because one can show that the fullness of the encoder and decoder buffer are complements of each other (see [8]). A leaky bucket model is characterized by three parameters (R, B, F), where:

- R is the peak transmission bit rate (in bits per second) at which bits enter the decoder buffer. In constant bit rate scenarios, R is often the channel bit rate and the average bit rate of the video clip.
- B is the size of the bucket or decoder buffer (in bits) which smoothes the video bit rate fluctuations. This buffer size cannot be larger than the physical buffer of the decoding device.
- F is the initial decoder buffer fullness (also in bits) before the decoder starts removing bits from the buffer. F and R determine the initial or start-up delay D, where $D = F/R$ seconds.

In a leaky bucket model, the bits enter the buffer at rate R until the level of fullness is F, and then b_0 bits for the first frame are instantaneously removed. The bits keep entering the buffer at rate R and the decoder removes $b_1, b_2, \ldots, b_{n-1}$ bits for the following frames at some given time instants. Figure 1 illustrates the decoder buffer fullness along time for a bit stream that is contained in a leaky bucket having parameters (R, B, F).

Let B_i be the decoder buffer fullness immediately before removing b_i bits at time t_i. A generic leaky bucket model operates according to the following equations:

$$B_0 = F$$
$$B_{i+1} = \min(B, B_i - b_i + R(t_{i+1} - t_i)), \quad i = 0, 1, 2, \ldots \quad (1)$$

Typically, $t_{i+1} - t_i = 1/M$ seconds, where M is the frame rate (in frames/sec) for the bit stream.

A leaky bucket model with parameters (R, B, F) *contains* a bit stream if there is no underflow of the decoder buffer.

We make the following observations:

- A given video stream can be contained in many leaky buckets. For example, if a video stream is contained in a leaky bucket with parameters (R, B, F), it will also be contained in a leaky bucket with a larger buffer (R, B', F), $B' > B$, or in a leaky bucket with a higher peak transmission bit rate (R', B, F), $R' > R$, *or in a leaky bucket with larger start-up delay* (R, B, F') $F' > F$.
- For any bit rate R', we can always find a buffer size and start-up delay that will contain a time-limited video bit stream. In the worst case, as R' approaches 0, the buffer size and initial buffer fullness will need to be as large as the bit stream itself.

Thus, for any value of the peak bit rate R, we can find the minimum buffer size B_{min} and the minimum initial buffer fullness F_{min} that will contain the bit stream. This can be done by iterating equation (1), as illustrated by the matlab code in Appendix B of [9]. Surprisingly, both minima can be achieved simultaneously, as we prove in Appendix A of [9], even though in general, the buffer size required to contain the bit stream may increase as the initial buffer fullness decreases. By computing B_{min} for each R, we can plot a curve of R-B values such as the one in Figure 2.

A key observation is that the curve of (R_{min}, B_{min}) pairs for any bit stream (such as the one in Figure 2) is piecewise linear and convex. A rigorous proof of the piecewise linear and convex properties of the curve of (R_{min}, B_{min}) pairs is provided in Appendix A of [9].

Because of the convexity, if N points of the curve are provided, the decoder can linearly interpolate the values to arrive at some points $(R_{interp}, B_{interp}, F_{interp})$ that are slightly but safely larger than $(R_{min}, B_{min}, F_{min})$. In this way, as we quantify in Section 4, one is able to safely reduce the buffer size, and the delay, by an order of magnitude, relative to a single leaky bucket containing the bit stream at its average rate. Alternatively, for the same delay, one is able to reduce the peak transmission rate by a factor of four, or possibly even improve the SNR by several dB.

3. A GENERALIZED HYPOTHETICAL REFERENCE DECODER

We propose a generalized hypothetical reference decoder (GHRD) that can operate given the information of N leaky bucket models,

$$(R_1, B_1, F_1), (R_2, B_2, F_2), \ldots, (R_N, B_N, F_N), \quad (5)$$

each of which contains the bit stream. Without loss of generality, let us assume that these leaky buckets are ordered from smallest to largest bit rate, i.e., $R_i < R_{i+1}$. Let us also assume that the encoder computes these leaky bucket models correctly and hence $B_i > B_{i+1}$.

The desired value of N is selected by the encoder. (If N=1, the GHRD is essentially equivalent to MPEG's VBV). The encoder can choose to: (a) pre-select the leaky bucket values and use rate control to ensure that bit stream meets the leaky bucket constraints, (b) encode the bit stream and then compute a set of leaky buckets containing the bit stream, or (c) do both. The number of leaky buckets N and the leaky bucket parameters are inserted into the bit stream. Thus, the decoder can determine the appropriate leaky bucket for its conditions. The leaky bucket models in (5) as well as all the linearly interpolated are available for use. Figure 3 illustrates a set of N leaky bucket models and their interpolated (R, B) values.

