CHARACTERIZATION OF QUALITY AND TRAFFIC FOR VARIOUS VIDEO ENCODING SCHEMES AND VARIOUS ENCODER CONTROL SCHEMES

İsmail Dalgıç and Fouad A. Tobagi

Technical Report No. CSL-TR-96-701

August 1996

This work was in part supported by NSF under grant NCR-9016032 and by Pacific Bell.

Characterization of Quality and Traffic for Various Video Encoding Schemes and Various Encoder Control Schemes

İsmail Dalgıç and Fouad A. Tobagi

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Gates Bldg. 3A, Room 339,
Stanford, CA 94305

Abstract

Lossy video compression algorithms, such as those used in the H.261, MPEG, and JPEG standards, result in quality degradation seen in the form of digital tiling, edge busyness, and mosquito noise. The encoder parameters (typically, the so-called quantizer scale) can be adjusted to trade-off encoded video quality and bit rate. Clearly, when more bits are used to represent a given scene, the quality gets better. However, for a given set of encoder parameter values, both the generated traffic and the resulting quality depend on the scene content. Therefore, in order to achieve certain quality and traffic objectives at all times, the encoder parameters must be appropriately adjusted according to the scene content. Currently, two schemes exist for setting the encoder parameters. The most commonly used scheme today is called Constant Bit Rate (CBR), where the encoder parameters are controlled to achieve a target bit rate over time by considering a hypothetical rate control buffer at the encoder's output which is drained at the target bit rate; the buffer occupancy level is used as feedback to control the quantizer scale. In a CBR encoded video stream, the quality varies in time, since the quantizer scale is controlled to achieve a constant bit rate regardless of the scene complexity. In the other existing scheme, called Open-Loop Variable Bit Rate (OL-VBR), all encoder parameters are simply kept fixed at all times. The

motivation behind this scheme is to presumably provide a more consistent video quality compared to CBR encoding. In this report, we characterize the traffic and quality for the CBR and OL-VBR schemes by using several video sequences of different spatial and temporal characteristics, encoded using the H.261, MPEG, and motion-JPEG standards. We investigate the effect of the controller parameters (i.e., for CBR, target bit rate and rate control buffer size, and for OL-VBR, the fixed quantizer scale) and video content on the resulting traffic and quality. We show that with the CBR and OL-VBR schemes, the encoder control parameters can be chosen so as to achieve or exceed a given quality objective at all times; however, this can only be done by producing more bits than needed during some of the scenes. In order to produce only as many bits as needed to achieve a given quality objective, we propose a video encoder control scheme which maintains the quality of the encoded video at a constant level, referred to as Constant Quality VBR (CQ-VBR). This scheme is based on a quantitative video quality metric which is used in a feedback control mechanism to adjust the encoder parameters. We determine the appropriate feedback functions for the H.261, MPEG, and motion-JPEG standards. We show that this scheme is indeed able to achieve a constant quality at all times; however, the resulting traffic occasionally contains bursts of relatively high-magnitude (5-10 times the average), but short duration (5-15 frames). We then introduce a modification to this scheme, where in addition to the quality, the peak rate of the traffic is also controlled. We show that with the modified scheme, it is possible to achieve nearly constant video quality while keeping the peak rate within 2-3 times the average.

Key Words and Phrases: Video Encoding, Constant Bit Rate (CBR), Variable Bit Rate (VBR), Constant Video Quality, Video Traffic Characterization, Video Quality Characterization, Feedback Control, H.261, MPEG, JPEG.

1 Introduction

In order to achieve high compression rates, today's prominent video encoding standards, such as H.261, MPEG, and motion-JPEG [1, 2, 3], are based on lossy video compression algorithms. Such loss results in digital tiling, edge busyness, and mosquito noise [4] in the encoded video. The encoder parameters (typically, the so-called quantizer scale) can be adjusted to trade-off encoded video quality and bit rate. Clearly, when more bits are used to represent a given scene, the quality gets better. However, for a given set of encoder parameter values, both the generated traffic and the resulting quality depend on the scene content. Therefore, in order to achieve certain quality and traffic objectives at all times, the encoder parameters must be appropriately adjusted according to the scene content.

Most of the existing video encoders are controlled according to the *Constant Bit Rate (CBR)* feedback control scheme, where the rate of the encoded video is kept constant at a target rate V at all times by dynamically adjusting the quantizer scale. CBR encoding is motivated by the fact that some communications technologies, such as ISDN, as well as some storage technologies, such as CD-ROMs, are able to accommodate only constant bit rate streams.

The CBR video encoder control scheme works as follows. The bits produced by the encoder are assumed to be placed in a hypothetical rate control buffer which is drained at rate V; the quantizer scale at a given time is then selected proportionally to the rate control buffer occupancy divided by the buffer size. It is important to note that in a CBR encoded video stream, the quality varies in time, since the quantizer scale is controlled to achieve a constant bit rate regardless of the scene complexity. For example, consider that while a video sequence is being encoded, at some point the amount of motion in the video increases. Then, the number of bits produced for the current value of the quantizer scale increases, which causes an increase in the buffer occupancy. As the buffer occupancy increases, the quantizer scale also increases, until the bit rate reduces down to V again. Thus, at steady state, the quantizer scale is greater for more complex scenes, and as a result the quality is likely to be lower for such scenes. For some scenes, the amount of motion may be so large that even at the maximum allowed quantizer scale, the bit rate produced may exceed V. In that case, the rate control buffer overflows; this causes some video information to be dropped, causing even greater quality degradation.

Many networking technologies such as LANs and ATM networks can support variable bit rate traffic by means of statistical multiplexing. Therefore, when video is to be transmitted over such a network, one can use variable bit rate encoding in order to provide a more consistent level of quality compared to CBR. For this purpose, many have considered *Open-Loop Variable Bit Rate (OL-VBR)* encoding, whereby the quantizer scale is simply kept at a constant value at all times. With OL-VBR encoding, a more complex scene is encoded using more bits; thus, the quality is indeed less variable in time compared to CBR encoding. Nevertheless, it can be shown that there are still variations in quality.

Since the quality varies with content in both CBR and OL-VBR schemes, if a minimum level of quality is to be attained at all times, then some scenes would be encoded using more bits than needed. Moreover, in the absence of a priori information about the video content, one cannot determine the smallest possible values of the encoder control parameters; thus, conservative values would have to be used, which would further result in excess bits produced. Clearly, in order to produce only as many bits as needed to achieve a given quality objective at all times, the video must be encoded at a constant level of quality. It is possible to achieve constant quality video encoding if one were to use a quantitative video quality measure and a feedback control mechanism to adjust the encoder parameters. In this report, we introduce such a scheme, referred to as *Constant-Quality VBR (CQ-VBR)*. We characterize the quality and traffic for the CBR, OL-VBR and CQ-VBR schemes for several categories of video content, namely, videoconferencing (i.e., head-and-shoulders), motion pictures, and commercial advertisements. We also characterize the delay in the source when video is transmitted over a circuit. We use several video sequences with different spatial and temporal characteristics, encoded using H.261, MPEG-1, and motion-JPEG standards [1, 2, 3]. In order to characterize the quality of the existing schemes, as well as to devise a scheme for encoding video at a constant quality, a quantitative video quality measure is required. We use in our study such a measure that has been developed at the *Institute for Telecommunication Sciences (ITS)*[5].

Note that an important goal behind characterizing the quality and traffic for video sources is to evaluate the performance of networks carrying such video traffic. Such an evaluation is a complex topic which cannot be contained within the scope of this report. Therefore, here we only give some preliminary results, and treat the topic fully elsewhere [6, 7].

The remainder of this report is organized as follows. In Section 2, we describe the prior work in video traffic characterization. In Section 3, we describe the system under consideration, mainly focusing on the video encoder. In Section 4, we describe the ITS video quality measure. In our evaluation of the encoder control schemes, we use 5 video sequences with different spatial and temporal characteristics, each of them several minutes long. We describe in Section 5 those video sequences, and how they are encoded. In Section 6, we describe the CBR scheme in more detail, and examine the traffic, delay, and quality characteristics as a function of the target bit rate, the rate control buffer size, and video content. In Section 7, we characterize the traffic, delay, and quality for the OL-VBR scheme as a function of the quantizer scale and video content. We show that for both CBR and OL-VBR schemes, the video content indeed has a significant effect on the resulting quality, thereby motivating the need for a scheme which achieves constant video quality. In Section 8, we characterize the traffic and quality for the CQ-VBR scheme. We show that the CQ-VBR scheme can maintain a given quality objective while producing fewer bits than the existing schemes; however, the resulting traffic occasionally contains bursts of relatively high magnitude (5-10 times the average), but short duration (5-15 frames). We then describe a modification to this scheme where in addition to the quality, the peak rate of the traffic is also controlled. We refer to this scheme as Joint Peak Rate and Quality Controlled VBR (JPQC-VBR). We show that with the JPQC scheme, it is possible to achieve near-constant video quality while keeping the peak rate within 2-3 times the average rate. Finally in Section 9, we present our concluding remarks.

