HRD Conformance for Real-time H.264 Video Encoding

Jennifer L. H. Webb

DSPS Solutions R&D Center, Texas Instruments, Incorporated, MS 6849
12500 TI Boulevard, Dallas, Texas 75243
webb@ti.com

ABSTRACT

The H.264 hypothetical reference decoder (HRD) ensures interoperability and smooth playback of video. Because the encoded bit rate may not match the channel rate, buffering and timing information must be specified for the decoder. These parameters can be obtained by analyzing a prerecorded bitstream, but if the video is encoded and transmitted in real-time, these parameters effectively constrain the rate control. We discuss how to set these parameters for real-time encoding, without having the entire bitstream available, and also how VProve can be used to verify HRD conformance.

Index Terms— H.264, HRD, video, SEI

1. INTRODUCTION

The H.264 video standard [1] has a hypothetical reference decoder (HRD) model, and includes special syntax to ensure that the video can be streamed smoothly, by ensuring that the encoder and decoder buffer levels remain complementary. Supplemental Enhancement Information (SEI) messages can provide the parameters necessary for streaming. In general, buffering is used to compensate for variations in the bit rate. Aside from preventing overflow and underflow in the encoder buffer, which is outside the scope of this report, the encoder should signal to the decoder sufficient information about when to remove bits from the coded picture buffer (CPB) for correct output timing. Without this information, a streaming decoder will pause the playback whenever its buffer underflows.

An excellent description of the H.264 HRD is given in [1]. Unlike MPEG-4 syntax, the H.264 HRD-related information is contained in network access layer (NAL) units that are separate from the NAL units containing slice data used to construct each frame. In this way, one can easily insert HRD information into a prerecorded bitstream. Information about the buffering and channel bit rate requirements are part of the video usability information (VUI) parameters. Also, for each random access point (instantaneous decoding refresh IDR) there should be a buffering period SEI message [1, D.1.1] to ensure the proper buffer fullness at start of decoding. And for each frame, there should be a picture timing SEI message [1, D.1.2]. To verify output timing conformance of a bitstream, these HRD-related parameters must be known.

The HRD parameters correspond to the channel bit rate, which is not necessarily the target bit rate used when encoding the bitstream. For a given prerecorded bitstream, the amount of buffering B required depends on the channel rate R, with more buffering required at low rates, and less buffering required at high rates. The H.264 syntax supports multiple (B,R) schedules. Given all of the frame sizes, the minimum B value can be determined for various R values. However, for the purposes of real-time encoding, we will assume that the encoder rate control targets a single channel rate, and generates HRD and SEI parameters without access to all of the frame sizes for the entire bitstream.

2. H.264 HRD REQUIREMENTS

The H.264 Annex C specifies, among other things, the behavior of a hypothetical decoder buffer model, and the timing information that should be provided. The HRD model assumes instantaneous encode and decode. Hence, the plot of cumulative bits generated looks like a stair step.

For a constant bit-rate channel, the encoder buffer and decoder buffer levels should be complementary, as illustrated in Fig 1 [2]. The "filled" areas represent the number of bits in the buffer, and the sloping lines show the bounds on the physical buffer. These lines represent the bits leaving the encoder buffer and entering the decoder buffer at the channel rate. We assume a constant-delay channel. Note that the physical buffer is not unique, for a given stair-step encoding sequence; the sequence can be contained with lines of different slope and spacing.

The buffer can be modeled as a leaky bucket with parameters (B, R, and F), where B is the buffer size, R is the rate of transmission, and F is initial buffer fullness [3]. If we did not assume instantaneous encode and decode, and the bitstream was produced and consumed at the exact channel rate, the buffer level would always be zero, a line with slope R. The buffering B allows the bitstream bit rate to vary from the channel rate. In Fig.1, the vertical spacing between the two limit lines is B, and the slope of the lines is R.

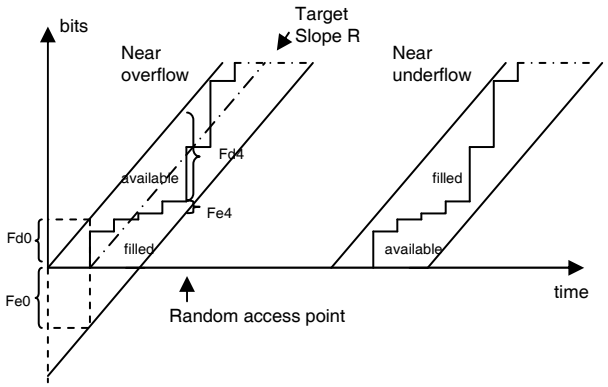

Figure 1. For constant delay and constant bit rate R, the encoder buffer and decoder buffer are complementary.

