Variable Bit-rate Video Coding

Antonio Ortega*†
Integrated Media Systems Center
Signal and Image Processing Institute
Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, California 90089-2564

Abstract

Emerging network infrastructure is making it easier to transmit video with a variable rate per frame. Variable bit rate (VBR) coding can be seen as the "natural" representation for video, given that each individual frame will have a different level of complexity, and thus require a different number of bits to be compressed with the same decoded quality. However, "pure" VBR is not used in practice, in part because of coding considerations (the purest form of VBR would assume the perceptual quality to be fixed) and in part because typical transmission environments may not allow arbitrary variations in transmission rate (or the transmission costs would preclude it). Thus, in practice there are many types of VBR video. The goal of this chapter is to discuss the various approaches as well as the coding strategies that allow the coder to operate at variable rate. A particular focus of our discussion will be the rate control algorithms, i.e., algorithms that allow the encoder to modulate its output to meet various constraints, such as minimizing distortion, preventing buffer overflow, meeting network transmission constraints, etc.

*Address all correspondence to: Antonio Ortega, <ortega@sipi.usc.edu>, ph: 213-740-2320, fax: 213-740-4651

†Antonio Ortega was supported in part by the National Science Foundation under grant MIP-9804959 and by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, with additional support from the Annenberg Center for Communication at the University of Southern California, the California Trade and Commerce Agency

Contents

1 Introduction

The goal of this chapter is to establish the importance of variable bit rate (VBR) coding of video as a means of providing good decoded video quality, and to show how VBR video can be best incorporated within a networking infrastructure, and in particular within a network environment that supports VBR transport. Note that throughout this chapter we will use the terms VBR and CBR (Constant Bit Rate) to refer to both coding modes (where the rate refers to the number of bits used per frame) and transport modes (where rate refers to the number of bits that can be transmitted during certain periods of time). Whether coding or transmission is being refered to should be clear from the context or will be stated explicitly.

We start by showing how video compressed at constant quality will result in a variable number of bits per frame. Thus, if the bitstream generated by a video encoder has to be kept very close to a constant rate (e.g., for transmission over a constant bit rate, CBR, channel), there will be a penalty in terms of quality (e.g., if a particular scene requires too many bits, for the available channel bandwidth, then the quality will have to be lowered).

Emerging networks provide a great deal more flexibility than traditional point-to-point dedicated links or circuit-switching networking techniques when it comes to bandwidth utilization. For example, new network environments based on packet switching, such as the Internet, lend themselves very naturally to VBR transmission, and thus seem to be better suited to transmit video streams, which tend to be VBR in nature. Therefore the variable rate produced by the encoder can potentially be matched to a variable transport rate, whereas in a CBR transmission environment the video encoder would have to adjust its rate to match the constant channel rate constraint. Still, while VBR transport is appealing, even if the network supports variable rate transmission it is not necessarily true that it can transport without loss any arbitrary video bitstream, in particular if transmission is subject to strict delay constraints. In this chapter we concentrate on the problem of matching VBR video coding to various kinds of VBR transport modes. Details on these various modes can be found in other chapters in this book, as well as in papers dealing with VBR transport of

video [1, 2, 3].

In Section 2 we begin by providing concrete examples of the variable rate nature of video sequences by examining bit rate traces for a video sequence coded with a single quantization stepsize. Obviously a variable rate compression can be achieved in many different ways and this section briefly highlights some popular techniques that lead to different conding rates for the same input sequence.

In Section 3, we then devote our attention to the delay constraints that arise in typical video applications. These delay constraints are the most general constraints imposed on a video communications system, since they depend solely on the specific video application being considered, rather than on the channel characteristics or the type of transport mode (VBR or CBR) that is selected. Networked video applications can be categorized into three main classes, (i) live interactive video, LIV, (e.g., video conferencing), (ii) live non-interactive video, LNIV, (e.g., broadcasting), and (iii) playback of pre-encoded video, PEV. For each of these scenarios we discuss the specific form of the delay constraints. While all three application classes are likely to be encountered in practice, in this chapter we discuss mainly live applications (LIV or LNIV) as they are more challenging in general.

Typically the video encoder has to select the coding parameters, and thus the coding rate, so as not to violate the delay constraints. Clearly, the degree of difficulty in meeting these constraints will depend on the bandwidth and delay characteristics of the transmission channel. Thus, in a network with unlimited bandwidth and low delay one could transmit very high quality video without risk of violating the delay constraints. In Section 4 we discuss several possible transmission modes that are often encountered in existing systems. These modes differ in the variability of the channel rate (e.g., CBR vs VBR) as well as in the amount of information that the encoder has about the channel conditions.

Given these transmission classes, in Section 5 we define three general problems that each lead to a different set of rate constraints for the encoder. These cases are (i) the CBR channel, (ii) the VBR channel with deterministic rates, and (iii) the VBR channel with randomly varying rates. For each of these scenarios we will provide rate constraints that the

decoder will have to meet in order to avoid data loss due to excessive delay.

Section 6 then presents a summary of the work that has been done in recent years to enable rate control at the encoder to meet the aforementioned constraints. A complete survey of rate control techniques falls outside of the scope of this chapter and thus we will emphasize approaches that attempt to optimize the selection of rates. While in real-world applications simpler techniques may be selected, focusing on the optimized approaches allow us to discuss some of the desirable features that all rate control algorithms (whether emphasizing optimality or simplicity) should seek to achieve. Finally, Section 7 summarizes the key ideas in the chapter and points to some of the transmission environments where VBR transmission of video is likely to remain a challenging area of research in the short term.

2 Variable rate compression of video

It is easy to define VBR video encoding: a video encoder is VBR if it produces a variable number of bits per frame. Of course this is not a very useful definition in that, as described in the video coding standards chapters, practically all video encoders of interest are VBR. Thus, this section will be devoted to discussing in more detail how VBR video is produced, and what its quality implications are. In particular we will emphasize that for practical video coding schemes there are many ways of producing a VBR stream, and these will have to be compared in terms of quality or distortion (how the quality or the distortion is measured will be discussed next in Section 2.1).

There are three factors that explain the variable rate nature of compressed video, namely, the coding frame type, the input video characteristics and the coding parameters. First, in most practical coders there exist various frame coding modes, which result in different rates for the same input frame. For example, MPEG-2 coders include I, P and B frames, and since P and B frames utilize motion compensated prediction, while I frames do not, P and B frames result in lower overall coding rate (refer to the video coding standards chapters for further details).

Second, even if one considers a single type of frame (e.g., all frames are coded in Intra

mode), and the remaining coding parameters are fixed, there will be rate variations due to the changes in the scene contents. These variations, as will be seen in Section 2.2, can be very significant. For example, a difference of a factor of two in bitrate between frames in a given sequence is fairly common.

Finally, the output rate depends on the selection of coding parameters. For the sake of flexibility and in order to incorporate the maximum number of features into the encoder, most standard video compression systems incorporate many different coding parameters (from quantizers, to coding modes, to motion compensation modes) that can be used to modulate the output rate. We will provide a brief overview of these in Section 2.3, and we once again refer to the video coding standards chapters for detailed descriptions of these algorithms.

2.1 Rate-Distortion trade-offs for VBR video

Video compression algorithms aim to achieve the best possible quality *delivered to the end user*. Thus, the video quality provided to the end user will depend not only on decisions made at the encoder, but also on how the video data is transported over the network (e.g., if the encoder produces too much data for the channel bandwidth, some of this data could be lost). Thus, an important theme in this chapter is that end-to-end video quality depends on both source and network and for this reason knowledge of the channel characteristics (e.g., its bandwidth, loss or delay characteristics) should be incorporated into the way the video encoder operates. With this knowledge, through intelligent rate control, the video encoder will be able to get the most end-to-end video quality out of the available channel.

For most applications of interest, video is compressed in a lossy manner, i.e., the decoded sequence is not an exact copy of the original video sequence. Perhaps the only two situations where a lossless compression may be needed are (i) scientific applications, where all the available information is likely to be needed for later processing and analysis, and (ii) entertainment applications, where lossy compression could lead to artifacts during production and thus is likely to be introduced only when the final version of the video is ready.

Thus, we will consider here lossy compression only. In order to meet our stated goal of providing the best video quality to the end user we first need determine ways of measuring this lossy video quality. Knowledge of perceptual issues for image coding [4] is more developed than for video, although in the latter case some basic facts are known [5]. The quality of individual frames in a video sequence can be assessed in much the same way as for still images. However the fact that video quality could change over time has to be taken into account. A typical assumption is that overall perceptual quality for a complete sequence will depend on the quality of its worst quality scene, given that the worst scene is sufficiently long [5].

While some general guidelines may apply, the quality assessment will also depend on the type of application being considered. Thus, quality expectations for a videoconferencing application are likely to be much lower than for the delivery of a movie over a network. Note also that we do not consider here the effect of packet losses on perceptual quality (these are considered in the error concealment chapter).

A popular approach for perceptually-based compression is to introduce in the encoding process some constraints that are perceptually meaningful and then let the encoder optimize an objective metric, such as mean squared error (MSE) or weighted MSE. One example of this approach is the design of quantization matrices for DCT–based coding (see the video coding standards chapter), where frequencies are given different weights according to their relative perceptual importance. Another example comes in the bit allocation among frames, where a generally accepted "rule of thumb" is that quality should be kept more or less constant from frame to frame, and therefore a good bit allocation should try to penalize large changes in quantization stepsize within a set of frames.

For applications that involve real time encoding and transmission (i.e., where the low overall delay is needed) these so called objective distortion metrics will be sufficient. As will be seen, for off-line encoding applications more sophisticated techniques are required to supplement MSE-based approaches as it becomes possible, for example, to determine which scenes in a given video sequence require more rate to achieve similar perceptual quality.

2.2 Achieving constant quality requires variable rate

Compression is achieved by exploiting the existing redundancies in the source (e.g., spatial, temporal and statistical redundancy). Since the level of redundancy varies from frame to frame, it comes as no surprise that the number of bits per frame be variable, even if the same quantization parameters are used for all frames.

Consider for example the spatial allocation of bits in a frame. As described in previous chapters, the number of bits required to encode an image block, with a given quantization stepsize, will depend on the frequency contents of the block. Therefore, images containing high frequency information will require more bits than frames having predominantly low frequency content. Similarly, scenes with significant motion will require more bits than scenes exhibiting little or no motion.

Thus, a fundamental result in video coding is that maintaining a constant (perceptual or objective) quality throughout a sequence requires a variable rate allocation. This large variability has been observed in video trace analysis work such as that in [6], or [7]. An example can be seen in the trace shown in Figure 1 corresponding to the *Mission Impossible* movie, where the sequence has been coded with a constant quantization parameter in a Motion JPEG framework[1]. As can be seen, changes of over a factor of two are common, and for longer sequences even larger variations are likely. Fig. 2 also shows how the MSE varies substantially from frame to frame. Note that if an MPEG coder [9] were used instead, the average rate per group of pictures (GOP) would show similar variability, but variations in rate within a GOP would depend mostly on the frame type (e.g., I-frames vs B-frames).

These traces illustrate how the problem of achieving constant perceptual quality for a whole sequence takes different forms depending on whether live or pre-encoded video is considered. On the one hand, if live video is considered it is not possible to guess what the contents of future frames will be, and therefore, at a given time, the encoder will have to make coding decisions to target the quality of the current frames.

[1]Each frame is coded using JPEG [8] and the same JPEG coding parameters are used for every frame. We use the measured total number of bits per frame and the average distortion (MSE) in our traces.