2 Prior Work on Video Traffic and Quality Characterization

There is a great deal of prior work on traffic characterization and modeling of variable bit rate video. Most of this work is focused on Open-Loop VBR. Usually the approach is to report the traffic statistics such as the histogram and autocorrelation functions, and peak, average, and standard deviation values of the number of bits per frame. Then, models which fit these statistics are devised. A brief description of each of these studies is as follows.

In [8] three H.261 encoded OL-VBR videoconferencing sequences are studied. The total

length of the sequences is 20 minutes. The video traffic statistics given are the average and peak frame sizes. The peak-to-mean ratios are shown to be in the range of 4 to 10 for the three sequences. In addition, the effect of smoothing on the bit rate is examined, and the peak bit rates over a smoothing interval of 4 frames are given. When smoothing is applied, the peak-to-mean ratios are shown to vary from 1.8 to 3.5.

In [9], also an OL-VBR encoded videoconferencing sequence is examined. The length of the sequence is 30 minutes, and the sequence is encoded using a proprietary encoder. The histogram and autocorrelation of frame sizes are given, and models are proposed to match the observed statistics. One important observation made in this paper is that for videoconferencing-type sequences, the number of bits per frame is a *stationary* stochastic process.

In [10], several sequences of the broadcast-video type are encoded using a proprietary video encoder which employs interframe prediction. It is shown that the video content has an important effect on the resulting statistics to the degree that it does not seem possible to devise a single model which is valid for all the sequences they considered.

In [11], several 2-minute sequences are OL-VBR encoded using an MPEG-1 encoder. The sequences considered include several excerpts from motion pictures, one from a boxing match, and one news clip. The effect of the quantizer scale and the video content on the resulting traffic statistics is investigated. It is shown that the average data rate varied between 2.1 Mb/s and 7.8 Mb/s. However, the peak-to-average ratio of the frame sizes were always around 2–3.

In [12], two 10-minute sequences (a news clip and an advertisements sequence), which are OL-VBR encoded using MPEG-1, are examined. The effect of smoothing the sequences over a given time interval is studied. It is shown that the video content affects greatly the peak-to-average ratios, even after smoothing.

In [13], a 23-minute excerpt from the movie "The Wizard of Oz" is OL-VBR MPEG-1 encoded. Individual statistics for the I, P, and B frames are given. They show that it is easier to devise models individually for each type of frame. In [14] five short test sequences are OL-VBR MPEG-1 encoded. In addition to the traffic statistics, also the SNR statistics for the sequences are shown for different values of the quantizer scale. The SNR values are compared for the OL-VBR and CBR sequences at the same average rate, and it is shown that the OL-VBR sequence has a consistently better SNR.

In [15], 14 video sequences, each of them 30 minutes long, are OL-VBR encoded using MPEG-1. Frame size distibutions for the I, P, and B frames, as well as autocorrelation functions are given for the frame sizes and Group-Of-Pictures (GOP) sizes (where a GOP is the collection of all the frames from one I frame up to the next I frame). It is shown that frame size distributions for a given frame type follow either the Gamma or Lognormal distribution. However, the parameters of the distribution vary from sequence to sequence. It is also shown that the frame-size and GOP-size autocorrelation functions vary from sequence to sequence, and a single model cannot be used to match all sequences. Even for the sequences of the same category (i.e., movies, or cartoons), the statistical properties differ significantly.

In [16], traffic statistics are given for a whole movie which is OL-VBR encoded using an MPEG-2 encoder. Traffic statistics with and without motion compensation are compared. It is shown that when no motion-compensation is employed (i.e., only I frames are used), the peak-to-mean ratio was equal to 2.6; with forward motion compensation (i.e., using I and P frames), it was equal to 7.6, and with both forward and backward motion-compensation (i.e., using I, B, and P frames), it was equal to 6.6.

Finally, in [17], a whole movie (Star Wars) is OL-VBR encoded using a proprietary intraframe encoder. It is shown that the frame sizes exhibit a long-range dependence, and the frame size distribution is heavy-tailed.

It is important to note that the usage of different encoding schemes, different video sequences, and different operating modes of the encoders used in these studies make it very difficult to compare the results of one with the other. Furthermore, with the exception of [14] (which used SNR), the others did not characterize the video quality.

What differentiates our work from the prior work is as follows. First, in addition to the traffic, we also characterize the *delay* and *quality* for various video encoder control schemes. In order to evaluate and compare different video encoder schemes, we provide a consistent framework by using several video sequences of different characteristics, which are encoded using common video encoding standards. Furthermore, we propose new video encoder control schemes where the objective is to maintain the video quality at a constant level by means of feedback control.

3 System Description and the Identification of the End-to-End Delay Components

In Figure 1, the block diagram of the system under consideration is shown. A frame is scanned by the video camera, and the resulting analog signal is sent to the digitizer. The data produced by the digitizer is then encoded by a video encoder, whose parameters are controlled according to a specific control algorithm. The bits produced by the encoder are then given to the host, which transmits them over the network to the receiving station, where the video is decoded and displayed. In the following, we describe each component in the system in more detail, explaining its operation, and identifying its contribution to the end-to-end delay.

In Section 3.1, we describe the digitization process of the video signal. In Section 3.2, we describe the video encoding process, discuss the specifics of the H.261, MPEG-1, and motion-JPEG standards, and identify the delays due to video encoding. In Section 3.3, we describe the delays due to the packetization and network. In Section 3.4, we describe the operation of the decoder and the display, and discuss the delays incurred therein.

3.1 The Video Signal and its Digitization

An analog video signal consists of a number of frames, generated at a certain rate F^{analog}. During one frame period, the video camera scans the frame line by line. For NTSC, the number of lines per frame (N_{lines}^{analog}) is equal to 455, and F^{analog} is equal to 30 frames per second (fps); for PAL, N_{lines}^{analog}= 525, and F^{analog}=25 fps.

The analog video signal is passed to a digitizer in real time, without any delay. The digitizer samples and quantizes the analog signal also in real-time (hence this process also involves no delay). Each sample thus created corresponds to a pixel. We let $N_{p/l}$ denote the number of pixels per line, and $N_{p/c}$ denote the number of pixels per column. (Note that $N_{p/c}$ is not necessarily equal to N_{lines}^{analog}; it can be made greater by means of interpolation, or smaller by means of decimation. Typically, the effective vertical resolution of an analog video signal is about one half of N_{lines}^{analog} as a result of effects such as motion break-up, aliasing, and Kell factor [18]. Thus, usually, $N_{p/c}$ is chosen to be smaller than N_{lines}^{analog}. Similarly, $N_{p/l}$ is not necessarily equal to the number of samples per line which would be

attained by sampling the signal at the Nyquist rate; it can be less, or if interpolation is used, greater.) The result is a $N_{p/l}$ x $N_{p/c}$ matrix of pixels for each frame. Three digital frame formats are commonly used: (i) CIF ($N_{p/l}$=352, $N_{p/c}$=288), (ii) SIF ($N_{p/l}$=352, $N_{p/c}$=240), and (iii) QCIF ($N_{p/l}$=176, $N_{p/c}$=144). In all three formats, the image is divided into 3 components: a luminance component, and two chrominance components. Since the human eye is less sensitive to the color of an image compared to its intensity, the chrominance components are subsampled at half the resolution in both horizontal and vertical dimensions. For both the luminance and chrominance components, 8 bits are used per sample. Note that, as a result of the digital sampling and quantization, there will be some degradation of quality in the digital signal with respect to the analog signal; however, we ignore this effect.

The pixels produced by the digitizer are passed to the encoder for encoding. In the worst case, the digitizer may accumulate all the bits corresponding to a frame before passing them to the encoder, in which case the transfer delay from the digitizer to the encoder will be $D_{transfer}^{dig-to-encoder}=1/F^{analog}$ (i.e., 33 ms for F^{analog}=30 fps). On the other extreme, the digitizer may pass the data to the encoder as soon as the smallest unit of information that the encoder can operate on is digitized. For DCT-based encoders (such as those considered in this study), that unit of information is a group of 16x16 pixels referred to as a *macroblock*; in that case, the digitizer could pass the information to the encoder in groups of 16 lines. Then, $D_{transfer}^{dig-to-encoder}=16/(N_{p/c}F^{analog})$, (e.g., for F^{analog}=30 fps, and CIF resolution, this delay is equal to 1.8 ms). In order to keep the end-to-end delay small, we suggest and consider the latter case.

3.2 Video Encoding

We consider that video is encoded according to any of the three standards, H.261, MPEG, or motion-JPEG. In this section, we first provide a brief description of these standards, and then discuss the delays associated with video encoding.

All three of these standards are based on Discrete Cosine Transform (DCT). In the encoder, a 16x16 block of samples in the luminance component is divided into 4 8x8 blocks, and the DCT is applied individually on each block. The DCT is also applied on the corresponding 8x8 block in the two chrominance components. The group of those six blocks are referred to as a *macroblock*; denoting the number of macroblocks in a frame by

M, we have M=396 for CIF, M=330 for SIF, and M=99 for QCIF.