The positioning of the lines is determined by leaky bucket parameters F_d and F_e, where $F_d+F_e=B$ [2]. F_d denotes the decoder buffer level when the decoder begins decoding. F_e can be visualized as dummy bits in the encoder buffer when the first frame is added. The dummy bits are not actually transmitted, but serve to delay the transmission of the first frame. F_d allows a margin for the actual bit rate to exceed the target bit rate, while F_e allows a margin for the actual bit rate to lag behind the target bit rate.

Note that with random access, the decoder may begin decoding in the middle of the bitstream. Parameter F_d should be provided for every random access point, or IDR, so that the decoder buffer level will be complementary to the encoder's buffer level. In general terms, F_d is the difference between the actual encoder buffer level and the maximum level, while F_e is the difference between the actual buffer level and the minimum level. For a streaming video application, the leaky bucket parameters ensure that once the decoder begins decoding, it may remove frames at regular intervals without stalling due to underflow.

3. RATE CONTROL AND HRD PARAMETERS

For real-time recording of all-purpose video using rate control, we will assume a target bit rate R and a fixed buffer size B. In contrast, for a given prerecorded video sequence, it is possible to specify parameters for multiple (B,R) configurations, using H.264 syntax [2] with cpb_cnt_minus1 > 0. In the VUI parameters of the sequence parameter set, two kinds of HRD parameters may be specified: nal_hrd_parameters and vcl_hrd_parameters. We will assume these are the same, although it is possible to specify a tighter bound on buffer size and bit rate for the VCL NAL units. The VCL (video coding layer) HRD parameters are provided for a decoder model in which a parser places only slice data in the buffer, removing other NAL units, such as sequence parameter sets, picture parameter sets, and SEI messages. For real-time rate control, it is simpler to model a single buffer which contains not only VCL data, but also SPS, PPS, SEI, etc.

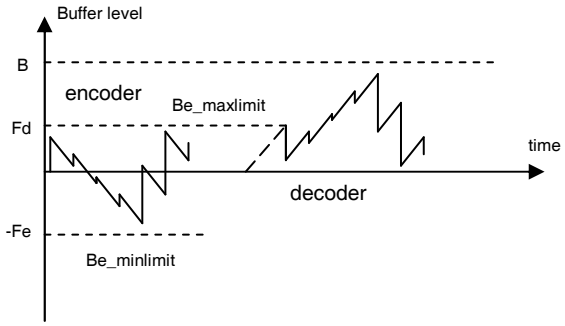

Figure 2. The encoder and decoder buffer levels above correspond to the example in Fig 1. After the initial buffer filling at the decoder, the plot for the decoder buffer level is the vertical flip of the encoder plot. The encoder buffer level must stay in the range [F_d, -F_e] to ensure that the decoder buffer level remains in the range [0, B], where $B=F_d+F_e$.

As part of the rate control, the encoder must compute the number of bits, B_e_level, being buffered. B_e_level is computed simply by adding in bits generated for each frame b_i, and subtracting out bits that would be transmitted for each frame.

From Fig. 2, noting that the decoder plot is vertically flipped from the encoder plot, we can see that keeping the encoder Be_level in the range [F_d, -F_e] will ensure that the decoder plot is in the range [0, B], where $B=F_d+F_e$. Mathematically, this can be shown by calculating the buffer levels at the encoder and decoder, both before(-) and after(+) adding or removing frame *k,* respectively. In the equations below, define $t_0 = 0$.

$$B_k^{d+} = F_d - \sum_{i=0}^{k} b_i + R \cdot t_k \tag{1}$$

$$B_k^{d-} = F_d - \sum_{i=0}^{k-1} b_i + R \cdot t_k \tag{2}$$

$$B_k^{e+} = \sum_{i=0}^{k} b_i - R \cdot t_k \tag{3}$$

$$B_k^{e-} = \sum_{i=0}^{k-1} b_i - R \cdot t_k \tag{4}$$

We can express the decoder buffer level in terms of the encoder Be_level, since they are complementary:

$$B_k^{d+} = F_d - \sum_{i=0}^{k} b_i + R \cdot t_k = F_d - B_k^{e+} \geq 0 \tag{5}$$

$$B_k^{d-} = F_d - \sum_{i=0}^{k-1} b_i + R \cdot t_k = F_d - B_k^{e-} \leq B_{vbv} \tag{6}$$

Then to avoid underflow and overflow at the decoder, Eq. (5) and (6), respectively, imply the encoder Be_level must satisfy

$$B_k^{e+} \le F_d, \quad (7)$$

$$B_k^{e-} \ge F_d - B = -F_e. \quad (8)$$

In Fig. 2, this is somewhat surprising, since a physical buffer level cannot be less than zero. However, if no bits are actually transmitted until the encoder buffer level reaches Fe, then the physical buffer level, after that time, will be Fe bits larger than the computed Be_levels above. Also, the result may seem counterintuitive, because it implies the encoder level is limited to Fd < B bits, where the upper limit on B is determined by the level_idc. However, whenever the encoder Be_level drops to –Fe, then at that time, it would be possible to generate a very large frame up to B bits in size.