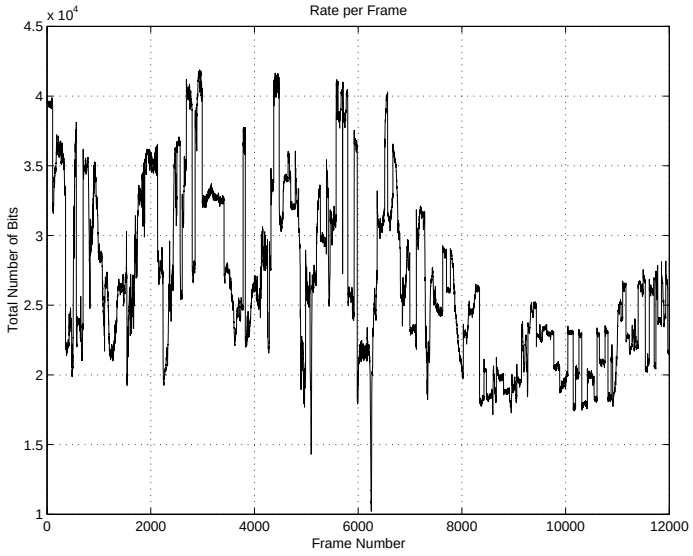

Figure 1: Mission Impossible trace: Rate

On the other hand, let us consider authoring of compressed movies for DVD storage as an example of pre-encoded video. In this particular scenario the only constraint is the overall storage capacity in the DVD (since data can be read into the player at much higher rate than the typical video coding rates), and therefore the number of bits assigned to different scenes can show significant variations. Moreover, given that these movies are compressed once, but are decoded many times, a relatively complex encoding process is acceptable as long as the achievable quality gains are significant.

Thus, a two-pass approach is typically used, where the first pass aims at locating scenes that have higher "complexity", i.e., will require a larger amount of bits in order to achieve the desired quality. This can be accomplished by coding the whole sequence with a single quantizer scale and then determining which parts of the sequence have required more bits (see for example [10, 11]). This information is then used in the second pass of the algorithm where the encoder is allowed to adapt the quantization step size, based on the results of the first pass. For example, for "easy" scenes it may possible to decrease the rate, with respect to the first pass, without affecting the perceptual quality.

The concept of scene complexity is illustrated by the plots of Fig. 3. The top two plots

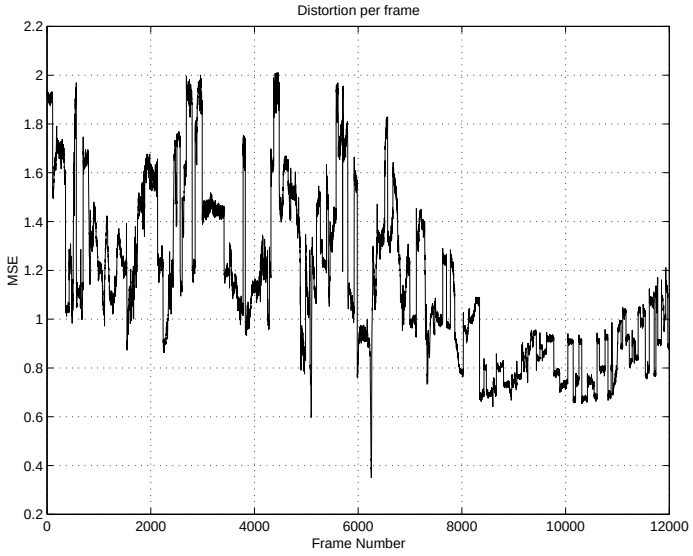

Figure 2: Mission Impossible trace: Distortion

are normalized versions of Figs. 1 and 2, where we have normalized by dividing all rate or MSE values by the maximum rate or MSE in the sequence (e.g., $R_i' = R_i/max_i(R_i)$, where R_i and R_i' are respectively the rate and normalized rate for frame i.) The bottom plot shows the normalized value per frame of the product of rate and MSE (i.e., $D \cdot R$) for each of the selected quantizers. Note that since typical rate-distortion characteristics can be roughly modeled as having the form $D = K/R$, for a given constant K, then $D \cdot R$ does provide an approximation to the complexity of a particular frame. Under the $D = K/R$ modeling assumption a large K means that many bits are required to achieve low distortion and thus a frame with large $D \cdot R$ can be deemed to have high complexity. Conversely, a small K would indicate low complexity, since a small number of bits can significantly reduce distortion. In the particular example of the bottom curve of Fig. 3, the final section of the sequence (say, after frame 8000) seems to be less complex that the beginning and thus a two-pass algorithm would tend to allocate more bits to the beginning of the sequence than to the end.

Interestingly, for professional DVD applications two-pass algorithms are not enough and there is usually a third stage before the final compressed version is produced. In this final stage a person actually views the decoded sequence and applies selected modifications to

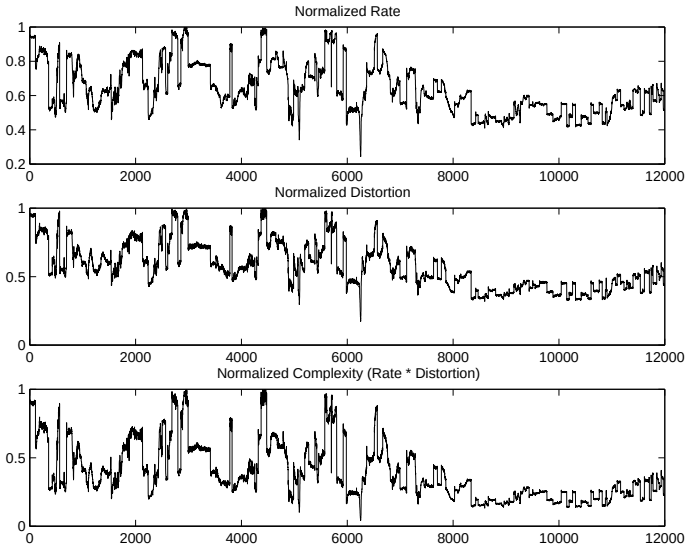

Figure 3: Mission Impossible trace: Normalized rate, distortion and complexity per frame. Note that variations from scene to scene are very significant.

the coding parameters to ensure that any remaining visible artifacts are removed. The role of this person (sometimes called the "compressionist"!) clearly demonstrates not only that MSE has its limitations but also that reliable, automatized, perceptual quality metrics remain difficult to define, and therefore achieving near perfect perceptual quality (under rate constraints) remains almost an art.

In summary, we can assume, as in [11], that through frame complexity estimation or other techniques it will be possible to determine the "ideal rate" for each segment of a sequence. Here ideal rate is defined as the minimum number of bits that produces transparent quality, i.e., such that the original segment and the decoded one are indistinguishable. Such an ideal rate, or an alternative rate allocation aiming at constant perceptual quality, will be different for each segment in the sequence, and therefore coding at constant quality will necessarily require variable rate per frame.

2.3 Mechanisms available to produce variable rate video

In the example of Figs. 1 and 2 we kept a fixed quantization step-size for all frames and employed Intra-only coding (i.e. frames were all coded using JPEG with same coding pa-

rameters for all frames). Obviously, as indicated in the video coding standards chapters, a state of the art coder incorporates many other coding options. Examples of how these coding options can be judiciously chosen to optimize performance are given in [12]. Here we will just provide a rapid overview of available mechanisms. We refer to the chapters on compressed video standards for further details about the various parameters that can be chosen for each frame, and to [12] for a detailed description of techniques that can enable an efficient selection of parameters.

2.3.1 Quantizer selection

The most immediate approach to produce a variable rate is to change the quantization step-size. All video compression algorithms used in practice, including those based on standards, allow flexibility in the selection of quantizer (see the chapters on compressed video standards). Typically, each individual block or macroblock within a frame can be assigned a different quantization step-size. These features are included in video compression algorithms primarily to allow the systems to operate at several channel rates and to provide the benefits of a good bit allocation within a given image. For example, one can allocate a finer quantizer to those parts of the image that may be more sensitive to quantization errors. Well known techniques such as Lagrangian optimization can be used to achieve an efficient allocation in the rate-distortion (RD) sense among the blocks [13, 14, 12].

For video coding, quantizer selection is further complicated by the fact that motion compensated prediction is used. Because motion compensated prediction is based on transmitting the difference between a previously quantized frame and the current frame, quantizer selection for one frame affects the RD performance for future frames [15].

2.3.2 Coding mode selection

In addition to the quantization stepsize, other coding parameters (or coding modes) can also be chosen to be used for each macroblock. In particular, for Inter frames each macroblock can be predicted, skipped, or coded in intra mode. Typically, this decision can be made by

considering which of the modes provides the least energy residue in the motion compensated predicted frame. However, alternative approaches can be used, where the trade-off between rate and distortion is considered. More generally, there could be other modes of operation that can be selected, including for example the type of motion compensation used (e.g., overlapped vs. non-overlapped, motion vector block size, etc.) [12].

2.3.3 Frame-type selection

Consider as an example MPEG-2 (see the corresponding chapter). The three frame types (I, P, B) can be chosen according to several criteria. For example, a specific application such as stored video may require that I frames be placed at fixed intervals (e.g., one every 0.5 secs) in order to enable random access and features such as fast forward and rewind. In a real time application, frame selection can be used to modulate the encoding rate. For example, the spacing between I frames can increase if lower rate is desired. As before the frame-type selection can be optimized based on the RD trade-off [16].

2.3.4 Frame skipping

Finally, certain video coding standards allow frame skipping leading to a variable frame rate and therefore variable rate video. Frame skipping means reducing the number of frames per second that are transmitted (for example from the original rate of 30 frames/s to, say, 10 frames/s) so that if lower source rate is required one can transmit fewer frames but use more bits per frame. This option is obviously more suitable for interactive video and in general low quality applications, where it is not crucial to maintain a constant number of frames per second. In particular, for computer based applications, as opposed to systems where specialized hardware is used, it may be easy to support the display a variable number of frames per second and to even introduce techniques for interpolation at the receiver.

3 Delay Constraints for Real-time Decoding and Display of Video

The two main ideas presented in the previous section are that constant quality video tends to require a variable bit rate and that there are many ways in which a variable rate bitstream can be generated. The encoder has the ability to select parameters for each of these modes and as in [12] methods such as Lagrangian optimization and dynamic programming [14] can be used to search the space of all possible operating points to find the one that is better in a rate distortion sense. Section 6 will discuss issues of rate control, that is, how to allocate bits among different frames in a sequence. Obviously these problems only arise when there are constraints (as there surely will be) on the total rate that can be used. The goal of this section is to describe the possible delay constraints that will make transmission at the ideal rate not feasible in practice. More detailed discussions of the various delay constraints can be found in [17, 18, 19].

3.1 Delay Constraints: Preventing Decoder Buffer Underflow

Let us assume that a video sequence with a total of M frames is transmitted at a fixed number of frames per second[2]. Let R_i be the number of bits assigned to the i-th frame[3]. Let $t_d = 0$ be the time at which the first bit of the video sequence is received by the decoder. Time is measured in units of number of frames at the decoder. Thus, $t_d = i$ corresponds to i frame intervals having passed, where one frame interval lasts δ_f seconds. For example, assuming that 30 frames per second are being displayed at the decoder, $\delta_f = 1/30 secs.$ Let C_i be the number of bits received by the decoder buffer during the i-th frame interval. Note that C_i does not correspond necessarily to compressed data for the i-th frame. This is because data for a particular frame could be transmitted over several frame intervals, depending on bandwidth conditions and the number of bits used to code the frame.

[2]Our formulation could be easily adapted to having a variable number of frames per second but we make this assumption for simplicity.

[3]It would be possible to consider smaller basic units, e.g., a set of blocks within a frame, but for simplicity we consider frames as our basic unit.

We will assume that the decoder actually begins decoding and displaying frames after ΔN_d frame intervals, or $\Delta T_d = \Delta N_d \cdot \delta_f$ seconds, have passed. Thus, the first frame (frame 1) will be decoded and displayed at time $t_d = \Delta N_d$. Although a more detailed analysis of the delays involved, e.g., computation delay, is possible (see for example [20]), here we assume that frames are instantaneously decoded and displayed.