The DCT coefficients are quantized by using an 8x8 quantization matrix. The elements of this matrix correspond to the quantization step size to be used for each DCT coefficient. The value of the DCT coefficient is divided by the quantization step size and rounded to the nearest integer. The quantization matrix is obtained by multiplying the coefficients of a "base" matrix by the quantizer scale q. Quantization is the only lossy step in the DCT-based video encoding process. Clearly, larger values of quantization step sizes correspond to coarser quantization, and hence, greater degradation in the quality (of course, smaller number of bits produced as well). As a result of the quantization, many of the DCT coefficients become zero, particularly at higher frequencies for typical scenes. Therefore, the zero DCT coefficients are run-length encoded. The non-zero coefficients are variable-length encoded, using fewer bits to represent coefficients which are more likely to occur.

Typically, there exist a strong correlation between successive frames of a video sequence. H.261 and MPEG encoding schemes make use of such temporal correlations in order to further compress the data by differentially encoding a macroblock with respect to another frame. A macroblock is said to be *intracoded* if it does not depend on the previous or the next frames. By contrast, if a macroblock is differentially encoded with respect to another frame, it is said to be *intercoded*.

We now examine the specific features of the H.261, MPEG, and Motion-JPEG video encoding standards.

A. Video Encoding Standards

a) H.261

The ITU-T Recommendation H.261 (also referred to as $p\times64$) [1] specifies a video encoding and decoding scheme for videophone, videoconference and other audiovisual services; this recommendation is conceived for sending video over circuit-switched links at the rates of $p\times64$ kbits/s, where p is an integer in the range 1 to 30. However, the techniques described therein are not limited only to circuit-switched networks, and may be applied in packet switched networks as well.

In H.261, a macroblock can be either intracoded, or intercoded with respect to the *preceding* frame. When a macroblock is to be intercoded, typically a "motion search" is performed to find the 16x16 area in the previous frame which best matches the macroblock

currently being encoded, and the macroblock is then differentially encoded with respect to that area. Clearly, on average, the intracoded macroblocks are encoded using more bits compared to the intercoded macroblocks. In H.261 the decision of intra- vs. intercoding of a macroblock is left up to the implementation. In any case, in the first frame of a video stream, all the macroblocks must always be intracoded, since there is no previous frame to take as reference for intercoding. Furthermore, when considering transmitting the video stream on a network where packets may be lost, some portion of the video data must be intracoded periodically in order to reconstruct the video signal at the receiver within a finite period of time after some loss occurs. One possible method for this is to intracode all macroblocks in one frame out of every K frames, and intercode all macroblocks in all the other frames. Another method is to intracode a fraction of each frame (other than the first one, which is fully intracoded), cyclically changing the intracoded region from frame to frame. In H.261, macroblocks are combined into 11x3 groups called a *Group of Blocks (GOB)*; typically, the portion of a frame that is intracoded is a single GOB. With the first approach, the intracoded frames will take significantly more bits than the intercoded frames; thus, the resulting traffic will be more bursty compared to the second approach. For that reason, typically the second approach is used for H.261 encoding, and here we take that approach as well.

In H.261, the base quantization matrix is $[2]_{8\times 8}$; thus, all the DCT coefficients of a block are quantized using the same quantization step size. The quantizer scale q can be specified on a macroblock by macroblock basis, and ranges from 1 to 31.

b) MPEG

MPEG (Moving Pictures Experts Group) is a standardization body under ISO (the International Standards Organization) that generates standards for digital video and audio compression. The first video compression standard devised by the MPEG group is intended for VCR quality video, using the SIF frame format, and a bit rate up to about 1.5 Mb/s. This standard is referred to as MPEG-1 [2]. The next MPEG video compression standard is MPEG-2 [19]. It has similar concepts to MPEG-1, but includes extensions to cover a wider range of applications. MPEG-2 introduces several enhancements over MPEG-1, such as support for interlaced video, scalability, low-delay mode of operation, increased DCT DC precision, non-linear quantization, new VLC tables, etc. The primary application targeted

during the MPEG-2 definition process was the all-digital transmission of broadcast TV quality video at coded bitrates between 4 and 9 Mbit/sec. However, the MPEG-2 syntax has been found to be efficient for other applications such as those at higher bit rates and sample rates (e.g. HDTV). In this report, we focus on MPEG-1, leaving the traffic and quality characterization of MPEG-2 encoded sequences for future work.

In both MPEG-1 and MPEG-2, in addition to a past frame, a future frame may also be used as reference for an intercoded macroblock. Furthermore, frames are divided into three types: (i) intracoded frames (I frames), which contain only intracoded macroblocks, (ii) predictive-coded frames (P frames), which can contain intracoded macroblocks, as well as intercoded macroblocks that use the nearest preceding I or P frame as reference, and (iii) bidirectionally predictive-coded frames (B frames), which can contain the macroblock types found in P frames, as well as intercoded macroblocks that use either the preceding, the following, or both of the I or P frames as reference. B frames are never used as reference by other frames. A number of frames is organized to form a *Group of Pictures (GOP)*, which always starts with an I frame, and contains a number of P and B frames. The GOP structure in MPEG has an important effect on the resulting traffic, as well as delay. In particular, when B frames are used, the encoding of the B frames are delayed until the subsequent I or P frame is encoded; similarly, the decoding and displaying of a B frame is also delayed until the subsequent I or P frame is decoded. Therefore, when the application requires a low end-to-end delay, the B frames are not used. In this report, we have consider two GOP structures for MPEG: (i) *IBBPBBPBBPBBI...* for non-interactive applications, and (ii) *IPPPPPPPPPPI...* for interactive applications. We refer to these GOP structures as *GS1* and *GS2*, respectively.

In MPEG-1, there are two base quantization matrices, one for the intracoded macroblocks, and one for the intercoded macroblocks. The quantization matrices may take either default values, or they may be uploaded at the beginning of the video sequence. The default base quantization matrix for the intercoded macroblocks is the same as in H.261. The default base quantization matrix for the intracoded macroblocks (denoted as Q_{intra}) is shown in Equation 1 below. (This matrix has been specified as one of the default quantization matrices for JPEG, and later on adopted by MPEG-1). For both intra- and intercoded macroblocks, again a quantizer scale q (in the range of 1 to 31, just as in H.261) is specified on a macroblock by macroblock basis.

$$Q_{intra} = 1/8 \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \quad (1)$$

c) Motion-JPEG

The motion-JPEG scheme is a straightforward adaptation of the still-image encoding standard JPEG [3] into moving pictures, whereby every frame in the video sequence is JPEG encoded independently of other frames. Therefore, in motion-JPEG, the temporal correlation among successive frames is not exploited.

In motion-JPEG, a quantization matrix can be specified on a frame by frame basis. The JPEG syntax does not include a quantizer scale; the elements of the quantization matrix are directly used as the quantization step sizes. However, one can define a frame-level quantizer scale by again starting with a base quantization matrix, and scaling it to determine the quantization matrix to be used in the current frame. In the JPEG encoder that we are using in this study [20], such an approach is used. The default quantization matrix used is the one shown in Equation 1, except that the multiplier at the beginning is 1/50 instead of 1/8. (Therefore, a quantizer scale of 50 in JPEG is equivalent to a quantizer scale of 8 in MPEG-1 I frames.) There is no specified upper limit for the quantizer scale. Here we consider a range from 1 to 500 (values greater than 500 result in a very large distortion of the image).

B. Video Encoding Delay

We consider that the bits produced by the encoder are placed in the encoder output buffer at regular time intervals of T_e seconds, and we denote the number of bits produced at the i'th time interval by m_i. Note that a macroblock is the smallest unit of data which can be encoded without any further information; therefore, T_e must be a multiple of macroblock

times. Furthermore, since the encoder operates in real time, it has to be able to keep up with the incoming data. This requires that the number of macroblocks produced in one time interval (denoted by N_m) must satisfy the equality $N_m = T_e FM$ where F denotes the rate of frames produced. Let the encoding delay for encoding information corresponding to a macroblock, say k, be $D_e(k)$ (measured from the time when all the bits corresponding to the macroblock are passed to the encoder, to the time they are encoded, and the resulting bits are placed at the output of the encoder). If macroblock k is the first one among a given group of macroblocks placed at the output of the encoder, then $D_e(k) = T_e$. For all the other macroblocks, D_e is less than T_e. In order to keep $D_e(k)$ at a minimum, one may streamline the encoder such that it operates on a macroblock-by-macroblock basis; in this case, letting $\tau \triangleq 1/FM$, $T_e = \tau$ (e.g., $T_e \approx 0.1$ ms for SIF resolution and for F=30 frames per second). On the other hand, if the encoder operates on a frame-by-frame basis, then T_e=33.3 ms for the same frame resolution and the same frame rate. In this report, we consider that the encoder operates on a macroblock by macroblock basis in order to keep the end to end delay at a minimum.

Note that we consider the encoder parameters to be controlled in real time, and based only on past information. Under these conditions, the particular choice of encoder control scheme has no effect on the encoding delay.

For MPEG, when B frames are used, an extra delay is incurred in the encoder in addition to T_e. This delay is equal to the number of consecutive B frames times the frame interval, because the encoding of the B frames have to be delayed until the subsequent P or I frame is encoded.

3.3 Packetization and Network Delay

The bits produced by the encoder are placed in the encoder's output buffer, from where they are retrieved by the host for transmission over the network. If the network is of the packet switching type, the bits are retrieved in blocks that correspond to the payloads of the packets; if the network is of the circuit switching type, again the bits are retrieved in blocks which are placed in frames and sent over the circuit.