The parameter Fe actually allows some margin for the rate control to lag behind the target bit rate. Without Fe>0, for any low-motion subsequence that does not use its full allocation of bits, those bits are forfeited, or used for generating stuffing bits. With Fe>0, the computed Be_level simply goes negative. Once the encoder buffer level is reduced to –Fe, the encoder must generate stuffing to avoid underflowing its physical buffer. In essence, Fe is the limit on the number of unused bits that may be allocated to future frames.

Note that Fd+Fe=B in Fig. 2. If Fd is large, there will be more delay before decoding begins, once the first bits are received. If Fd is small and Fe is large, there will be more delay before the encoder begins sending bits. The total delay depends on B, not the partitioning between Fd and Fe. The only way to reduce the total delay is to reduce the total B. Also note, if Fd is small, the encoded size of the first frame(s) is more restricted. For a given pre-recorded bitstream, we may choose Fd and Fe in a way that forms a tight bound on the bitstream stair-step [2]. The parameter Fe is not used by the decoder, only by a hypothetical stream scheduler (HSS) that ensures bits arrive at the decoder in time. For simplicity, the encoder buffer computations may assume transmission starts when the first frame is encoded, and impose a minimum Be_level of –Fe, as in Fig. 2.

The figures illustrate a constant frame rate, but for real-time encoding, some frames may be skipped. The bitstream should indicate not only the Fd value, but also the timing for the removal of each frame. These parameters are transmitted in buffering period SEI and picture timing SEI messages.

4. BUFFERING PERIOD SEI MESSAGES

The Buffering period SEI message occurs before each IDR, and provides the initial buffering requirement, initial_cpb_removal_delay, at random access points. Note that the decoder ignores subsequent buffering period SEI messages, once it begins decoding. The purpose of the initial buffering is to keep the encoder and decoder buffer levels complementary. Then the decoder buffer level can be inferred from the encoder buffer level, and the rate control at the encoder can prevent overflow or underflow at the decoder by controlling the buffer level at the encoder.

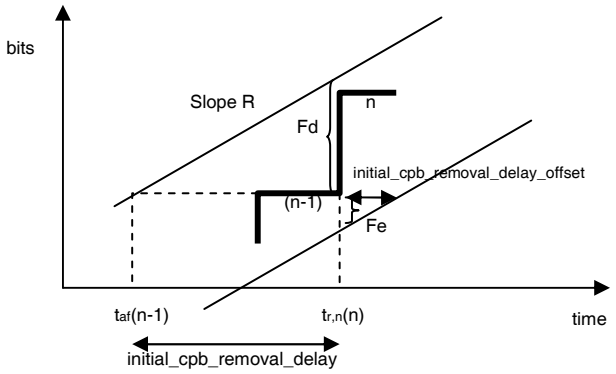

Figure 3. In Buffering period SEI messages, the initial_cpb_removal_delay in the buffering period SEI message is the time between the arrival of the first bit of an IDR, and the time when the frame is scheduled to be removed. The initial_cpb_removal_delay may be calculated as the ratio of Fd to bit rate (slope R), where Fd = Be_maxlimit – (buffer level). Similarly, initial_cpb_removal_delay_offset may be calculated as the ratio of Fe to bit rate, where Fe = (buffer level) – Be_minlimit.

For the buffering period SEI, rather than transmit a decoder buffer level Fd, there is a syntax element initial_cpb_removal delay. According to the standard, (C-14) through (C-16), the initial_cpb_removal_delay is the difference between the time that the IDR is removed, and the time that the last bit from the previous frame arrived, expressed with accuracy of a 90 kHz clock. This is illustrated in Fig. 3. Note that the slope R, which is the channel bit rate, is the ratio of Fd to initial_cpb_removal_delay. Due to fixed-point quantization, the initial_cpb_removal_delay may not correspond to the exact number of bits for Fd, but should be within (bit rate) / 90,000 bits.