Under this framework, the system will function normally as long as the decoder buffer is "fed" frames sufficiently fast so that after time $t_d = \Delta N_d$, the decoder can play δ_f^{-1} frames per second. Note that so far we have made no assumption as to how the frames are generated at the transmitter or when they are transmitted. Since we assume that frames are played back at a constant rate, it follows that if the first frame is decoded at time ΔN_d, then the i-th frame must be available at time $i-1+\Delta N_d$, i.e., $(i-1+\Delta N_d)\cdot\delta_f$ seconds after the first bit was received at the decoder. In the most general case, the goal of the transmitter will be to avoid decoder buffer underflow (see also Fig. 4). Obviously decoder buffer overflow as well as encoder buffer overflow and underflow also have to be addressed. However overflow at either buffer can be addressed by providing sufficient buffer memory. Given the continued decrease in memory costs, the assumption that sufficient memory is available seems to be reasonable for many applications. Encoder buffer underflow can be easily avoided by not transmitting data. Thus from now on we will focus on decoder buffer underflow prevention and this will be the main objective of the rate control algorithms of Section 6.

In our system, the compressed video data is placed in a transmission buffer, from where it is drained and transmitted. We assume that the video encoder and the transmitter are combined so that the video enconder can control when data is transmitted.

Formulation 1 Decoder buffer underflow prevention *In order to prevent the decoder from losing frames the transmitter has to ensure that all the information corresponding to frame i has arrived to the decoder before $t_d = i-1+\Delta N_d$. This is equivalent to the channel rates having been sufficient to transmit all the necessary data,*

$$\sum_{k=1}^{i-1+\Delta N_d} C_k \geq \sum_{k=1}^{i} R_k, \quad \forall i \tag{1}$$

that is, the total channel rate used up to time $i-1+\Delta N_d$, has been enough to transport the first i frames.

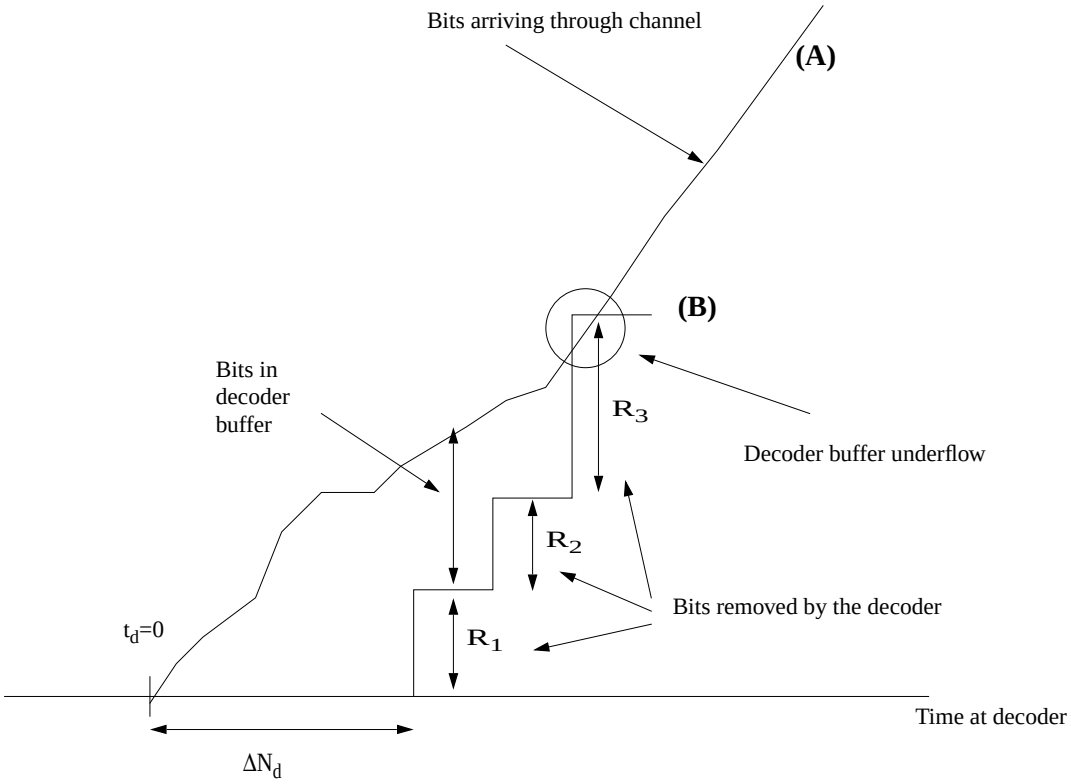

Figure 4: Data received at the decoder and data that has to be available for decoding to proceed. Clearly, if the decoder demand, (B), exceeds the channel supplied data (A), at any time, the decoder will be unable to operate correctly. This is what is called decoder buffer underflow.

Note that we are assuming that no frames are skipped at the encoder. Clearly, if the encoder was allowed to skip frames, some of the time and delay constraints might be less stringent than if all frames had to be encoded and delivered. However, even in the case where the encoder generates a variable number of frames per second, constraints similar to those above would hold, with the only difference being that the frame intervals would have variable size.

We continue to assume that the encoding and decoding delays are constant and thus are not taken into account in what follows. Thus, the two delay components that can result in

a violation of the constraint of (1) are:

- **Transmitter buffer delay**, δ_{tb}. This is the delay due to the time it takes to drain video data corresponding to a particular frame, after it has been placed in the transmit buffer by the encoder. This delay exists only if the channel bandwidth is limited and does not match the data produced for all frames, otherwise data could be drained into the channel as soon as it is produced. Clearly, given the video data, the lower the channel rate, the longer this delay will be.

- **Transmission channel delay**, δ_{ch}. This is the delay suffered by packets of video data being transmitted through the network, i.e., from the time they have been extracted from the transmit buffer, to the time when they are available at the receiver buffer. There are numerous scenarios where this delay may be variable, such as transmission over a shared network or transmission over a lossy link (here delay is assumed to include the time needed for retransmission of lost data if applicable).

Clearly, then, for a given coded sequence the channel resources will have to be selected such that the total delay $\delta_{tb} + \delta_{ch}$ incurred by each video frame does not result in a violation of the decoder buffer underflow constraint.

As an example of the delay variability, consider Fig. 5 which depicts the number of frame intervals required to transmit 100 frames at a constant rate equal to the average rate of the sequence. This was generated by taking the rate trace of Fig. 1, computing the overall average rate, then dividing the total rate for each group of 100 frames by 100 times the average frame rate. As can be seen, when most of the frames are below the average rate, transmission can be completed in less than 100 frame intervals, while when the rate is greater a longer delay is incurred. In other words, if a delay constraint of 100 frame intervals ($\Delta N_d = 100$) were to be enforced, lower average rate would be required for many frames in the initial part of the sequence.

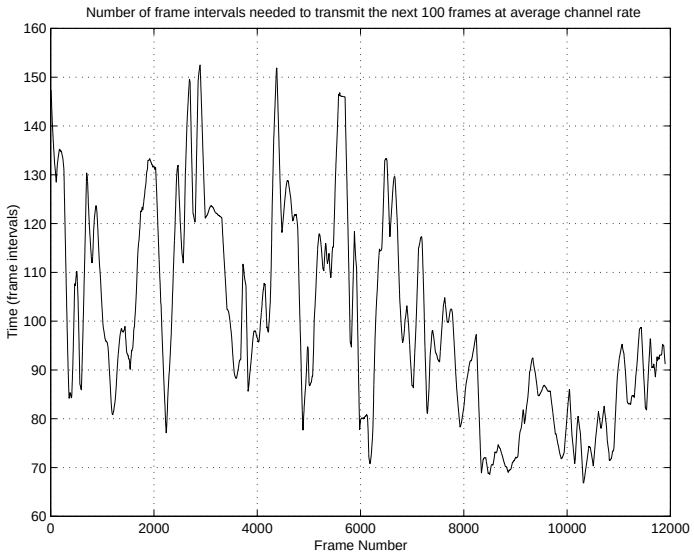

Figure 5: Total Delay in transmitting 100 frames. For each frame we plot the number of frame intervals that would be needed to transmit the next 100 frames under the assumption that the average rate per frame for the whole sequence is used as the constant channel rate. Note that the delay oscillates around 100 frames and that some scenes require more than 100 frames for transmission.

3.2 Real-time vs pre-encoded video

Note that the delay constraint comes into play because the decoder cannot "wait" for late frames. Thus, in order to know whether the delay $\delta_{tb} + \delta_{ch}$ exceeds this maximum delay we will need to know first *when* the encoded frames first become available. To do so we have to differentiate between live and pre-encoded video.

3.2.1 Pre-encoded video

Let us consider first the case of offline playback of a pre-encoded source. In this case the whole video sequence has already been encoded and is ready to be transmitted. Thus, in this scenario we can assume that the transmission buffer is very large and contains the whole video sequence at the beginning of transmission. Even if, as is the case for disk based video servers, there are two separate levels of storage (e.g., disk and transmission buffer) it is still likely that the bottleneck will be in transmission, rather than in moving data from the storage device into the transmission buffer. Thus, since all the frames are available at the

transmitter from the beginning, and the deadline depends only on the decoder, the maximum delay that a frame can experience will be variable. For example, as illustrated by Fig. 6, for a given channel rate, if the number of frames per second transmitted is large (e.g., more than 30 frames per second) then subsequent frames may be transmitted over a longer interval of time without being lost at the decoder. Thus, in the PEV case, ignoring the channel delay, we have ΔN_d frame intervals to transmit the first frame and in general $\Delta N_d + i$ intervals to transmit the i-th frame since all the frames are available at the transmitter at the start of the video transmission. In summary, as seen in Fig. 6, the maximum delay per frame is different for each frame and there is more flexibility in transmission than in the live video case, which we consider next.

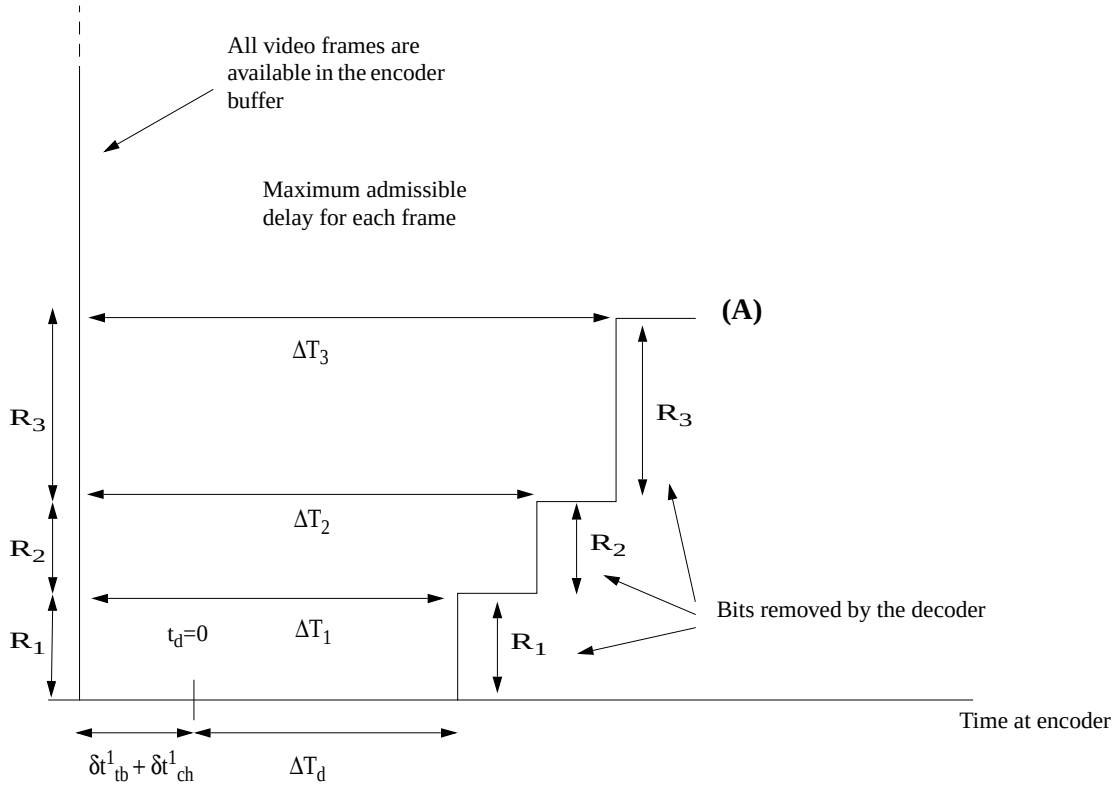

Figure 6: Delay constraints in the PEV case

3.2.2 Live Video

Consider the scenario where video is captured in real-time, then compressed and transmitted. Assume that the first frame experiences delays δ^1_{tb} and δ^1_{ch} after the time the frame has been captured and compressed. Normally δ^1_{tb} will tend to be very small or zero, since there are no other frames waiting to be transmitted. However a delay may exist because time is needed to set up the transmission or because the encoder decides to store a few frames in the encoder buffer before starting to transmission actually starts (e.g., in order to avoid encoder buffer underflow).