The delays in a packet switched network depend strongly on the packetization process, and the network and traffic scenarios under consideration. We address this topic elsewhere [6, 7]. In this report, we consider the case where the network is of circuit switching

type.

Consider that a single video stream is sent over a circuit with bandwidth C, and that there is no framing; i.e., the bits placed in the encoder output buffer are immediately available for transmission over the circuit. If $m_i > CT_e$ for the i'th time interval, then the excess bits are buffered, incurring some delay. We denote the delay incurred by a macroblock k in the encoder output buffer by $D(C,k)$ (defined from when the bits corresponding to macroblock k enter the encoder's output buffer, until they are taken out of the buffer). It is this delay that we will focus on in this report, since it is the only component of the source delay that depends on the generated traffic. (The delay in the circuit is equal to the propagation delay, which is constant over time.)

Note that for the parts of the video sequence where there are underflows in the encoder output buffer, choosing a larger T_e may prevent some underflows, and thus cause a decrease in $D(C,k)$ for the subsequent macroblocks. However, the maximum value of $D(C,k)$ is likely to be independent of T_e, unless that maximum is very small; this is because when the maximum D^s is reached, the encoder output buffer is not likely to underflow. In the following sections, we show that this is indeed the case.

3.4 Decoder and Display

At the receiver, the variations in delay should be removed, so as to be able to playback the video stream in a continuous fashion. In order to accomplish this, the clocks in the sender and the receiver are synchronized (e.g., using a protocol such as NTP [21]); the encoder time-stamps each macroblock, and the decoder buffers each received macroblock so that the delay from the output of the encoder to the output of the playback buffer is equal to the end-to-end delay bound D_{max} minus the delays due to the decoding and display[1].

Let $D_{dec}(k)$ be the delay from when all the bits corresponding to macroblock k is placed in the decoder until it is decoded and ready to be displayed. Similarly to the encoder, we assume that the decoder is streamlined and fast, so that it operates on a macroblock-by-macroblock basis, and decodes and outputs each macroblock in a time equal to $1/FM$.

[1]If a group of macroblocks are received in a packet, there is no reason to pass them to the decoder in smaller groups; thus, all the macroblocks belonging to the same packet are passed to the decoder as a single group, when the delay of the first macroblock in the packet reaches D_{max} minus the decoding and display delays.

Under that condition, $D_{dec}(k)$ is negligible. If the decoder were to operate on a frame-by-frame basis, it would decode and output each frame in a time equal to $1/F$; thus, $D_{dec}(k)$ would be equal to $1/F$.

When the video is MPEG-encoded using B frames, then an extra delay of $1/F$ must be added to the D_{dec}, since the subsequent P or I frame must be decoded and stored before the current B frames can be decoded. (Along with the increase in the encoder delay, the B frames therefore cause an increase in the end-to-end delay equal to the frame interval multiplied by the number of successive B frames in a GOP plus one. For example, for F=30 frames per second, and for a GOP structure of IBBPBBPBB... , the extra delay caused by the encoding/decoding of B frames is equal to 100 ms.)

Let the delay from when a macroblock k is decoded until it is displayed be $D_{disp}(k)$. If there is no synchronization between the decoder and the display, then D_{disp} takes a value between 0 and 33 ms, depending on the timing relationship between the display scanning and the placement of the macroblocks in the frame buffer. On the other hand, if the decoder and the display are synchronized such that the display scanning begins when the first line of macroblocks is decoded, then D_{disp} will be equal to the decoding time of one line of macroblocks, (i.e., 1.8 ms for CIF, 2.2 ms for SIF, and 3.6 ms for QCIF).

4 ITS Quantitative Video Quality Measure

A quantitative video quality measure has been designed at the Institute for Telecommunication Science (ITS) that agrees closely with quality judgments made by a large number of viewers [5]. To design this measure, the authors first conducted a set of subjective tests in accordance with CCIR Recommendation 500-3 [22]. The viewers were shown a number of original and degraded video pairs, each of them 9 seconds long, and they were asked to rate the difference between the original video and degraded video as either imperceptible (5), perceptible but not annoying (4), slightly annoying (3), annoying (2), or very annoying (1). The video impairments used in those tests included digital video compression systems operating at rates around 700 kb/s and lower.

As described in [5], the quantitative measure $\hat{s}$ is a linear combination of three quality impairment measures. Those three measures were selected among a number of candidates such that their combination matched best the subjective evaluations. The correlation

coefficient between the estimated scores and the subjective scores was 0.94, indicating that there is a good fit between the estimated and the subjective scores. The standard deviation of the error between the estimated scores and the subjective scores was 0.4 impairment units on a scale of 1 to 5; thus, the subjective interpretation of a quality estimate given by $\hat{s}$ is not more accurate than ± 0.4 units. However, we have observed that for a given video sequence, a difference of 0.2 units in $\hat{s}$ is subjectively noticeable; therefore, when comparing various encoding schemes for the same sequence, we consider a difference of 0.2 units to be meaningful.

The three measures are based upon two quantities, namely, *spatial information* (SI) and *temporal information* (TI). The spatial information for a frame F_n is defined as

$$SI(F_n) = STD_{space}\{Sobel[F_n]\},$$

where STD_{space} is the standard deviation operator over the horizontal and vertical spatial dimensions in a frame, and $Sobel$ is the Sobel filtering operator, which is a high pass filter used for edge detection [23].

The temporal information is based upon the motion difference image, ΔF_n, which is composed of the differences between pixel values at the same location in space but at successive frames (i.e., $\Delta F_n = F_n - F_{n-1}$). The temporal information is given by

$$TI[F_n] = STD_{space}[\Delta F_n].$$

Note that SI and TI are defined on a frame by frame basis. To obtain a single scalar quality estimate for each video sequence, SI and TI values are then time-collapsed as follows. Three measures, m_1, m_2, and m_3, are defined, which are to be linearly combined to get the final quality measure. Measure m_1 is a measure of spatial distortion, and is obtained from the SI features of the original and degraded video. The equation for m_1 is given by

$$m_1 = RMS_{time}(5.81 \left| \frac{SI[O_n] - SI[D_n]}{SI[O_n]} \right|),$$

where O_n is the n^{th} frame of the original video sequence, D_n is the n^{th} frame of the degraded video sequence, and RMS denotes the root mean square function, and the subscript *time* denotes that the function is performed over time, for the duration of each test sequence.

Measures m_2 and m_3 are both measures of temporal distortion. Measure m_2 is given by

$$m_2 = f_{time}[0.108 MAX\{(TI[O_n] - TI[D_n]), 0\}],$$

where $f_{time}[x_t] = STD_{time}\{CONV(x_t, [-1, 2, -1])\}$, STD_{time} is the standard deviation across time (again, for the duration of each test sequence), and $CONV$ is the convolution operator. The m_2 measure is non-zero only when the degraded video has lost motion energy with respect to the original video.

Measure m_3 is given by

$$m_3 = MAX_{time}\{4.23LOG_{10}(\frac{TI[D_n]}{TI[O_n]})\},$$

where MAX_{time} returns the maximum value of the time history for each test sequence. This measure selects the video frame that has the largest added motion. This may be the point of maximum jerky motion or the point where there are the worst uncorrected errors.

Finally, the quality measure $\hat{s}$ is given in terms of m_1, m_2, and m_3 by

$$\hat{s} = 4.77 - 0.992m_1 - 0.272m_2 - 0.356m_3.$$

Note that the definition of $\hat{s}$ given above implies that the quality of each test sequence is represented by a single number. This is appropriate for short sequences, such as those used in the ITS experiments. However, for long sequences, which would most likely contain multiple scenes, it is more meaningful to measure the quality for short time intervals, therefore capturing the quality variations over time. In the sequences we use, there are several scenes which are only one or two seconds long. The interval for measuring the quality should also be chosen large enough to correspond to the response time of the human visual system. With these considerations in mind, in this report we measure the quality in one-second intervals.

Note that many researchers have considered Signal-to-Noise Ratio (SNR) as a quantitative video quality measure. SNR for frame n is defined as

$$\mathrm{SNR}(n) = 10 \log_{10} \frac{\sum_{i=1}^{N_p} o_i^2(n)}{\sum_{i=1}^{N_p} (o_i(n) - d_i(n))^2},$$

where $o_i(n)$ and $d_i(n)$ are the luminance values for the i'th pixel of the n'th original and encoded frames, respectively, and N_p is the number of pixels in a frame.

As illustrated in the following sections, SNR does not capture well the quality degradations due to digital video compression. When comparing the relative quality for the same video content compressed in different ways, SNR usually provides the correct ranking. However, the problem with SNR is that there is no one-to-one mapping between the

absolute magnitude of SNR and perceived quality; such a mapping depends highly on the video content. Thus, using SNR it is not possible to determine whether the degradations in an encoded sequence are acceptable or not. As an example, consider the following two example video contents: (i) a scene consisting of some text displayed on the screen against a flat background, and (ii) another scene consisting of a view of a flower garden. In the first scene most of the pixels will constitute the flat background, which can be encoded using very few bits without introducing any significant error. Thus, the SNR for such a case would be very high, even when there are severe distortions in the text characters being displayed due to coarse quantization. By contrast, in the second scene, there are lots of irregular, small patterns; even though the encoded video may contain distortions, they may be not be perceived due to such irregularity in the content. Thus, the SNR for such a scene may be low, while the quality degradations may not be very perceivable.