5. PICTURE TIMING SEI MESSAGES

The picture timing SEI message is expected whenever nal_hrd_parameters_present_flag is set in the VUI parameters of the sequence parameter set. There is also a vcl_hrd_parameters_present_flag. The picture timing SEI gives, at minimum, cpb_removal_delay and dpb_output_delay for each picture. For baseline profile H.264, there are no B-frames, so we may assume that the

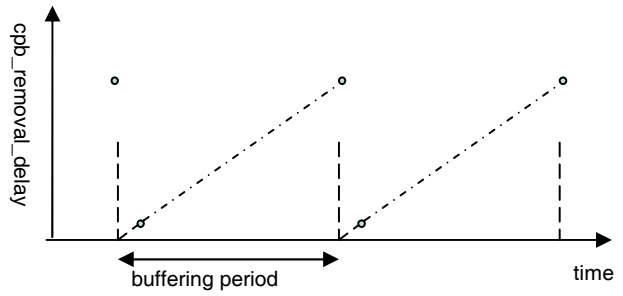

Figure 4. In Picture timing SEI messages, cpb_removal_delay is always relative to the start of the previous buffering period, including for the first access unit of a buffering period.

picture display order matches the order of decode. We further assume instantaneous display, with dpb_removal_delay of zero. The cpb_removal_delay indicates when a frame skip occurs, to keep the encoder and decoder levels complementary. The encoder cannot manage the decoder buffer level unless there is a direct correspondence between the decoder buffer level and the encoder buffer level.

The cpb_removal_delay is used, as specified in C.1.2, to determine when the bits are ideally removed from the decoder's buffer. The nominal removal time, given in (C-8), is relative to the nominal removal of the first access unit of the previous buffering period. Note that for the first access unit of a buffering period, i.e., the first picture timing SEI message following a buffering period SEI message, the cpb_removal_delay is relative to the beginning of the *previous* buffering period. Generally, the resulting removal time should be the same as indicated in the buffering period SEI message, but once a decoder has initialized its HRD, it disregards all subsequent buffering period SEI messages, and does not use buffering period SEI messages to compute the removal time. For a fixed frame rate, the cpb_removal_delay values will vary periodically, as shown in Fig. 4, with the highest value at the start of the buffering period, and the smallest value for the second access unit in the buffering period.

8. TESTING WITH VPROVE

When HRD parameters are present in a bitstream, Vprove [4] can be used to test output timing conformance. Fig. 5 shows a plot of the hypothetical decoder's buffer level. This can be selected under the Analysis menu, or with an icon that looks like a plot. On the x-axis, the graph shows the timing for each frame. This example shows a sequence coded at 10 frames per second. At the bottom, there is a slider to view different time intervals. Error messages are generated for any inconsistencies in initial_cpb_removal_delay or cpb_removal_delay.

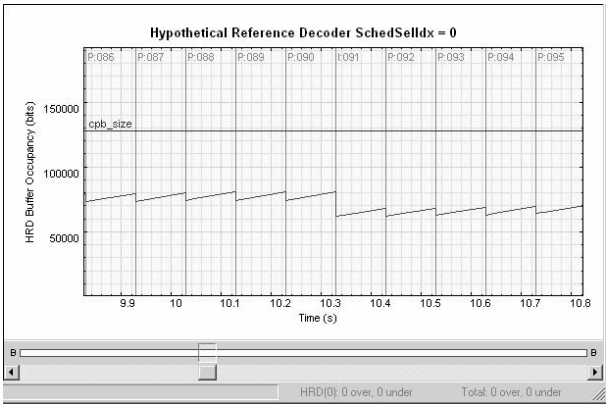

Figure 5. Vprove provides a plot of the HRD buffer occupancy, by selecting "view buffer analysis."

9. SUMMARY

The H.264 HRD parameters specify the buffering required for smooth playback of a streamed bitstream whose bit rate may vary from frame to frame. For a prerecorded video sequence, the minimum buffer size B can be computed from the frame sizes [2]. For real-time encoding and transmission, an HRD conformant bitstream can be generated by placing appropriate constraints on the rate control, and by inserting the corresponding SEI messages to support streaming.

REFERENCES

[1] "Text of ISO/IEC 14496 10 Advanced Video Coding 3rd Edition," ITU-T Recommendation H.264, July 2004.

[2] J. Ribas-Corbera, P. Chou. and S. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," IEEE Trans Circuits Syst. Video Techol., pp. 674-687, July 2003.

[3] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," IEEE Trans. Circuits Syst. Video Techol., pp. 361-372, Dec 1992.

[4] VProve, http://www.vqual.biz/ aka Tektronics MTS4EA