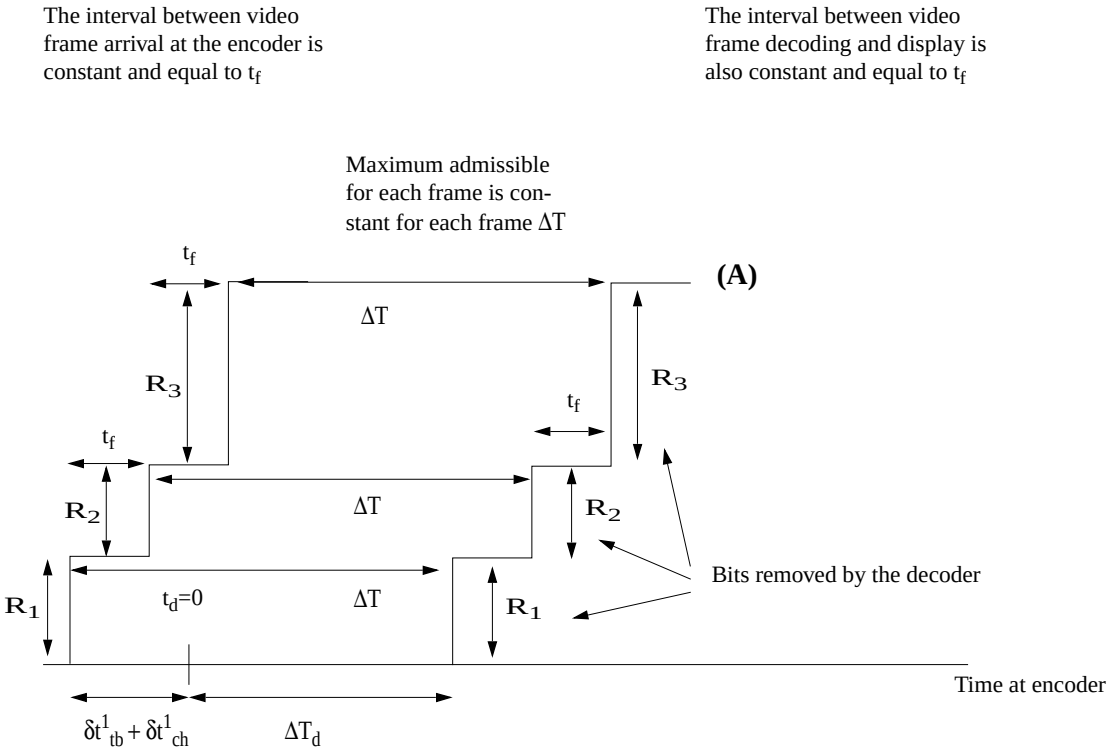

Figure 7: Delay constraints in the live video case

Thus, the first frame will experience, from capture to display, an overall end-to-end delay, ΔT, which can be written as

$$\Delta T = \delta^1_{tb} + \delta^1_{ch} + \Delta T_d, \tag{2}$$

where ΔT_d is the time the decoder waits before starting the decoding process, measured from the time the first bits of video data were received. That is, the time between the capture and encoding of frame i and its decoding and display will be ΔT for any i. It is important to note that even though all the delay components will vary over time, so that frame i will experience delays δ_{tb}^i and δ_{ch}^i, the overall delay ΔT has to remain constant. This is illustrated by Fig. 7.

Thus, in this real-time encoding case, frames arrive at the encoder at a constant rate (δ_f^{-1} frames per second), are encoded and put in the encoder buffer. The first frame is decoded after $\Delta N_d = \Delta T_d/\delta_f$ frame intervals and, unless frames are dropped, the i-th frame also has to be available at the decoder by time $i + \Delta N_d$. In this case each frame spends exactly ΔT seconds in the system and there are always exactly $\Delta N = \Delta T/\delta_f$ frames in the system.

Clearly, given that ΔT determines the maximum delay a frame experiences, having a large ΔT chosen will tend to reduce the chance of violating the delay constraint. As we will see in Section 6 this can alternatively be seen as meaning that the rate control algorithm will have more degrees of freedom and thus the overall video quality will be better.

3.3 Interactive vs Non-interactive video

The tightness of the constraint will also depend on how much overall delay the specific application can withstand, i.e., how large can ΔT be for a specific application.

In a non-interactive application this delay ΔT has the effect of an initial latency in the playback. Anyone who has used video or audio feeds from the Internet may have observed how data is initially buffered before playback begins. One question that comes to mind is what benefit, if any, can be derived from having a large ΔT. In the case of unreliable environments such as the Internet the answer is immediate. Since the end-to-end delay determines the delay constraint for any given frame, longer ΔT means that larger variations in the delay components can be tolerated. This means that the system will be more robust to losses and jitter (more time for retransmission) but also that the individual transmission delays of each frame (source rate for the frame divided by allocated channel rate) can vary

significantly. This enables more flexibility in allocating bits per frame and allows us to approach "ideal" VBR coding rates. Thus, in general, larger ΔT tends to result in higher quality.

The live interactive video scenario only differs from the above case in that the overall delay has to be lower, i.e., ΔT has to be small. A typical assumption in this case is that a 150msec roundtrip delay (from the time one of the users of the system says something, until the time this user receives an answer). This constraint severely limits the potential for improving performance through rate control. Under these delay constraints, only a few frames are in the system at any given time and the encoder and decoder buffers have to be small.

4 The Impact of Transmission Modes on End-to-end Video Quality

To this point we have described the delay constraints as observed by the encoder and decoder but have not considered how the achievable end-to-end performance is affected by the selection of transmission mode.

As an example consider the extreme case where the channel delay δ_{ch} is negligible and where each frame at the transmitter buffer is transmitted completely within a single frame interval, that is, $C_i = R_i$. In this case, an example of VBR transmission, the system would be able to operate with practically no delay. However, given rate traces such as those presented earlier it is unlikely that such a network service will be available, unless capacity is significantly overdimensioned with respect to the expected number of users.

VBR transmission of video encoded at variable rate seems to be a desirable objective, since it would match the variable rate output of a coder to a variable transmission rate over the channel. Substantial activity in exploring VBR transmission of video dates back to the mid to late eighties (for example [21, 22, 23]) and is still ongoing [24]. However, researchers have used the term VBR to describe different types of transmission modes, ranging from modes that guarantee quality of service (QoS) in the data delivery, to approaches with "best

effort" performance, as is the case with transmission over the Internet. Each of these VBR modes presents different challenges to the video application to ensure transmission under the constraints of Formulation 1.

There are numerous transmission modes that can be used to transport video data. Our objective here is not to provide an exhaustive enumeration and we refer the reader to the chapters on ATM and on feedback of rate and loss information, where some of these transmission modes will be presented. Instead we provide an overview of the properties of these transmission modes. We present these at a high level, with our emphasis being the clarification of how selecting one mode or another will affect the video transmission. Our motivation is to provide a simple classification of transmission characteristics, and in particular as they are relevant to the video encoder.

4.1 CBR vs. VBR transmission

VBR transmission has been deemed attractive for both source quality and network utilization. Thus, some of the promised advantages of VBR video combining VBR coding and transport are

- (i) *Better video quality* for the same average bit-rate, by avoiding the need to adjust the quantization as in CBR, i.e., we may be able to operate close to the "ideal rate" for the source,

- (ii) *Shorter delay*, since the encoder buffer size can be reduced without encountering an equivalent delay in the network. In the simplest case, if we can use for each frame as many bits as needed to transmit it within a single frame interval, the delay will be low indeed.

- (iii) *Increased call-carrying capacity* because the bandwidth per call for VBR video may be lower than for CBR for equivalent quality. This is also described as statistical multiplexing gain, with the basic idea being that if a number of VBR sources are

grouped together they are unlikely to all make usage of their maximum rate at the same time, so that lower-than-peak-rate bandwidth allocation is possible.

While these potential advantages were heavily emphasized in early packet video papers, further research has shown that no design can maximize them simultaneously. In this chapter we concentrate on video coding aspects, but it is clear that, as argued in [24], a VBR transport design can have advantages for both video and network sharing and thus it would have to be evaluated based on both types of metrics.

4.2 QoS Guarantees vs Best Effort

Much of the early work on packet video assumed transmission under some sort of Quality of Service (QoS) guarantees would be available. These QoS guarantees are generally assumed to be probabilistic. For example, for a particular service there may be a guarantee that the delay will not exceed some value more than a certain percentage of the time. While there has been extensive research on QoS guarantees in small scale ATM environments (e.g., a local network, or a switch) there is as yet no deployment of services with end-to-end QoS over the Internet. For this reason video over the Internet operates under completely best-effort conditions. From the perspective of the video encoder/transmitter, operating in a best-effort environment may require that a transmission scheme be designed with features to ensure robustness to packet losses as well as potentially significant variations in delay.

4.3 Constrained vs Unconstrained Transmission Rate

Normally, if QoS guarantees exist, then there are also some constraints on the number of bits that can be transmitted through the channel. Obviously, this is true in the CBR transmission case, where rate is guaranteed to be constant but a strict limit on the channel rate exists. More interestingly, constraints will also be applied in the VBR transmission case, i.e., even though a variable number of bits can be transmitted, the transmission rate is subject to some conditions. For example the long term sustainable rate may have to be below some maximum value and the short term or "peak" rate may likewise be limited.

These constraints are typically defined in terms of some operational measurements (typically simple counters are used), that the network uses to determine whether specific sources are complying with the constraints. These are called policing functions and examples include the Leaky Bucket, the sliding window, and the jumping window [25, 26]. Note that in some cases the constraints are explicit (e.g., when policing functions such as the leaky bucket are used). This is particularly likely when QoS is provided since the transmitter agrees to abide by certain rules in order to receive the promised guarantees from the service provider. In other cases, especially if only best-effort transmission is available, the constraints will be implicit and will only be indirectly observed. For example, if transmission using TCP/IP is used, there will be no explicit conditions on the rate to be transmitted but as congestion arises the transmission rate will have to be lowered.

4.4 Feedback vs No Feedback

As argued in [24], feedback to the video encoder is one of the key characteristics of video (as opposed to data) transmission over packet networks. In data transmission, feedback can only be used to adapt the way the information is sent (e.g., reducing the transmission rate if congestion occurs as in TCP/IP) but the information itself cannot be changed. Thus, in a TCP/IP transmission, rate reduction means that the overall transfer time will be longer, since the data to be transmitted remains the same but the channel rate is lower. Instead, video encoders can modulate the data they produce by adjusting a number of parameters, including quality, frame rate and resolution. This is particularly useful when there are time variations in the channel characteristics (e.g., due to network congestion).