5 Evaluation Scenarios

Our numerical results are obtained using five different video sequences. Three of these sequences are taken from motion pictures: (i) Star Trek VI: The Undiscovered Country, (ii) Indiana Jones: Raiders of the Lost Ark, and (iii) Terminator-2. The Star Trek sequence is 9 minutes long, and the Raiders and Terminator-2 sequences are each 4 minutes long.

The Star Trek sequence contains a combination of fast action scenes and other slower-moving scenes; particularly difficult to encode are some scenes where there is a lot of irregular camera shaking. Moreover, in that sequence, the scenes are very short, averaging about 2 seconds per scene.

The Raiders sequence starts slowly, with a shot of Indiana Jones hiding in a hill looking over a group of people. Then the scenes speed-up: Indiana Jones starts riding a horse, chasing a group of soldiers; at that part of the scene, there is a lot of camera panning, which sometimes gets very fast, and includes forward camera obstruction. Here too, the scenes are quite short, on average 3 seconds. However, the variations in content from one scene to the next are not as drastic as in the Star Trek sequence.

The Terminator-2 sequence does not contain as much motion as the other two sequences. It comprises a mixture of real and synthetic images (with relatively sharp edges), where the terminator T1000 changes its shape from the floor tiles to the human form.

We have also a videoconferencing type sequence, where a person is sitting in front of a camera in a computer room, talking, and occasionally showing a few objects to the camera. This sequence is 3–minutes long.

Finally, we have a sequence of commercials. This sequence is about 50 seconds long, and contains 3 different advertisements. The first one contains panning, and fading of one scene to the next, the second one is an animated commercial, with very fast movement, and the third one is a mixture of animation and real-life images, again with very fast movement.

In this report, we describe our results for one minute of the sequences, and we note that the results are very similar for any minute of a given sequence. We present most of our results using H.261 encoded sequences. Many of the results are similar for MPEG and motion-JPEG encoded sequences; after showing all the results for H.261, we describe the differences caused by using MPEG and motion-JPEG encoding standards. For all three encoding schemes, we use encoders/decoders developed by the Portable Video Research Group (PVRG) at Stanford University [20]. We encode the sequences at SIF resolution (352x240), 30 frames per second.

6 Constant Bit Rate Video Encoding

In Figure 2, we show the block diagram of a station which encodes video according to CBR and transmits the encoded video stream over a network. As shown in the figure, to generate a constant bit rate stream, a *hypothetical rate control buffer* of size B bits is assumed to exist at the output of the encoder, which is drained at the target rate V bits/s. In order to ensure that the rate control buffer does not underflow, stuffing bits are inserted if the buffer would otherwise be empty. Likewise, in order to ensure that the buffer does not overflow, whenever the buffer cannot accommodate a newly generated macroblock, the macroblock is dropped. In that case, in order to maintain the continuity of the video syntax, a small code is inserted which instructs the decoder to display the macroblock located at the same position in the previous frame.

In order to reduce the likelihood of such underflows and overflows, the buffer occupancy level $b(k)$ (at the time when the bits corresponding to macroblock k are placed in the buffer) is used to adjust the quantizer scale $q(k+1)$ for macroblock $k+1$. The feedback function $q = f(b)$ is a linear function of the buffer occupancy (within the allowed limits for q, i.e.,

from 1 to 31), and its slope is inversely proportional to the buffer size B_{max}; i.e.,

$$q(k+1) = \begin{cases} \left\lceil \frac{q_{max}}{\alpha} \frac{b(k)}{B_{max}} \right\rceil & \text{if } b(k) < \alpha B_{max} \\ q_{max} & \text{otherwise} \end{cases}$$

where α is a constant which is recommended to be equal to 0.4 [24, 25], and and q_{max} is the maximum value allowed for the quantizer scale[2]. The relationship between $q(k+1)$ and $b(k)$ is also illustrated in Figure 3. The buffer occupancy $b(k)$ can be expressed as $b(k) = \sum_{i=1}^{k}(m_i - VT_i)$ where m_i is the number of bits for macroblock i, and T_i is the time elapsed between the encoding of the $(i-1)$'st and i'th macroblocks. (If we assume that the macroblocks are generated at regular intervals, then $T_i{=}\tau$ for all i.) Note that if we denote by $D_r(k)$ the delay experienced in the rate control buffer by macroblock k from the time the bits corresponding to the macroblock enter the buffer until the time they leave the buffer, then $D_r(k) = b(k)/V$.

To summarize, in CBR encoder control scheme, the hypothetical rate control buffer absorbs the short term variations in the bit rate, while the longer term behavior of the encoder is governed by the feedback control mechanism such that the average bit rate remains equal to V.

In Section 6.1, we characterize the CBR video quality for various video contents, and show how the quality depends on V and B. Then in Section 6.2, we consider the transmission of a CBR video stream over a circuit-switched network, and examine the resulting delay. In Section 6.3, we consider the statistical multiplexing of CBR streams over a circuit-switched network. We first characterize the fluctuations in the CBR traffic, and show that such fluctuations are of short-term, which indicates that such multiplexing is likely to be beneficial even for a small number of streams that can be multiplexed. We then show by some examples that this is indeed the case. In Sections 6.1 to 6.3, we focus on H.261 encoded video sequences; in Section 6.4, we show the differences resulting from using MPEG-1 and motion-JPEG standards.

[2]Note that in practice, the quantizer scale is not updated for every macroblock. Typically, in H.261, it is updated every 11 macroblocks, and in MPEG-1, every 22 macroblocks. For motion-JPEG, since the syntax does not allow for varying the quantizer scale within a frame, q is updated on a frame by frame basis.

6.1 Quality Characterization for Constant Bit Rate Video Encoding

Clearly, in CBR, as the scene complexity is increased, the quantizer scale used also increases so as to achieve the target bit rate V; as a result of the increase in the quantizer scale, the quality decreases. Some scenes may be so complex that even at the maximum allowed quantizer scale, the bit rate produced may exceed V. In such cases, the rate control buffer acts as a cushion to hold the excess bits produced. If the buffer is not large enough to accommodate all the excess bits, some macroblocks get dropped, causing a large amount of quality degradation. Thus, for a given V, if a particular choice of B results in buffer overflows, increasing B would improve the quality.

Now consider the case where B is large enough that there are no buffer overflows. In this case, the average data rate produced by the encoder must be equal to the target bit rate V, regardless of the buffer size. Therefore, the average quantizer scale for a given scene is fairly independent of the buffer size chosen. However, the magnitude of fluctuations in the quantizer scale become smaller as the buffer size is increased. Thus, for a small value of B, even if there are no buffer overflows, due to the large fluctuations in q, the quality may be degraded; as B is increased, the quality would improve. However, it would eventually reach a plateau at the region where B is large enough that the quantizer scale does not fluctuate too much for a given scene content.

Therefore, a limit is reached in the quality improvement when V is fixed and B is increased indefinitely. If the quality is desired to be increased beyond that point, then V must be increased.

In the following, we illustrate the effect of B, V, and video content on the quality for all five video sequences, for V values of 384 kb/s and 1536 kb/s, and B values of 19.2 kbits, 384 kbits, and 1920 kbits. For all possible combinations of the content, V, and B, we show b, q, $\hat{s}$, and Signal-to-Noise Ratio (SNR) versus time in Figures 4 through 33. We start with the Star Trek sequence as an representative example. We first show the examine the effect of B for a given V, namely, V=384 kb/s. We then consider the case of V=1536 kb/s, and examine the differences. Then we repeat the same progression for the other video sequences.

Now consider Figure 4, which is for the Star Trek sequence encoded using V=384 kb/s,

and B=19.2 kbits. The scene changes are shown by vertical dotted lines in the figures. The changes in b and q from one scene to another are quite apparent, and in some scenes the buffer is nearly empty, while in others it is full. Those scenes which cause the large buffer occupancy levels contain a lot of irregular camera movement (i.e., shaking of the camera). Therefore, the contents of a given frame are less likely to be correlated with those of the previous frame, and hence such scenes require more bits to be encoded for a given q. The $\hat{s}$ values vary quite significantly in time, at times becoming as low as 1. The points where the quality dips sharply correspond to those where the rate control buffer overflows, as expected. As for the SNR, while there is some correspondence between the SNR and $\hat{s}$ values, there is no one-to-one mapping; in particular, the SNR does not capture all the coarse degradations that occur due to buffer overflows, such as around frame number 800.

Now consider Figures 5 and 6, where B is increased to 384 and 1920 kbits, respectively. As B is increased, the buffer occupancy levels decrease with respect to B; for those particular B values chosen, there are no buffer overflows, and thus the quality degradations are not as large as for B=19.2 kbits. It is interesting to note that between B=384 kb/s and B=1920 kb/s, the quality does not change significantly; in fact, the minimum level of quality even slightly decreases as B is increased. The reason is that it takes longer to empty a larger buffer; therefore the quality is reduced for those scenes which come just after the scenes that cause the buffer to become full (e.g., the scene around frame 1500).