The interpolated buffer size B between points k and k+1 follow the straight line:

$$B = \frac{R_{k+1} - R}{R_{k+1} - R_k} B_k + \frac{R - R_k}{R_{k+1} - R_k} B_{k+1}, \quad R_k < R < R_{k+1}. \tag{6}$$

Likewise, the initial decoder buffer fullness F can be linearly interpolated:

$$F = \frac{R_{k+1} - R}{R_{k+1} - R_k} F_k + \frac{R - R_k}{R_{k+1} - R_k} F_{k+1}, \quad R_k < R < R_{k+1}. \tag{7}$$

The resulting leaky bucket with parameters (R, B, F) is guaranteed to contain the bit stream, because the minimum buffer size B_{min} is convex in both R and F, that is, the minimum buffer size B_{min} corresponding to any convex combination $(R,F) = a(R_k,F_k) + (1\text{-}a)(R_{k+1},F_{k+1})$, $0<a<1$, is less than or equal to $B = aB_k + (1\text{-}a)B_{k+1}$.

If R is larger than R_N, the leaky bucket (R, B_N, F_N) will also contain the bit stream, and hence B_N and F_N are the buffer size and initial decoder buffer fullness recommended when $R \geq R_N$.

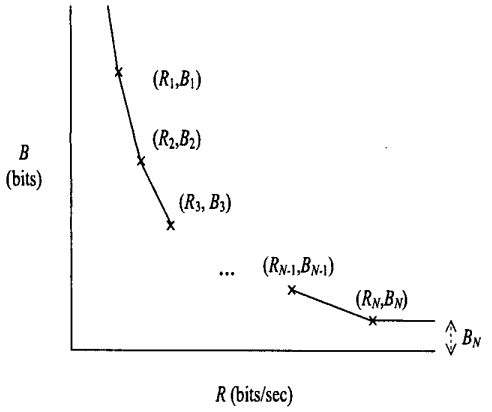

Figure 3. Example of (R, B) values available for the generalized hypothetical reference decoder (GHRD), all of which are guaranteed to contain the bit stream.

4. RESULTS: EVALUATION OF THE GHRD

To evaluate the benefits of the generalized hypothetical reference decoder (GHRD), we encoded a 130-sec video clip (which contained all the MPEG clips combined) with the test model TML4 (version 4.3) of JVT [7] using a fixed quantizer and typical options (see [9] for details). We set F=B and used the formulas in (1) to provide the (R_{min}, B_{min}) plot in Figure 4.

The bit stream in Figure 4 was produced with QP=16 and yielded an average bit rate of 797 Kbps. As shown in the figure, at a constant transmission bit rate of 797 Kbps, the decoder needs a buffer size of about 18,000 Kbits. With an initial decoder buffer fullness equal to 18,000 Kbits, the start-up delay is about 22.5 seconds. Thus, this VBR encoding (produced with no rate control) shifts bits by up to 22.5 seconds in order to achieve the essentially best possible quality for its overall encoded length.

The figure also shows that at a peak transmission bit rate of 2,500 Kbps (e.g., the video bit rate portion of a 2x CD), the decoder needs a buffer size of only 2,272 Kbits, sufficiently small for a consumer hardware device. With an initial buffer fullness equal to 2,272 Kbits, the start-up delay is only about 0.9 seconds.

Thus for this encoding two leaky bucket models might typically be useful:

1. (R=797 Kbps, B=18,000 Kbits, F=18,000 Kbits). This leaky bucket permits transmission of the video over a constant bit rate channel, with a delay of about 22.5 seconds. This delay is probably acceptable for internet streaming of movies.
2. (R=2,500 Kbps, B=2,272 Kbits, F=2,272 Kbits). This leaky bucket permits transmission of the video over a shared network with peak rate 2,500 Kbps, or permits local playback from a 2x CD, with a delay of about 0.9 seconds. This sub-second delay is acceptable for random access playback with VCR-like functionality.

If only the first leaky bucket is specified in the bit stream, but not the second, then even when playing back over a channel with peak bit rate 2,500 Kbps, the decoder would use a buffer of size 18,000 Kbits and thus the delay would be F/R = 18,000 Kbits / 2,500 Kbps = 7.2 seconds. This is too large for random access playback with VCR-like functionality. However, if the second leaky bucket is specified as well, then at peak bit rate 2,500 Kbps the buffer size drops to 2,272 Kbits and the delay drops to 0.9 seconds, as we have seen.

On the other hand, if only the second leaky bucket is specified, but not the first, then at a constant transmission rate of 797 Kbps, even a smart decoder would be forced to use a buffer that is far larger than necessary, to ensure that the buffer will not overflow: $B' = B + (R - R')T$ = 2,272 Kbits + (2,500 Kbps – 797 Kbps) x 130 seconds = 223,662 Kbits. This corresponds to an initial delay of 281 seconds, or nearly 5 minutes, over twice the length of the original clip, which is typically far from acceptable. However, if the first leaky bucket is specified as well, then at rate 797 Kbps the buffer size drops to 18,000 Kbits and the delay drops to 22.5 seconds, as we have seen.

Moreover, if both leaky buckets are specified, then the decoder can linearly interpolate between them (using (6) and (7)), for any bit rate R between 797 Kbps and 2,500 Kbps, thereby achieving near-minimal buffer size and delay at that rate. Extrapolation is also more efficient both below 797 Kbps and above 2,500 Kbps, compared to extrapolation with only a single leaky bucket anywhere between 797 Kbps and 2,500 Kbps, inclusive.