An example of the role of feedback can be seen by considering a generic system, where the video encoder emits encoded data to a buffer and encoded data, or video traffic, is then drained at a variable rate, which is monitored at the User Network Interface (UNI) [27]. This system would be typical of an ATM based interface, but similar methods might be applied under other transport mechanisms. The UNI monitors the transmitted rate and compares the connection parameters to those negotiated between the user and the network

at the time of connection set-up. The network policing functions, if used, can be considered to be implemented at the UNI. Finally, traffic transmitted through the network experiences some QoS. Information about the currently available QoS, as determined for example by the existence of congestion, might be also transmitted back to the UNI and thence to the source (as for example in an Available Bit Rate, ABR, scheme [28]).

Under this generic system set-up we can define the following modes of operation [24]:

1. Unconstrained VBR (U-VBR), where the video encoder operates independently of the UNI. For example the encoder operates with a constant quantization scale throughout transmission. Most video rate modeling efforts have been based on U-VBR traces.

2. Shaped VBR (S-VBR), where the buffer is linked to the UNI, but is not connected to the encoder. In this case, the encoder produces a bitstream that is identical to that in U-VBR. However, now a shaping algorithm can determine the actual transmission rates C_i. While the content of the bitstream is unaffected, the traffic patterns may be smoothed out at the cost of some additional delay.

3. Constrained VBR (C-VBR)[4], where the encoder has knowledge of not only the buffer state but also the networking constraints at the UNI. Thus the video encoder can modulate its output so as to maximize the video quality given all the applicable constraints, including those related to delay and transmitted rate. Here, the bitstream content *is* affected, but the changes are made by the video encoder, which can change the rate in a manner that has the least impact on perceptual quality.

4. Feedback VBR (F-VBR), which adds information about the network state to what is made available to the encoder. This allows the same trade-offs as in C-VBR to be considered, with the additional advantage that the encoder can adjust to changes in the state of the network (for example congestion periods).

[4] This was referred to as Shaped Bit-Rate, or SBR by Hamdi *et al.* [29, 30], but here we use the same naming convention as in [24].

In this chapter we concentrate on the C-VBR and F-VBR scenarios, as these are the only two modes of operation that entail some sort of feedback to the video encoder. These are the most likely to provide the best performance. The rate constraints that will be presented in Section 5 make explicit how the encoder has to operate under the delay constraints of Formulation 1 for each channel transmission scenario. Then, in Section 6 we will provide an overview of algorithms that can be used to meet these constraints.

It is worth emphasizing that in much of the early work (such as [21, 22, 23]) feedback was disregarded. For example the experimental performance analysis work employed bit traces that were generated by video encoders operating "open loop," that is, without any kind of feedback. On the other hand, most practical systems will incorporate some sort of feedback and therefore feedback should be incorporated into the analysis.

5 Encoder Rate Constraints

In the preceding sections we have described the delay constraints and the various transmission modes available. We now derive the rate constraints that the encoder/transmitter has to meet in order to avoid violating the delay constraints, for a given channel configuration. For the purposes of our discussion we will divide the channels into three classes, namely CBR, VBR with predictable, but constrained, channel rates and VBR with unpredictable channel rates.

Note that in designing a complete video transmission system, several parameters, other than transmission rate, need to be taken into account. For example, one can think of end-to-end delay as a resource, since for the same rate the coded quality increases when ΔT_d increases. While this comes at the cost of increased initial latency it may be a worthwhile trade-off for LNIV or PEV applications. Likewise, memory may be a limited resource even if delay is not, so that the overall buffer space in the decoder will have to restricted (e.g., a hand-held device or a set-top box).

Figure 8 provides an illustration of the relevant rate constraints. In this figure (A) represents the total accumulated bits in the transmitter buffer (each step represents the bits

corresponding to one frame). (B) represents the bits corresponding to frames that have been removed from the decoder buffer (each step corresponds to removing one frame so that it can be decoded). Note that (A) and (B) are simply shifted versions of each other, since the frames that are put in the transmitter buffer are eventually removed from the decoder buffer in order to be decoded. Also note that in this example the delay, the difference in the time axis between the (A) and (B), is made up of two components, a delay in starting the transmission and a delay in starting the decoding.

(C) represents the rate at which bits are transported from encoder to decoder. In this case we are representing a CBR transmission mode since (C) is shown as a straight line. Thus (A)-(C) corresponds to the bits remaining in the encoder buffer and (C)-(B) to those present in the decoder buffer.

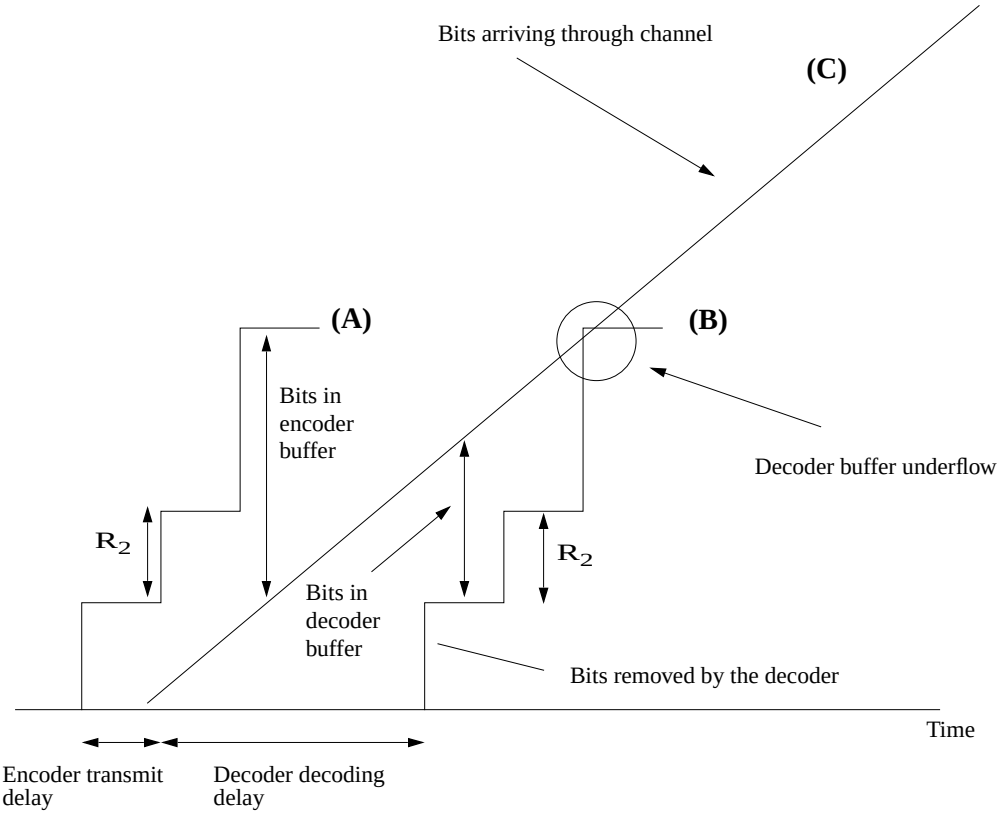

Figure 8: Rate Constraints

From Figure 8 we can also visualize the four possible constraints that one may have to take into account in the coding process, namely, the overflow and underflow conditions at encoder and decoder buffers. Encoder buffer overflow occurs when the buffer memory at the encoder is exceeded, that is, if the difference (A)-(C) becomes greater than the maximum memory. Conversely, if the difference (A)-(C) becomes zero (it cannot be negative) what we observe is encoder buffer underflow and in this situation no transmission takes place, or "filler bits" are sent. At the decoder, if (C)-(B) is too large, the decoder buffer may overflow. On the other hand, if (C)-(B) becomes negative (as in the example of Figure 8) that means that the decoder demand exceeds the bits supplied by the channel and therefore decoder buffer underflow occurs.

In this chapter we concentrate on the problem of decoder buffer underflow (frames arriving too late at the receiver). As mentioned before, encoder/decoder buffer overflow problems can be solved easily if sufficient memory is available and decoder buffer underflow is avoided. Moreover, we will assume that encoder buffer underflow can be handled without penalty through the introduction of filler bits.

5.1 CBR

Clearly, a Constant Bit Rate (CBR) transmission mode, because of its predictable traffic patterns, makes the task of network management easier (in that all decisions can be based on deterministic traffic characteristics), although it also precludes gain via statistical multiplexing. Further, the CBR mode is not well matched to the inherently VBR characteristics of coded video. However, it is difficult to support good quality video using VBR transport unless the network provides feedback to the source, or provides end-to-end delay and loss guarantees with VBR transmission constraints that are known to the source[5].

In a CBR environment we have, as in Figure 8, a fixed number of bits transmitted through

[5]Transmission of video over channels without such feedback is possible, as demonstrated by live video multicasts over the Internet. However, the resulting video quality cannot be as high as in a guaranteed transmission environment, since the video encoding algorithm has to be made robust to almost certain channel losses (and thus both video quality and compression factors suffer).

the channel. Let us consider now the state of the encoder and decoder buffers, under the condition that, as in Formulation 1, frames put in the encoder are removed from the decoder buffer after a fixed number of frame intervals ΔN_d has passed. To simplify the notation, we assume that the channel delay is negligible and that data is sent from the transmitter as soon as it is available, i.e., $\delta_{tb}^1 + \delta_{ch}^1 = 0$, and in general $\delta_{ch}^i = 0$. Thus the only delay in the system corresponds to the ΔN_d frame intervals that the decoder waits before removing the first frame from the buffer. Under these conditions the encoder and decoder buffer states at the i frame interval are respectively,

$$B^e(i) = \sum_{j=1}^{i} R_j - \sum_{j=1}^{i} C \tag{3}$$

$$B^d(i) = \begin{cases} \sum_{j=1}^{i} C - \sum_{j=1}^{i-\Delta N_d} R_j, & \text{when } i \geq \Delta N_d \\ \sum_{j=1}^{i} C, & \text{when } i < \Delta N_d. \end{cases} \tag{4}$$

Thus the encoder contains simply all the bits produced up to that time, minus those that have been transmitted, at the constant channel rate C. The decoder buffer at time i on the other hand contains all the bits received from the channel up to that time, minus the frames that have been decoded. Note that, consistent with our delayed decoding assumption, at time i the $i - \Delta N_d$-th frame is being decoded. Note also that we assume that there is no encoder buffer underflow, i.e., $\sum R_j$ is always $\sum C$, to preserve the linearity of the constraints. This is an easy constraint to meet in practice since the encoding rate can be increased so as to avoid underflow.

If we now combine the encoder buffer occupancy (3) at time i and decoder buffer occupancy (4) at time $i + \Delta N$, we have that:

$$\begin{aligned} B^d(i + \Delta N_d) &= \sum_{j=1}^{i+\Delta N_d} C - \sum_{j=1}^{i} R_j = \sum_{j=i+1}^{i+\Delta N_d} C - (\sum_{j=1}^{i} R_j - \sum_{j=1}^{i} C) \\ &= \sum_{j=i+1}^{i+\Delta N_d} C - B^e(i). \end{aligned} \tag{5}$$

Thus, in order to prevent the decoder buffer from underflowing, we have to keep the right hand side of (5) always greater than zero. We can introduce the concept of *effective buffer size*, $B_{eff}(i)$, which we define as the maximum level of buffer occupancy that the encoder

can reach at time i such that the channel rates are adequate to transport all the bits without violating the end-to-end delay constraint (i.e., without producing decoder underflow). From (5) the maximum level of encoder buffer occupancy is $\sum_{j=i+1}^{i+\Delta N_d} C$, thus we have

$$B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N_d} C \tag{6}$$

and thus we have that the effective buffer size

$$B_{eff}(i) = \Delta N \cdot C \tag{7}$$

should be constant.

We summarize this result as follows,

Formulation 2 Rate Constraints for CBR *In a CBR channel operating under end-to-end delay ΔN_d at a rate C, in order to prevent decoder buffer underflow the encoder must avoid the encoder buffer size exceeding $B_{eff} = \Delta N_d \cdot C$. It can be shown that if the encoder buffer does not exceed the maximum occupancy B_{eff}, then the decoder buffer will not exceed that maximum buffer occupancy either [17].*

Note that we refer to *effective* buffer size because, given that there is sufficient memory available at encoder and decoder, this effective memory is determined by the delay, rather than by memory considerations. As an example, assume large amounts of memory are available at encoder and decoder, then the same system could operate with different end-to-end delays ΔN_d and for each chosen value there would be a different B_{eff}.