In Figures 7 to 9, we again show the same four types of plots, this time for V=1536 kb/s; the figures are again for B values of 19.2, 384, and 1920 kbits, respectively. For B=19.2 kbits, it is clearly seen that the q values fluctuate between the minimum and maximum points (i.e., 1 and 31). We have observed that in general, such large-magnitude fluctuations occur when B/V is less than about 40–50 ms; the reason is that at such small values of B the slope of the feedback function becomes too steep, causing large deviations in q for even a small change in the buffer occupancy level. As a result of such large fluctuations,$\hat{s}$ sometimes gets low despite the high bit rate used.

For B=384 kbits and B=1920 kbits, where the quantizer scale does not fluctuate too much, $b(k)$ and $q(k)$ are much smaller compared to V=384 kb/s. Furthermore, for V=1536 kb/s, the variations in q are much less compared to V=384 kb/s for the same buffer size. This is because for a given q and a given scene, a particular amount of increase in q causes a greater decrease in the data rate for a smaller q. (This trend is examined

further in the next section.) Therefore, for V=1536 kb/s smaller variations in q suffice to maintain the target bit rate. As a result of the smaller q values used, for V=1536 kb/s, the quality is quite good when B is large enough that the feedback control system is stable.

In Figures 10 through 33, we similarly show $b(k)$ vs. time, $q(k)$ vs. time, $\hat{s}$ vs. time, and SNR versus time for the Videoconferencing, Terminator 2, Raiders, and Commercials sequences. A general trend in the figures is that for a given B, the buffer occupancy for V=1536 kb/s is smaller than that for V=384 kb/s (except for B=19.2 kbits, in which case for all the sequences q values fluctuate across the entire range); indeed, the buffer in many cases gets full for V=384 kb/s, but it always remains within 25% of B for V=1536 kb/s. As a result, the quality for V=1536 kb/s is generally above 4.0.

As for the specifics for each sequence, first consider Videoconferencing. Recall that in that sequence, there are no scene changes, and the content does not vary significantly over time. Therefore, as expected, for that sequence the variations in $b(k)$ and $q(k)$ are smaller (except for V=1536 kb/s, B=19.2 kbits, which shows a large degree of variation for the same reason as for the Star Trek sequence.) This implies that the $\hat{s}$ and SNR values for the Videoconferencing sequence should be fairly uniform. This is indeed confirmed in the Figures 10 through 15.

For the Terminator 2 sequence, the $b(k)$ vs. time and $q(k)$ vs. time curves are quite similar to those for the Videoconferencing sequence. This is because the Terminator 2 sequence also does not contain a large degree of motion. However, particularly for V=384 kb/s, the $\hat{s}$ values are somewhat smaller in Terminator 2 compared to Videoconferencing. Likewise, for both V=384 kb/s and V=1536 kb/s, the SNR values for Terminator 2 are smaller than those for Videoconferencing by up to 5 dB.

In the Raiders sequence, we have one scene between frames 700 and 900 which is more complex than the other scenes; this scene causes buffer overflows for V=384 kb/s, and for B=19.2 kbits and 384 kbits. This leads to significant quality degradation for that scene. However, for V=1536 kb/s, the encoding of the same scene does not cause any increase in the buffer occupancy; hence, the quality remains at a good level. It is also interesting to note that here too, the SNR does not properly capture the large quality degradation due to the buffer overflows.

For V=384 kb/s, the Commercials sequence can be encoded at reasonable buffer levels up to frame number 800, and hence the quality remains fairly good during that period.

That part of the sequence corresponds to a car advertisement where any motion that exists is in the form of slow panning, which can be handled easily by motion compensation. Around frame 800, another advertisement begins, which consists mainly of cartoon images that move very quickly. This causes the buffer to fill-up, leading to large levels of quality degradation. For V=1536 kb/s the buffer level also gets somewhat larger for that portion, but not as significantly as for V=384 kb/s. As for SNR, it again does not capture well the degradations that occur during buffer overflows.

Now we examine more closely the effect of B on quality. Recall that for a given video content and a given V, as B is increased, the quality is expected to increase until the point where there is no buffer overflow; after that point, the quality would reach a plateau (and it may even decrease if B is made very large). In Figure 34 we illustrate this by plotting the maximum, average, and minimum values of $\hat{s}$ (computed over time) as a function of B for the Star Trek sequence; parts (a), (b), and (c) of the figure are for V=384, 512, and 1024 kb/s, respectively. The plateau effect is observed for all three values of V, with the difference that for a larger V, the achievable quality is better.

As for the effect of V on quality, in Figure 35, we plot the maximum, average, and minimum values of $\hat{s}$ as a function of V for the Star Trek sequence, for B=$\{$76.8,384$\}$ kbits. It can be seen that the quality reaches a plateau as V is increased as well. The plateau is reached around 600–700 kb/s for the minimum quality, around 500 kb/s for the average quality, and for less than 300 kb/s for the maximum quality. The values of the plateau are greater for B=384 kbits compared to B=76.8 kbits. This also confirms that one cannot choose an arbitrarily small buffer size and still maintain a given target quality by increasing V. For values of B greater than 384 kbits, we have observed that the $\hat{s}$ versus V curves look nearly identical, since the $\hat{s}$ statistics do not change when B is increased beyond 384 kbits for a given V as described above.

In Figure 36, we show the minimum value of $\hat{s}$ over time (denoted by $\hat{s}_{min}$) as a function of V for all five sequences, for B=384 kbits. Especially for values of V smaller than 1400–1500 kb/s, for a given V, the Commercials sequence has a much lower $\hat{s}_{min}$ compared to the other four sequences. For low values of V, such as 400–500 kb/s, there is a significant difference among the other four sequences as well. In particular, the Videoconferencing sequence is encoded with a very good quality value across the entire range of V values considered (384 kb/s to 1.5 Mb/s), while the other sequences have low values of $\hat{s}_{min}$ when

V is less than about 600 kb/s.

Now consider the effect of B and V taken together. In Figure 37, we show the equal $\hat{s}_{min}$ contours in the B–V space for various values of $\hat{s}_{min}$. The curve indicates that there is a trade-off between B and V to achieve a given quality objective. When B is chosen smaller, V must be chosen accordingly larger, and thus, more network bandwidth would be consumed by the video stream. When B is larger, V can be smaller, but then the delay for transmission over a circuit of bandwidth V would increase, as shown in the next subsection. What is most important is that the contours exhibit a very sharp knee behavior. Therefore, a good operating point is at the corner of the contour, where for both V and B near-minimum values are achieved.

Now let us examine the other video contents from the same point of view. In Figure 38, we plot the $\hat{s}_{min} = 4.2$ contours in the B–V space for all five video sequences under consideration. It is clear that the values of B and V which can achieve a quality of 4.2 units depend significantly on the video content. As expected, the Videoconferencing sequence attains the quality objective with the smallest B and V choices; conversely, the Commercials sequence can attain the quality objective only for very large values of B and V. Considering also that a video conferencing application may not require as good a quality as a motion picture with commercial advertisements, the differences become even more significant. The sharp-knee behavior is observed in the videoconferencing and Terminator-2 sequences as well, but not in Raiders and Commercials.

6.2 Characterization of Network Delay for Transmission over a Circuit

d In this section, we consider transmitting a CBR encoded video sequence with a given (B,V) pair over a circuit-switched network of bandwidth C, and examine the resulting delays. Note that C may be equal to or greater than V. Recall that we denote by $D(C,k)$ the delay for a macroblock k from the time the encoder places the bits corresponding to macroblock k into the encoder output buffer, until the time all the bits corresponding to macroblock k are delivered to the circuit. The end-to-end delay is then equal to $D(C,k)$, plus the propagation delay over the network, plus the delays due to encoding and decoding. All the delays other than $D(C,k)$ can be considered constant, and hence, it is this delay

that we focus on here.

A. CBR Video Transmission over a Circuit of Bandwidth Equal to V

First, consider that the circuit has a bandwidth C equal to V. In this case, $D(V,k) = D_r(k) = b(k)/V$; hence, $D(V,k)$ is directly proportional to the rate control buffer occupancy. Therefore, as B is increased, $D(V,k)$ also increases. To further quantify this increase, in Figure 39, we show the maximum, average, and minimum values of the delay experienced in the rate buffer as a function of B for V=$\{$384,1536$\}$ kb/s for the Star Trek sequence. As the figure indicates, the average buffer occupancy increases linearly with B. This is because for a given scene, the value of q which produces the target bit rate is independent of the value of B (ignoring the transients which occur when the content changes); for a given q, the buffer occupancy, and therefore the delay experienced in the buffer, is directly proportional to B. On the other hand, the maximum buffer occupancy exhibits a piecewise linear dependence on B; for V=384 kb/s, it first increases with a larger slope until about B=200 kbits, and for larger values of B it continues increasing, but with a smaller slope. This is because for values of B up to 200 kbits, the rate buffer sometimes gets full and overflows. For B >200 kbits, the buffer is large enough not to ever overflow. Similar trends have also been observed for the other video sequences.