As the above example shows, even just two leaky buckets can provide an order of magnitude reduction in buffer size, and an order of magnitude reduction in delay at a given peak transmission bit rate.

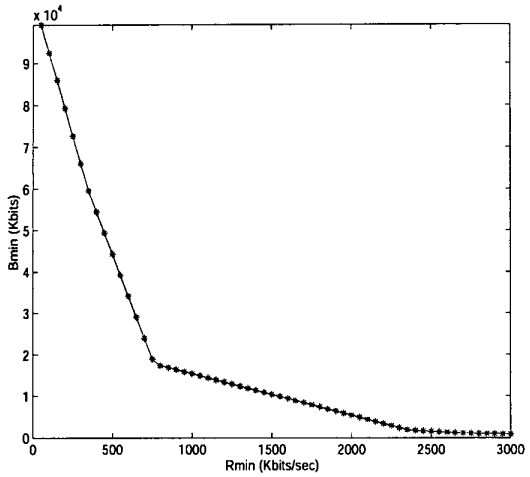

Figure 4. Plot of leaky bucket parameters (R, B) for an JVT compressed video clip with QP=16. The points labeled with "*" correspond to the minimum buffer size B_{min} needed to contain a bit stream with the associated rate R_{min}. These points scan bit rates from 50 Kbps to 3 Mbps in increments of 50 Kbps. The other points between the "*" are linearly interpolated. Observe that only a subset of those points would characterize this curve fairly well.

Conversely, it is also possible to reduce the peak transmission bit rate for a given decoder buffer size. For example, we know from the figure that if the decoder has a fixed buffer of size 18,000 Kbits, then the peak transmission bit rate for the encoding can be as low as 797 Kbps. However, if only the second leaky bucket is specified, but not the first, then the decoder can reduce the bit rate to no less than $R' = R - (B' - B)/T$ = 2,500 Kbps – (18,000 Kbits – 2,272 Kbits) / 130 seconds = 2,379 Kbps. In this case, compared to using a single leaky bucket, using just two leaky buckets reduces the peak transmission rate by a factor of four, for the same decoder buffer size.

Having multiple leaky buckets can also improve the quality of the reconstructed video, *at the same average encoding rate*, in the following sense. Suppose both leaky buckets are available for the encoding described above. Then, it is possible to play back the encoding with a delay of 22.5 seconds if the peak transmission rate is 797 Kbps, and with a delay of 0.9 seconds if the peak transmission rate is 2,500 Kbps. However, if the second leaky bucket is unavailable, then the delay increases from 0.9 to 7.2 seconds at 2,500 Kbps. To reduce the delay back to 0.9 seconds without the benefit of the second leaky bucket, it would suffice to re-encode the clip *with rate control* by reducing the buffer size (of the first leaky bucket) from 18,000 Kbits to (0.9 seconds) x (2,500 Kbps) = 2,250 Kbits. This would ensure that the delay is only 0.9 seconds if the peak transmission rate is 2,500 Kbps. As a side effect, the delay at 797 Kbps would also decrease, from 22.5 to 2.8 seconds, and the quality (SNR) would also decrease, probably by several dB, especially for a clip with a large dynamic range. (Unfortunately we cannot yet evaluate this decrease in SNR with objective tests, because there is no rate control as of yet in the test model of JVT.) In this way, specifying a second leaky bucket can increase the SNR by possibly several dB, with no change in the average bit rate (except for 64 additional bits per clip to specify the second leaky bucket, using the proposed syntax). This increase in SNR will be visible on playback for every peak transmission rate.

REFERENCES

[1] *Video Coding for Low Bit Rate Communication, ITU-T recommendation H.263*, Annex B "Hypothetical Reference Decoder", Sept. 1997.

[2] *ISO/IEC 138180-2, Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video (MPEG-2/H.262)*, Annex C "Video Buffering Verifier", 2nd Edition, 2000.

[3] J. Crowcroft, M. Handley, and I. Wakeman, *Internetworking Multimedia*, Morgan Kaufmann Publishers, 1999.

[4] J. Keyes, *Webcasting*, Mc-Graw Hill, 1997.

[5] C-Y. Hsu, A. Ortega, and A.R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE Journal of Selected Areas in Communications*, pp. 1016-1028, Vol. 15, No. 6, Aug. 1997

[6] A.R. Reibman and B.G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 361-372, Vol. 2, No. 4, Dec. 1992.

[7] JVT Test Model Long Term number 4 (TML-4), *ITU-T SG16/Q15, VCEG*, G. Bjontegaard, Ed., Osaka, Japan, May 2000, doc Q15-J-72

[8] H.-M. Hang and J.J Chen, "Source model for transform video coder and its application – Part II: Variable frame rate coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 299-311, April 1997.

[9] J. Ribas-Corbera, P.A. Chou and S.L. Regunathan, "A generalized hypothetical reference decoder for H.26L," submitted to IEEE *Trans. On Circuits and Systems for Video Techology.*