5.2 VBR with known channel rates

Consider now the VBR transmission case where the channel rates are known. This would be the case in a VBR transmission under leaky bucket constraints where the encoder will be able to decide on the channel rate during the frame interval i, C_i, as long as it complies with the leaky bucket constraints. We can then revisit the above formulation and replace

the constant rate C by a variable one C_i. The encoder and decoder buffer occupancies are now

$$B^e(i) = \sum_{j=1}^{i} R_j - \sum_{j=1}^{i} C_j \tag{8}$$

$$B^d(i) = \begin{cases} \sum_{j=1}^{i} C_j - \sum_{j=1}^{i-\Delta N_d} R_j, & \text{when } i \geq \Delta N_d \\ \sum_{j=1}^{i} C_j, & \text{when } i < \Delta N_d. \end{cases} \tag{9}$$

where as before the decoder waits ΔN_d frame intervals before starting to decode the video frames available in its buffer. Again, we can combine the encoder buffer occupancy (8) at time i and decoder buffer occupancy (9) at time $i + \Delta N$, we have that:

$$\begin{aligned} B^d(i+\Delta N) &= \sum_{j=1}^{i+\Delta N} C_j - \sum_{j=1}^{i} R_j = \sum_{j=i+1}^{i+\Delta N} C_j - (\sum_{j=1}^{i} R_j - \sum_{j=1}^{i} C_j) \\ &= \sum_{j=i+1}^{i+\Delta N} C_j - B^e(i). \end{aligned} \tag{10}$$

so that decoder buffer underflow is prevented if the right hand side of (10) always greater than zero. And from (10) the maximum level of encoder buffer occupancy is $\sum_{j=i+1}^{i+\Delta N} C_j$, so that the effective buffer size is now,

$$B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C_j. \tag{11}$$

The main difference between this and the CBR case is that now the effective buffer size depends on the frame interval, i, and is equal to the sum of the future ΔN_d channel rates. We can guarantee that if the encoder buffer fullness $B^e(i)$ is always smaller than $B_{eff}(i)$, then the decoder buffer will not underflow.

Thus in the VBR transmission case we can have the following constraints

Formulation 3 Rate Constraints for VBR *In a VBR channel operating under end-to-end delay Δ_d at a known rate $C(i)$, in order to prevent decoder buffer underflow the encoder must prevent the encoder buffer size exceeding $B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N} C_j$.*

Note that in a "controlled" VBR channel environment the transmitter may be able to decide the number of bits to use for both source and channel. As shown in [19] this can be

done in practice and the channel rate selection can be performed so that the rates comply with some of the constraints (e.g., Leaky bucket) that were mentioned above.

5.3 VBR with unpredictable channel rates

The most important characteristic of Formulation 3 becomes apparent when one considers the case where rates are not deterministically known. There are, of course, numerous instances of this scenario, such as a reliable channel with variable delay, where the effective throughput thus varies over time, an unreliable channel with retransmission [20], or a congestion-controlled channel.

The key difficulty in dealing with these scenarios is that the encoder has to make decisions on the source coding, which are based on deterministic knowledge of the source, but somehow have to take into account the random behavior of the channel. The channel behavior is such that the data may arrive reliably to the receiver but, due to random delay or to the need for retransmission, *it may arrive at the decoder too late* to be decoded in time for display. Thus, while the encoder cannot control the channel behavior it can use whatever knowledge of the channel is available in order to avoid decoder buffer underflow.

Formulation 3 provides a simple way to formulate the desired behavior of the system. It is sufficient to guarantee that

$$B_e(i) \leq B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N_d} C_j \tag{12}$$

in order to ensure that the decoder buffer will not underflow. The future channel rates C_{i+1} through $C_{i+\Delta N_d}$ are not known at time i. However, in (12) the deterministic part of the system (the selection of source rates at the encoder) is clearly separated from the random components (the channel behavior).

If there is no knowledge about the channel very little can be done, other than perhaps assume worst case behavior. On the other hand if we have an *a priori* model of the channel and/or some online observation of its current state then it is possible to attach a likelihood of achieving a certain channel rate.

Let us call $P_i(c) = P_i(c|$ a priori channel model, channel observation), the probability (given applicable observations and prior channel model) of the available channel rate being c at time i. Note that this model could be discrete or continuous, and will depend on specific channel characteristics. An example of such a model can be found in [20], where a point-to-point wireless link based on the CDMA IS-95 standard is used. Note also that such a formulation can also be extended to the case where a random delay, rather than a random rate, has to be considered. Finally, it should be clear that the overall performance of a system under such random channel behavior will depend both on the rate control algorithms to be discussed in the next section and on the accuracy of the channel models.

Given the model $P_i(c)$ then we can define two different formulations as follows.

Formulation 4 Expected Rate Constraints *Let the system operate with delay ΔN_d and random channel rate characterized by $P_i(c)$. Then the goal is to select the encoding rate so as to meet the expected rate constraints, i.e., for each i choose the rates such that:*

$$B_e(i) \leq E[\sum_{j=i+1}^{i+\Delta N} C_j] \tag{13}$$

where $E[]$ indicates the expectation with respect to the probabilities $P_i(c)$.

It is obvious that the above formulation does not guarantee that the data will be transmitted correctly, but it is simple and it can be easily incorporated into a rate control algorithm in the form of a table that provides expected values for each possible observed channel state.

Formulation 4 lead us to a more general formulation where we take into account the likelihood of losses due to decoder buffer underflow.

Formulation 5 Minimizing Loss Probability *With the system again operating with delay ΔN_d and random channel rate characterized by $P_i(c)$, the goal is to select the encoding rate so as to meet a certain threshold in the loss probability, i.e., select the $B_e(i)$ such that:*

$$P(B_e(i) > \sum_{j=i+1}^{i+\Delta N} C_j) \leq Threshold \tag{14}$$

where the probability obviously depends on $P_i(c)$.

Examples of these two approaches can be found in [20].

5.4 Special cases

We now consider two special cases where transmission may fall in one of the categories considered above (say CBR or VBR) but where additional constraints may have to be taken into account. Without going into details we just point out that the Formulations provided before may have to be modified to address these scenarios.

5.4.1 VBR for Stored Video

When pre-encoded video is considered, we can consider two different scenarios, namely, local playback (e.g., from a DVD player) or networked playback (e.g., video on demand service to a home). The former case, as discussed earlier, involves a global allocation among the different scenes in the movie, while the channel constraints do not play much of a role (the "channel", i.e., the connection between storage device and decoder, is supposed to have sufficient bandwidth to accommodate even the highest rate scenes).

In the second, and more interesting, scenario the video sequence will have to be encoded in such a way as to meet any applicable channel constraints. This will result in different encoding results depending on whether a CBR or a VBR transmission mode is selected. In addition to these transmission-related issues, designers also need to take into account the impact of the storage medium on performance. As an example, consider the case where data is to be stored on a disk. Typically the disk will be organized into segments that will constitute the basic storage unit. Thus disk reading hardware is designed to be able to read one complete disk segment into a buffer and then move on to another segment. There will be a latency involved in moving from one disk segment to another, and since the disk is rotating, the latency will depend on the relative position of the two segments. Thus the disk placement policies will have an impact on performance. For example, if the system incurs excessive latency in some of its read operations the output buffer may empty and a decoder buffer underflow may occur.

Examples of studies of data placement on a disk for later playback can be found in [31, 32, 33, 34]. An example of work that incorporates the disk placement constraints into

the rate control algorithm can be found in [35].

5.4.2 Video Caching

Proxy caching is playing an increasing role in evolving network infrastructure. However, while proxy caching of web objects is commonplace [36] and there are several commercially available proxy caching products, the same is not true of video objects. A few examples of recent work in this area include [37, 38, 39, 40, 41].

We consider two potential benefits of proxy caching. First, it provides more efficient sharing of network resources since several users can have access to data stored locally, without needing to access it directly from the server. In particular, given that the data is available from a network node close to the user, it is possible to greatly reduce, or even remove, the initial latency in the video display. Since our system requires the decoder to wait to receive ΔN_d frames before starting to decode, we are normally constrained by the time it takes to transport these frames, and this creates an initial latency (users of audio or video over the Internet will have experienced this). Instead if all these initial frames are available from the proxy, then the delay will be less than ΔN_d frame intervals. Second, even if users do not share data, it can be shown (see [39]) that caching allows a more robust delivery of data, since it is less likely that losses (or excessive delay) will occur in transmitting data from the cache to the decoder, whereas transfer from the more "distant" (in terms of network topology) server would be less reliable.

We can roughly divide the approaches uses for video caching into four classes. First it is possible to store complete video sequences only, thus replicating parts of the content of a server in the cache. This approach is attractive in its simplicity, and indeed is the one most widely used nowadays. For example this is achieved by replicating video sequences close to the end users, so that in effect mirror sites are build that store the relevant video sequences and from which end users can directly stream sequences of interest. This mirroring has the disadvantage of not being very scalable given the relatively large sizes of the video data.

Second, it is possible to store only a prefix of the video data, i.e., all the frames that

the decoder will need to start decoding and for which it would normally wait ΔT_d seconds. If this prefix is cached as proposed in [38] all the initial latency will be absorbed without requiring that the whole sequence be stored. A third approach would allow both the prefix and a set of intermediate frames to be stored as in [37, 39]. This approach provides the same benefits of prefix caching but also adds the attractive feature of more robust delivery. The intuition is simple: if intermediate samples are stored, and these can be delivered reliably to the decoder, then the probability that the decoder buffer will "starve" is clearly reduced. Finally, it is possible to define caching strategies that rely on a scalable format for the video stream as in [41]. In this scenario, the cache will store some of the layers of the video stream so that the server only needs to provide the higher resolution layers, if the cache already contains the coarse layers. This approach is similar to the "soft caching" approach used for images in [42, 43] and preserves the video delivery at lower quality in cases where the network bandwidth is low.

It is not necessary in general to perform the encoding in a special manner because the data is going to be cached. In fact it may be preferable for the encoder to make no assumptions about whether the data will be cached. However, the problem of deciding which frames to cache (or at what resolution) is very similar to the rate control problems we will consider next, and similar techniques can be used to solve them.

6 Rate Control Algorithms

The previous section has shown how guaranteeing that the decoder always has data to decode requires the transmitted bitstream to comply with a series of rate constraints. These constraints depend on the type of transmission environment considered, including the variability, or lack of, in the transmitted rate, the end-to-end delay, etc. Thus, after a transmission environment has been chosen, the video encoder will be responsible for producing a bitstream that meets the relevant delay constraints. To do so we need a rate control algorithm at the video encoder, i.e., an algorithm to assign a quantizer to each frame or coding unit, such that the constraints are met and the video quality is good. Note that such a rate control

algorithm is needed regardless of the scenario considered (LIV, LNIV or PEV). However in the PEV case the rate selection has to be done based solely on *a priori* assumptions on the channel behavior, while in the other two cases it is possible to take into account current channel conditions to determine the rate to use at the encoder.

In this section we will provide an overview of rate control techniques that have been proposed in recent years. It is worth noting that while much of the research in video coding has been driven by the definition of widely accepted standards (such as H.263, MPEG-2, etc), these standards do not define the operation of the encoder (in fact this probably has a lot to do with their success since this approach allows companies to provide optimized encoders that provide a differentiated product while maintaining standard compatibility). Thus, rate control, which is performed at the encoder, is not part of the standard, so that many different rate control techniques can be applied while preserving standard compatibility at the decoder.