In Figure 40, we show the maximum, average, and minimum values of $D_r(k)$ as a function of V, again for Star Trek, and for B=384 kbits. The results indicate that the delay experienced in the rate control buffer decreases as V increases. This is because for a given video content, smaller values of q are used in CBR encoding in order to match a greater target bit rate V. Thus, the buffer occupancy level remains relatively small.

Now consider the effect of B and V together on the delay. In Figure 41, we show the equal $\max_k\{D(V,k)\}$ contours in the B–V space. In order to achieve the given delay objective, one must operate at or below the curve corresponding that delay constraint. These curves indicate that as V is increased, a greater value of B can be used to achieve the same maximum delay objective. Clearly, increasing both B and V also results in an improved quality. This suggests the existence of a minimum V (and a corresponding B) for which given delay and quality objectives can be achieved. To illustrate that this is indeed the case, in Figure 41, we also show again the equal $\hat{s}_{min}$ contours for various values of $\hat{s}_{min}$. It is clear that given certain minimum quality and maximum delay objectives, there

is an *optimum* (B,V) pair that would achieve those performance objectives while producing the fewest number of bits; this (B,V) pair is at the intersection of the quality and delay contours. For example, if a minimum quality level of 4.2 and a maximum delay of 50 ms is to be achieved at all times, the optimum choice of the (B,V) pair is (300 kbits,1050 kbits/s).

In Figure 42, we show the $\max_k\{D(V,k)\} = 100$ ms contours in the B–V space for the five video sequences. Here too, there is a significant difference among the sequences: for the videoconferencing sequence B can be relatively large while $\max_k\{D(V,k)\}$ remains less than 100 ms; on the other extreme, for the Commercials sequence, B must be chosen small to keep $\max_k\{D(V,k)\}$ less than 100 ms. This indicates that for the values of V considered, the videoconferencing sequence is encoded using a relatively small value of q (thus keeping the rate control buffer mostly empty), while the Commercials sequence requires using larger q values, which implies that the rate control buffer occupancy tends to be greater for that sequence.

Figures 38 and 42, taken together, imply that the pair (B,V) which satisfies a given performance objective $(D_{max},\hat{s}_{min})$ is very much content-dependent. For example, to meet the performance objective of (D_{max}=100 ms,$\hat{s}_{min}$=4.2), the Videoconferencing sequence can be encoded using ($B \approx$150 kbits,$V \approx$370 kb/s), while the Commercials sequence requires ($B \approx$300 kbits,$V \approx$2500 kb/s).

B. CBR Video Transmission over a Circuit of Bandwidth Greater than V

We now consider the case where $C > V$. In Figure 43, we show $D(C,k)$ versus time for the Star Trek sequence, V=384 kb/s, B=384 kbits, and for C=384 kb/s and C=512 kb/s. It is clear that increasing C to 512 kb/s significantly reduces the delay.

In Figure 44, we show $\max_k\{D(C,k)\}$ versus C for V=384 kb/s, and B=$\{$38.4,153.6,384$\}$ kb/s. The figure indicates that the maximum delay decreases first very rapidly as C is increased, and then it keeps decreasing, but at a slower rate. In all cases, when C is equal to about 1.3–1.5 times V, the delay becomes less than about 100 ms, even when a very large buffer is used. Similar results also apply to the other sequences. Therefore, given a certain transmission capacity C, and certain quality and delay objectives $\hat{s}_{min}$ and D_{max}, one can use a value of V smaller than C to meet those objectives. This is especially useful when the maximum delay objective is very small, such as 20–30 ms. For such small delay values, if one chooses $V = C$, then one must choose B=$V \times D_{max}$, since the rate control

buffer's occupancy level fluctuates largely, many times reaching the full level; this may lead to a low quality. If instead one chooses an appropriate V which is smaller than C, and choose a relatively large B value, one can meet both the quality and delay objectives. As an example, consider that C=1024 kb/s, D_{max}=30 ms, and $\hat{s}_{min}$=4.2. We have determined that for the Star Trek sequence, it is not possible to set V=1024 kb/s and satisfy these delay and quality objectives at the same time for any value of B. On the other hand, according to Figure 37, for V=768 kb/s and B=384 kbits, the quality objective would be met. Thus, consider encoding the sequence at that (B,V) pair, and transmitting it over the circuit of C=1024 kb/s. In Figure 45, we show $D(1024\ kb/s, k)$ versus time for that case. Indeed, it is clear that the maximum delay is about 25 ms, and therefore both the delay and quality objectives are met.

6.3 Traffic Characteristics for CBR Video

Given that multiple CBR streams may be statistically multiplexed over a network in order to reduce the end-to-end delay, it is of interest to examine the traffic characteristics for CBR video as well. In Figure 46, we show the frame sizes as a function of time for the Star Trek sequence, V={384,1536} kb/s, B={19.2,384,1920} kbits. It is clear that the fluctuations in the frame sizes increase as B is increased. However, those fluctuations do not appear to last long, which is as expected given that the CBR feedback control does not allow the produced bit rate to deviate from V for long periods of time. In the figures, some single frames appear quite larger than the others; those frames correspond to the scene changes. For a given B, the magnitude of the spikes corresponding to the scene changes are about the same between V=384 kb/s and V=1536 kb/s.

In Figure 47, we show the frame size histograms for the same cases. The standard deviation of the frame sizes is also shown in the figure for each case. It is interesting to note that for a given V, the histograms are always concentrated within the same region regardless of B. However, as B increases, the tail of the distribution also increases, thus increasing the standard deviation[3].

In Figure 48, the frame size autocorrelation function is shown for the same cases. This

[3]One exception to this is V=1536 kb/s, in which case the standard deviation is greater for B=19.2 kbits as compared to B=384 kbits. This is because of the instability in the feedback function for B=19.2, causing relatively large fluctuations in the frame sizes.

function indicates how long the deviations from the average value persist. As expected from the CBR feedback function, the autocorrelation function becomes small very rapidly; thus, the deviations from the average do not last very long.

In Figures 49 to 60, we show the frame size versus time, frame size histogram, and frame size autocorrelation function for the other four sequences, and for the same (B, V) pairs as above. In all cases, the trends are very similar.

In Table 1, we show the maximum, minimum, and standard deviation of bits per frame for the five sequences, V=1536 kb/s, and B=384 kbits. (Note that the average number of bits per frame for V=1536 kb/s is 51200.) The Commercials sequence exhibits the largest maximum and standard deviation of bits per frame, while the Videoconferencing sequence exhibits the smallest ones. In particular, the maximum number of bits per frame is only about 20% greater than the average for the Videoconferencing sequence; this is because there is no scene change in that sequence. In all cases, the standard deviation is within 4% to 10% of the average, indicating that the variations are not very large.

These results indicate that while there are some differences from one sequence to another, and from one B value to another, a common trait is that the variability in the CBR sequences is short-term, and of relatively small magnitude compared to the average. This suggests that it is possible to achieve significant reductions in the end-to-end delay by statistically multiplexing CBR sources, even on a network with a relatively low bandwidth. While we do not address here the full treatment of the CBR video performance over networks, we consider a simple example where N_v CBR streams are multiplexed over a circuit of bandwidth W as shown in Figure 61. The multiplexer buffer size is considered to be unlimited. The bits corresponding to each macroblock are sent to the multiplexer buffer instantaneously, as soon as they are generated. We ignore any overhead due to framing. We denote the delay incurred in the multiplexer buffer by macroblock k as $D(W, k)$. We have simulated this scenario, driving the simulator by the video traces obtained here. We have considered that each video source uses the same video sequence; the sequences are treated as circular lists of macroblocks, and each source starts transmitting at a random point in the list in order to reduce the correlations between the sources. We have performed a large number of simulation runs for any given scenario, each time choosing different random starting points, and recorded in each simulation run the maximum delay incurred by any source in the multiplexer buffer (denoted by $\max_k\{D(W, k)\}$).

As examples, consider the Videoconferencing and Commercials sequences. We have encoded them such that they have a reasonable quality at the source (at least 4.0 at all times). For Videoconferencing, this is accomplished with V=384 kb/s, B=38.4 kbits, and for Commercials, V=1536 kb/s, and B=768 kbits. In both cases, we have varied N_v from 1 to 16, and chosen W to be equal to 1.01, 1.04, 1.10, and 1.20 times VN_v.

In Figures 62 and 63, we show the histogram of $\max_k\{D(W,k)\}$ for the Videoconferencing and the Commercials sequences, respectively. It is clear that as N_v is increased, the maximum delay experienced in the multiplexer buffer decreases significantly, even when $W = 1.01VN_v$. For N_v=16, and for $W = 1.20VN_v$, the maximum delay becomes less than 10 ms for Videoconferencing, and less than 20 ms for the Commercials. Therefore, even when there are a moderate number of CBR streams to be transmitted, significant reduction in the delay is possible by means of statistical multiplexing, as opposed to partitioning the network bandwidth for each CBR stream.