In Section 2 we outlined techniques to estimate rates at which sequences or segments within a sequence such that degradation is practically imperceptible. While the discussion focused on off-line encoding (e.g., using a two-pass algorithm) it is conceivable that approximate on-line techniques could be derived from these approaches. The next step in analyzing the performance of a system for video delivery over a network is to determine what resources are required to deliver this perceptually lossless video. The resources we are interested in are the channel characteristics, including its rate and delay performances, the delay experienced by the user before video data playback starts and the memory required at the decoder. It is because these other resources (e.g., the network bandwidth) are not abundant or free that delivering the sequence at its ideal rate may not be feasible.

6.1 Basic problem formulation

In each of the Formulations discussed in the previous section, we derived a set of constraints on the rates produced by the video encoder. Note that all of these constraints had the form of restrictions on the average rate, or the total rate for particular sets of frames. In other

words, these constraints did not specify the number of bits to be used for individual frames but rather indicated the overall rate for *a set of frames*.

Assume that we consider one such constraint, i.e., the total number of bits for a set of frames has to be less than some prescribed amount. Then the question remains of how to allocate bits among the frames. The goal in deciding this allocation should be to provide the maximum overall quality. The issue of quality is, as mentioned early in the chapter, a difficult one for video. In many cases the selected quality criterion is to minimize the average distortion over the set of frames, although approaches are also used in practice, such as the Lexicographic optimization [44] or the Min-Max criterion [45]. These two techniques seek, in different ways, to minimize the distortion of the worst frame in the set, and thus are somewhat closer to the known characteristics of the human visual system. Still, due to its relative simplicity average distortion continues to be widely used and many of the algorithms described below seek to minimize this metric.

Given that the goal of rate control is to maximize a quality measure while complying with some rate constraints, a natural way of tackle these problems is to formulate as rate-distortion (RD) optimization problems. Thus a typical formulation poses a problem of the sort: Find the bit allocation to the frames such that the overall distortion is minimized and the applicable rate constraints are met.

The overview we provide next will not try to be exhaustive. Rather, our goal is to provide examples of rate control algorithms that have been proposed in recent years. Our emphasis will be on algorithms based on rate-distortion optimization. This class of algorithms is useful even though their complexity tends to be high. RD algorithms can be used to provide benchmarks for other approaches. Moreover, one can also use them to derive approximate algorithms, where, for example, the "true" RD values are replaced by values obtained from models.

We refer the reader to [14] for an overview of rate distortion optimization techniques and their application to resource allocation problems in image and video compression. More detailed examples of the benefits of these techniques in image and video compression can be

found in [12].

6.2 CBR Algorithms

For many applications of interest Constant Bit Rate (CBR) transmission is used. When transmitting over a CBR channel, buffers at the encoder and decoder are required to smooth out the variations in the encoding rates. Buffering data requires extra memory at both encoder and decoder and introduces additional delay to data transmission. As the encoder is allowed to produce a more variable rate (more bits for difficult frames, fewer bits for easy frames, for example) the overall quality will be better. Thus larger buffers, or equivalently, increased end-to-end delay, will tend to result in higher video quality [17, 18]. Traditionally, rate control has been studied from the point of view of memory, i.e., rate control was required to avoid overflowing the available buffers at encoder and decoder. However, as in [17, 19], we have assumed that sufficient physical memory is available and formulated the problem from the point of view of end-to-end delay.

Algorithms for rate control under CBR channel conditions have been studied for years. The initial emphasis was to derive approaches that would prevent encoder buffer overflow and in the beginning these algorithms were targeted at live video and were required to be relatively simple. Early examples include [46, 47], where the emphasis is simplicity. Another example is the test model 5 [48] algorithm in MPEG-2. In this simple algorithm the rate controller defines bit rate targets for each of the frame types, and then attempts to keep the rate within the target by varying the quantization step size. Other examples of rate control include [49] as well as algorithms targeted for more recent standards, such as MPEG-4 [50, 51].

Rate-distortion based optimization techniques start with the assumption that each coding unit (e.g., each frame) can be coded with one among a finite number of coding parameters, and that each of these coding parameters will correspond to a rate and distortion pair. A typical approach consists then in measuring those RD parameters and then searching the space of all admissible solution for the one that provides the lowest distortion without

violating the rate constraints. Early examples of RD optimization algorithms include [18], [52], which deal with frame level optimization.

One issue that has to be taken into account in typical video coding algorithms is dependency, i.e., the fact that quantizer selections for some frames affect the rate distortion performance for other frames. This problem was first studied in [15]. Algorithms such as those in [53] or [16] take into account these dependencies.

Blockwise allocation algorithms are then needed to determine how to distribute the bits among the blocks in a frame. Approaches based on RD optimization include [54, 55, 56], while simple techniques such as the test model 5 quantization selection scheme are also useful if complexity is a concern.

Complexity in RD-optimized rate control algorithms is due to two factors, first the rate-distortion values at each of the operating points would have to be computed, then the best among all the available operating points would have to be found. By far the first complexity term is the most expensive since obtaining the RD data entails compressing then decompressing the data, and this may have to be done for a large number of operating points. Thus, several authors have proposed applying the RD optimization after models of the RD characteristics have been derived. Examples of this approach include [57, 58, 59].

As new standards for compression are defined there are frequently new modes of operation that were not possible with previous standards and that have to be taken into account by the rate control algorithm.

As an example, the H.263 standard allows a variable number of frames per second to be used. An example of rate control using a variable number of frames can be found in [60, 61]. There are two main difficulties in making a decision about the "ideal" frame rate for a given scene. First consider rate. For a given rate target it may be that lowering the frame rate to lower the coding rate (fewer frames to be coded) becomes counterproductive since the energy in the residue frames increases as the frame rate decreased (since frames are further apart there is less temporal redundancy). Thus there is likely to be, for a given coding rate, an optimal frame rate, in the sense of the quality of the reconstructed sequence including

error possibly the effect of frame interpolation. Second consider the issue of perceptual quality. Even if the sequence being considered does not have a very high level of motion reducing the frame rate will come at a cost of significant perceptual quality degradation, since the decoded sequence may appear to have "jerky" motion. This jerkiness can be removed to some extent by using frame interpolation techniques, in particular motion compensated frame interpolation techniques such as those found in [62].

The new MPEG-4 standard raises other interesting questions, such as methods to optimize the coding of contour information in video objects [63], where again non-standard distortion measures need to be used. Once the objects have been defined it is necessary to decide how to allocate bits among the various objects [64]. These new areas present significant challenges and are still the object of very active research.

6.3 VBR Algorithms

We will consider two different VBR transmission environments. First, in this section, we will consider situations where the encoder has the freedom to choose the transmission rate and it is assumed that the network will permit transmission at this rate, perhaps under the restriction that some channel rate constraints are met. The following section will analyze the case where the channel rates are random and will consider the benefits of performing rate control to increase robustness.

While VBR transmission has been said to produce better quality than CBR, it is often difficult to quantify the gains, as there are many parameters involved in the comparison (e.g., rate, delay, etc.). While the benefits of VBR transmission have often been touted, comparisons have sometimes ignored some of the factors. For example, as discussed in [24], while VBR transmission has been said to provide benefits in lower delay, higher video quality, and increased network utilization (statistical multiplexing gain), it is unclear that all these benefits can be achieved simultaneously. There is evidence of the potential benefits of VBR video transmission terms of perceptual quality [65], although it is difficult to quantify the exact increase in performance.

One particularly interesting and realistic scenario is that where the variable channel rates are subject to constraints, such as for example the Leaky bucket [25, 26]. In this scenario, the encoder, while able to select the channel rates, will have to ensure that these are within the pre-specified constraints. As shown by [17], and derived in a previous section, each selection of channel rates leads to different constraints for the encoder. Recent work [19, 66] has derived algorithms to optimally select the source and channel rates. For algorithms such as the Leaky Bucket, which tends to monitor the long term average transmission rate and keep it close to a given value, it can be shown [19] that the main benefit of VBR transmission is to achieve the same quality as a CBR scheme with the same long term average rate, but with a reduced end-to-end delay. Intuitively, in CBR transmission quality increases as the end-to-end delay increases, while in VBR transmission it is possible to reap the same benefit by having the rate of several frames averaged in the network, rather than in the encoder/decoder buffers.

It is worth pointing out that once rate control has been performed, as in [19], there will be many choices of channel rate that will (i) meet the desired channel rate constraints, and (ii) accommodate the optimal video quality. Thus interesting issues arise where one can explore channel rate selections that have good properties in terms of smoothness or other network based criteria. As in [67], one can assume that the source has been coded and then find the transmission rate that accommodate transmission within the delay constraints while meeting applicable rate smoothness constraints.

6.4 Real-time adaptation to channel conditions

The second VBR scenario of interest is that where the rate is random. As shown in [20], and outlined above, it is possible to use available channel information (assuming that there is a back channel) to modify the encoder's behavior. The main intuition is that the encoder should lower the rate per frame once it becomes clear that the channel bandwidth is lower. While [20] provides RD optimized solutions to Formulations 4 and 5 it is also possible to derive simpler solutions.

Clearly the success of these approaches will depend on the existence of a feedback channel, as will be seen in Section 7, but also on the application and the type of channel considered. Rate control at the encoder will only be effective if it is possible for the encoder to react to the channel changes. Thus, these techniques will be effective if (i) the channel memory is sufficiently long (e.g., when the channel remains in a given state for times that are of the order of magnitude of a few frame intervals), (ii) the end-to-end delay is longer or of the same order of magnitude as the channel memory, and (iii) the channel feedback delay is not too long.

6.5 Layered/scalable video

The foregoing discussions have assumed that rate control involves selecting coding parameters at the video encoder on a frame-by-frame or block-by-block basis. However it is worth mentioning that there is one situation where rate control is particularly easy, namely, when the bitstream has already been organized in layers, using a scalable or multiresolution encoding algorithm. This solution is attractive in its simplicity since it allows the transmitter, or an intermediate node in the network, to decide the number of layers to send given the current channel conditions. The basic idea is that the video encoder produces a base layer, which gives a coarse (relatively low quality) approximation to input sequence, and then a series of enhancement layers that successively can be used to improve the quality.

While scalable video coding has been proposed by numerous authors, and has been incoporated in the MPEG-2 standard, its widespread application has been hampered by a real or perceived reduction in performance as compared to non-scalable algorithms. This lower performance can be measured in terms of higher complexity and lower quality at the same rate as compared to a single layer algorithm. In the MPEG-2 context there are three modes of scalability, namely temporal, spatial and SNR. Temporal and spatial scalability provide low resolution coded streams that represent the input data with an approximation comprised of fewer frames or smaller size frames, respectively. For the purpose of rate control as discussed in this chapter, the SNR scalability mode (each layer represents a coarse

version of the input) seems to be better suited. It is worth mentioning that a coarse of multiresolutions coding, the so-called data partitioning approach [68], can be used for rate control.

For all its simplicity few real applications of scalable video in a communications context have been reported. One of the few examples of application of these techniques to Internet video is the proposed layered multicast approach [69], where the end clients can subscribe the number of layers (and thus overall resolution) that their bandwidth can support.

7 Conclusions: transmission issues for VBR video

To conclude this chapter we summarize the key ideas and comment on some of the issues that still remain to be addressed before true VBR video networking becomes reality. We have shown that VBR coding is the natural form of representation for video, and thus it would be desirable to transmit video end-to-end in such a way as to allow the "ideal" VBR representation of the data. We have shown that, due to real time nature of the video decoding and display, transmission has to follow certain rules so as to guarantee that transmitted data will arrive in time to be decoded, i.e., without producing decoder buffer underflow. We also made the case that the actual constraints in fact depend on the channel conditions.

There are already numerous systems in operation that allow transmission of video over constant rate channels. These require that the video coding parameters be adjusted so that the long term average transmitted rate remains constant, thus resulting in some quality degradation with respect to a purely VBR coding and transmission mode. Other environments, such as the Internet, can support variable rate transmission but provide no QoS guarantees on the rate provided.

It is likely that near term efforts in networked video transmission, and in related video compression issues, will focus on the two extreme cases of VBR transmission, namely (i) VBR transmission over channels with QoS guarantees, and (ii) transmission over lossy channels. Both environments have in common the fact that they present significant challenges for both compression and transmission.

In the case of VBR transmission with QoS guarantees, establishing conditions that will assure these guarantees is still a subject of research. Even the notion of QoS is itself somewhat misleading in that the end user of a video transmission will really care about the decoded video quality, rather than about those parameters (losses, delay jitter, etc) that are usually taken to measure the transmission quality. One question of interest that has only partially been addressed to date, includes the definition of algorithms to map levels of desired decoded video quality into combinations of networking parameters. For example, the video application may have to determine what combination of network services is needed to provide, say, broadcast quality at the receiver. As shown in this chapter, the video encoder is best placed to make decision about what coding information is most important and so it is the video encoder that can best handle the trade-offs, given any applicable network constraints. Thus, a promising direction will be in defining simple interfaces between the video server and the network that abstract the details of the operation of one from the other. The examples we discussed with a video coder optimizing its quality to match a given Leaky Bucket constraint constitutes a simple instance of this approach.

The second area where significant progress is needed, but where the potential benefits are substantial, is accessing video over a wireless link. In this case the video encoder will have to be robust to potential data losses and also be able to accommodate changes in bitrate. Here again one of the possible solutions is to introduce feedback about the state of the channel, so that the encoder can adjust its behavior depending on channel conditions.

References

[1] I. Dalgic and F. Tobagi, "Performance evaluation of atm networks carrying constant and variable bit-rate video traffic," *IEEE J. Selected Areas in Communications*, vol. 15, pp. 1115–1131, Aug. 1997.

[2] S. Gringeri, K. Shuaib, R. Egorov, A. Lewis, B. Khasnabish, and B. Basch, "Traffic shaping, bandwidth allocation, and quality assessment for mpeg video distribution over

broadband networks," *IEEE Network Magazine*, pp. 94–107, Nov/Dec 1998.

[3] M. Krunz and S. Tripathi, "Bandwidth allocation strategies for transporting variable-bit-rate video traffic," *IEEE Communication Magazine,*, pp. 40–46, Jan. 1999.

[4] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. of the IEEE*, Oct. 1993.

[5] B. Girod, "Psychovisual aspects of image communications," *Signal Processing*, vol. 28, pp. 239–251, 1992.

[6] M. W. Garrett, *Contributions Toward Real-Time Services on Packet Switched Networks.* PhD thesis, Dept. of Electrical Eng., Columbia Univ., 1993.

[7] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," Tech. Rep. 101, University of Wuerzburg, Institute of Computer Science Research Series, Feb 1995.

[8] W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, 1994.

[9] J. Mitchell, W. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard.* New York: Chapman and Hall, 1997.

[10] I. Koo, P. Nasiopoulos, and R. Ward, "Joint MPEG-2 coding fpr multi-program broadcasting of pre-recorded video," in *Proc. Intl. Conf. on Acoustics, Speech and Signal Proc., ICASS'99.*, (Phoenix, AZ), Mar. 1999.

[11] N. Duffield, K. Ramakrishnan, and A. R. Reibman, "SAVE: An algorithm for smoothed adaptive video over explicit rate networks," *IEEE/ACM Trans. on Networking*, vol. 6, pp. 717–728, Dec 1998.

[12] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, pp. 74–90, Nov. 1998.

[13] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. ASSP*, vol. 36, pp. 1445–1453, Sep. 1988.

[14] A. Ortega and K. Ramchandran, "Rate-distortion techniques in image and video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, Nov 1998.

[15] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Proc.*, vol. 3, pp. 533–545, Sept. 1994.

[16] J. Lee and B. W. Dickinson, "Rate distortion optimized frame-type selection for MPEG coding," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 7, pp. 501–510, June 1997.

[17] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 2, pp. 361–372, Dec. 1992.

[18] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. on Image Proc.*, vol. 3, pp. 26–40, Jan. 1994.

[19] C.-Y. Hsu, A. Ortega, and A. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1016–1028, Aug. 1997.

[20] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over wireless channels," *IEEE J. on Sel. Areas in Comm.*, vol. 17, pp. 756–773, May 1999.

[21] W. Verbiest, L. Pinnoo, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE J. on Sel. Areas in Comm.*, vol. 6, pp. 1623–1632, December 1988.

[22] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Trans. on Comm.*, vol. 36, pp. 834–843, July 1988.

[23] P. Sen, B. Maglaris, N. Rikli, and D. Anastassiou, "Models for packet switching of variable-bit-rate video sources," *IEEE J. on Sel. Areas in Comm.*, vol. 7, pp. 865–869, June 1989.

[24] T. V. Lakhsman, A. Ortega, and A. R. Reibman, "VBR video: Trade-offs and potentials," *Proceedings of the IEEE*, vol. 86, pp. 952–973, May 1998.

[25] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. on Sel. Areas in Comm.*, vol. 9, pp. 325–334, April 1991.

[26] L. Dittmann, S. B. Jacobsen, and K. Moth, "Flow enforcement algorithms for ATM networks," *IEEE J. on Sel. Areas in Comm.*, vol. 9, pp. 343–350, April 1991.

[27] ATM Forum, *ATM User-Network Interface Specification, Version 3.0.* Prentice-Hall, 1993.

[28] R. Jain, S. Kalyanaraman, S. Fahmy, R. Goyal, and S.-C. Kim, "Source behavior for ATM ABR traffic management: An explanantion," *IEEE Communications Magazine*, vol. 34, pp. 50–57, Nov. 1996.

[29] M. Hamdi and J. W. Roberts, "QoS guaranty for shaped bit rate video connections in broadband networks," in *Proc. of Intl. Conf. on Multimedia Networking, MmNet'95*, (Aizu-Wakamatsu, Japan), Oct. 1995.

[30] M. Hamdi, J. W. Roberts, and P. Rolin, "Rate control for VBR video coders in broadband networks," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1040–1051, Aug. 1997.

[31] E. Chang and A. Zakhor, "Disk-based storage for scalable video," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 7, pp. 758–770, Oct. 1997.

[32] E. Chang and H. Garcia-Molina, "Reducing initial latency in media servers," *IEEE Multimedia*, pp. 50–61, Fall 1997.

[33] D. W. Brubeck and L. A. Rowe, "Hierarchical storage management in a distributed VOD system," *IEEE Multimedia*, pp. 37–47, Fall 1996.

[34] S. Ghandeharizaheh, S. H. Kim, and C. Shahabi, "On configuring a single disk continuous media server," in *Proceedings of the ACM SIGMETRICS/PERFORMANCE*, May 1995.

[35] Z. Miao and A. Ortega, "Rate control algorithms for video storage on disk based video servers," in *32nd Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA), November 1998.

[36] A. Chankhunthod, P. B. Danzig, C. Neerdals, M. F. Schwartz, and K. J. Worrell, "A hierarchical internet object cache," in *USENIX Tech. Conf.*, 1996.

[37] Y. Wang, Z.-L. Zhang, D. Du, and D. Su, "A network conscious approach to end-to-end video delivery over wide area networks using proxy servers," in *Proc. IEEE INFOCOM*, (San Francisco, CA), Apr. 1998.

[38] S.Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *IEEE Infocom*, (New York, USA), March 1999.

[39] Z. Miao and A. Ortega, "Proxy caching for efficient video services over the internet," in *Proc. of Packet Video Workshop, PVW'99*, (New York, NY), Apr. 1999.

[40] S. Acharya, *Techniques for improving multimedia communication over wide area networks*. PhD thesis, Cornell University, 1999.

[41] R. Rejaie, M. Handley, H. Yu, and D. Estrin, "Proxy caching mechanism for multimedia playback streams in the internet," in *WWW Caching Workshop*, (San Diego, CA), Jun. 1999.

[42] A. Ortega, F. Carignano, S. Ayer, and M. Vetterli, "Soft caching: Web cache management techniques for images," in *1st IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, (Princeton, NJ), June 1997.

[43] J. Kangasharju, Y. Kwon, and A. Ortega, "Design and implementation of a soft caching proxy," in *3rd WWW Caching Workshop*, (Manchester, England), June 1998. Will also appear in a special issue of *Computer Networks and ISDN Systems*, Elsevier, North-Holland.

[44] D. T. Hoang, E. L. Linzer, and J. S. Vitter, "Lexicographic bit allocation for MPEG video," *Journal of Visual Communication and Image Representation*, vol. 8, Dec. 1997.

[45] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers," *IEEE Transactions on Multimedia*, vol. 1, pp. 3–17, Mar 1999.

[46] J. Zdepsky, D. Raychaudhuri, and K. Joseph, "Statistically based buffer control policies for constant rate transmission of compressed digital video," *IEEE Trans. on Comm.*, vol. 39, pp. 947–957, June 1991.

[47] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudo-constant bit rate video coding," *IEEE Trans. on Image Proc.*, vol. 2, pp. 50–59, Jan. 1993.

[48] MPEG-2, *Test Model 5 (TM5) Doc. ISO/IEC JTC1/SC29/WG11/93-225b*. Test Model Editing Committee, Apr. 1993.

[49] G. Keesman, I. Shah, and R. Klein-Gunnewiek, "Bit-rate control for MPEG encoders," *Signal Processing: Image Communication*, 1993. Submitted.

[50] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 7, pp. 246–250, Sept 1997.

[51] J. Ribas-Cordera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 9, pp. 172–185, Feb 1999.

[52] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. on Image Proc.*, vol. 3, pp. 546–558, Sept. 1994.

[53] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation among segmentation, displacement vector field and displaced frame difference," *IEEE Trans. on Image Proc.*, vol. 6, pp. 1487–1502, Nov 1997.

[54] A. Ortega and K. Ramchandran, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders," in *Proc. of SPIE, Digital Video Compression: Algorithms & Technologies 95*, (San Jose, CA), Feb. 1995.

[55] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 6, pp. 182–190, April 1996.

[56] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation between displacement vector field and displaced frame difference," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1739–1751, Dec 1997.

[57] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 8, pp. 446–459, Aug. 1998.

[58] H.-M. Hang and J.-J. Chen, "Source model for transform video coder and its application," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 7, pp. 287–311, Apr. 1997.

[59] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 6, pp. 12–20, Feb 1996.

[60] H. Song, J. Kim, and C.-C. J. Kuo, "Real-time encoding frame rate control for H.263+ video over the internet," *Signal Processing: Image Communication*, Sept 1999. To appear.

[61] H. Song, *Rate Control Algorithms for Low Variable Bit Rate Video.* PhD thesis, University of Southern California, Los Angeles, CA, May 1999.

[62] T.-Y. Kuo and C.-C. J. Kuo, "Motion compensated frame interpolation for low-bit-rate video quality enhancement," in *Proc. of Visual Comm and Signal Processing, VCIP'99*, (San Jose, CA), Jan. 1999.

[63] G. M. Schuster and A. K. Katsaggelos, "An optimal boundary encoding scheme in the rate-distortion sense," *IEEE Trans. on Image Proc.*, vol. 7, pp. 13–26, Jan. 1998.

[64] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 9, pp. 186–199, Feb 1999.

[65] M. R. Pickering and J. F. Arnold, "A perceptually efficient VBR rate control algorithm," *IEEE Trans. Image Proc.*, vol. 3, pp. 527–532, Sep. 1994.

[66] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding of video signals over ATM networks," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 1002–1015, Aug. 1997.

[67] J. Rexford and D. Towsley, "Smoothing variable-bit-rate video in an internetwork," *IEEE/ACM Trans. on Networking*, vol. 7, pp. 202–215, Apr. 1999.

[68] A. Eleftheriadis and D. Anastassiou, "Constrained and general dynamic rate shaping of compression digital video," in *Proc. of ICIP'95*, vol. III, (Washington, D.C.), pp. 396–399, 1995.

[69] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. on Sel. Areas in Comm.*, vol. 15, pp. 983–1001, Aug. 1997.