6.4 Results for MPEG-1 and Motion-JPEG Compression Schemes

A. MPEG-1

In Figure 64, we show the number of bits per frame as a function of time for the Star Trek sequence, using MPEG-1 with the GOP Structure 1, for V=384 kb/s and B=384 kbits. As expected, there are a lot more variations in this figure compared to its H.261 counterpart (Figure 46(b)) as a result of the differences among the I, P, and B frames. In Figure 65, we show $b(k)$ versus time for the same sequence. It is interesting to note that the variations of the buffer occupancy over time look quite similar between MPEG-1 and the corresponding H.261 sequence (Figure 5(a)) despite the greater frame size variations in MPEG; however, the maximum buffer occupancy in MPEG-1 is slightly larger.

In Figure 66 we show the maximum, average, and minimum $\hat{s}$ as a function of B for the Star Trek sequence, CBR encoded using the GOP Structure 1; parts (a) and (b) of the figure are for V=384 kb/s and V=1024 kb/s, respectively. The figure indicates that for a given V, the MPEG encoded sequence reaches its plateau value at a smaller B compared to the H.261 encoded sequence (Figure 34). The minimum quality attained at the plateau is about the same for MPEG and H.261.

As far as the GOP Structure 2 is concerned, we have observed that the buffer occupancy

characteristics for the MPEG encoded sequences with GS2 are very similar to those for H.261. As a result, the delay and $\hat{s}$ statistics are also similar between the MPEG GS2 encoded sequences and H.261 encoded sequences.

As for the effect of content and the optimum value of the (B,V) pair to achieve given performance objectives, the results for MPEG-1 are very similar to those for H.261.

B. Motion-JPEG

Recall that in Motion-JPEG, the quantizer scale can only be updated on a frame-by-frame basis. Thus, the sampling rate for the feedback function is much lower compared to the H.261 and MPEG. As a result, especially for small-to-medium buffer sizes (i.e., 150–500 kbits), the feedback control mechanism exhibits an oscillatory behavior. First, an entire frame is encoded using a very small q, which fills-up the buffer; then, a number of frames are encoded using a large value of q until the buffer is emptied; then another frame is encoded using a very small q, and so on. As an example, in Figure 67, we show q versus time for the Star Trek sequence, encoded using Motion-JPEG, V=1536 kb/s, and B=153.6 kbits. Clearly, the feedback control mechanism is unstable, and the oscillations cover the full range of q. In Figure 68 we show the corresponding number of bits per frame. Following the q values, the frame sizes fluctuate from about 10 kbits to about 500 kbits, a factor of 50. As a result of this instability, in practice to generate constant-bit-rate Motion-JPEG video, iterative procedures are used, for example, by performing a search for the quantizer scale until the right value that gives the target bit rate is found. However, since we are interested in real-time encoding, such approaches are considered out of the scope of this report.

Another way of removing the instability is to use a large buffer size. In Figure 69, we show q as a function of time for V=1536 kb/s, and B=1536 kbits. In this case, the oscillations have been mostly eliminated. In Figure 70, we show the corresponding number of bits per frame. Although there are occasional peaks which correspond to the times when q gets very small, generally the number of bits per frame curve is stable.

In Figure 71, we show the minimum, average, and maximum $\hat{s}$ versus V for the Star Trek sequence, Motion-JPEG, CBR, B=1536 kbits. It is interesting to note that the minimum quality first increases as V is increased, and then somewhat decreases. This is because for a given buffer size, the feedback mechanism tends to be less stable as V is increased,

resulting in larger fluctuations in q.

7 Open-Loop Variable Bit Rate Video Encoding

In Figure 72, we depict the block diagram of the sending station for Open-Loop VBR. As shown in the figure, the quantizer scale q is simply kept at a constant value q_0 at all times. Thus, the feedback loop is "open," and hence the name Open-Loop VBR (OL-VBR). When OL-VBR encoded video is to be sent over a network, the main issue is to select q_0 appropriately. Clearly, for a given video sequence, greater values of q_0 would result in fewer bits to be produced, but also cause greater quality impairment. Therefore, the choice of q_0 represents a trade-off between quality and delay. However, for any given q_0, both the quality and rate of the encoded video vary according to the content. In Sections 7.1 and 7.2, we demonstrate the effect of the video content on quality and traffic, respectively, focusing on the H.261-encoded video sequences. We show that for small values of q_0, the quality is generally good, but the amount of traffic produced is very large, and it is highly dependent on the content. As q_0 is increased, the quality decreases, and becomes more dependent on the video content; the traffic also decreases, but it still remains content dependent. Thus, it is difficult to specify a clear methodology for selecting q_0. In Section 7.3 we consider the MPEG and motion-JPEG standards, and discuss the similarities and differences observed in the quality and traffic characteristics when those standards are used as opposed to H.261.

7.1 Quality Characterization for Open-Loop VBR

Here we examine the quality characteristics, starting with the Star Trek sequence as a representative example. In Figure 73, we show $\hat{s}$ (again measured in one second intervals) as a function of time for the Star Trek sequence, and for various values of q_0. For clarity, we show the figure in two parts; part (a) is for q_0=$\{4,16\}$, and part (b) is for q_0=$\{8,31\}$. It is clearly seen that as q_0 is increased, not only the overall quality decreases, but also the variations in quality become more accentuated.

In Figure 74, we show the average, maximum, and minimum values of $\hat{s}$ as a function of q_0 for the Star Trek sequence. As the figure indicates, all three quantities decrease nearly linearly with q_0. What is important to note is that the minimum quality decreases very rapidly as q_0 increases, and it gets to "annoying" levels for q_0 greater than about 10–12.

Now consider the other four sequences as well. In Figures 75 to 78, we plot $\hat{s}$ versus time for those sequences. For all five sequences, $\hat{s}$ is very good for q_0=4 (i.e., it is always above 4.5). As q_0 is increased, $\hat{s}$ decreases, and in all the sequences variations within the sequence get accentuated. The smallest variations within the sequence occur for Videoconferencing, which is expected given the relatively uniform content of that sequence.

In Figure 79, we show the minimum value of $\hat{s}$ over time (denoted as $\hat{s}_{min}$) as a function of q_0 for all five sequences. As the figure indicates, while the $\hat{s}_{min}$ decreases nearly linearly for all the sequences, the rate of decrease is highly dependent on the sequence. Consequently, the maximum value of q_0 for which a given quality objective is attained at all times is dependent on the content quite significantly. For example, to achieve a quality of at least 4.0 at all times, the Star Trek sequence must be encoded at q_0=8 or less, while the Videoconferencing sequence can be encoded up to q_0=22.

We have established in the previous section that SNR does not provide a good match to $\hat{s}$(and hence the perceived video quality). In order to further demonstrate that, in Figure 80 we plot SNR versus time for the Commercials sequence, for q_0 values of 4,8,16, and 31. Clearly, there is a discrepancy between the SNR values and the $\hat{s}$ values shown for the same sequence in Figure 78. In particular, the region between frames 850 and 1050 has very good SNR values, without a correspondingly good $\hat{s}$. In that region, some text is displayed on the screen, and a large portion of the screen contains a flat background. The macroblocks that correspond to such a flat background have only a DC component, which is quantized at a small quantizer step size (independently of q_0); thus, the noise introduced by quantization is very small in the background regions. Since a large portion of the display consists of the background, this makes the SNR to be high. However, for large values of q_0, the text characters get distorted quite significantly, and therefore, the quality in these regions should not be judged as good. The ITS measure captures the degradations in that region more accurately.

7.2 Traffic Characterization for OL-VBR

Let us again start by examining the Star Trek sequence as a representative example. In Figure 81, we show the number of bits per frame as a function of time for that sequence, for q_0=8. The dotted vertical lines in the figure indicate scene cuts. Note that when there is a scene cut, the first frame in the new scene is likely to be uncorrelated with the

preceding frame. Therefore, the first frame in a scene is encoded using a larger number of bits compared to the other frames in the same scene. Thus, the spikes that we see in the figure correspond to scene cuts. It is interesting to note that the upper end of the number of bits per frame (and therefore the data rate) varies significantly from one scene to another, while the lower end remains roughly the same for all scenes, around 10–15 kbits. Thus, the average data rate also varies significantly from frame to frame.

The reason for having many frames in the 10–15 kbits range regardless of the particular scene being encoded is as follows. The Star Trek sequence is converted from 24 frames per second film to 30 frames per second NTSC video using the so-called 3:2 pulldown scheme [26], and then to 30 frames per second SIF format by sampling the odd fields. The net effect of this conversion is the repetition of one frame every four frames. Since a repeated frame contains the same image as its preceding frame (except for any noise which may have been introduced during the analog-to-digital conversion), it is encoded using a relatively small number of bits. The same effect is observed in Raiders and Terminator 2 sequences as well, since they are also converted from film. (This effect is not as pronounced for the CBR sequences, since the "repeated" frames would be encoded using different values of q from each other, depending on the rate control buffer level.) In the remainder of this paper, when we plot the number of bits per frame as a function of time, we do not show these repeated frames in order to make it easier to visualize the variations in the traffic. (However, when computing traffic statistics, we take into account every frame.) Now consider Figure 82, which is the counterpart of Figure 81 without the repeated frames being shown. Indeed, here the variations in the frame sizes from scene to scene are more apparent.

In Figure 83, we show the frame size histogram for various values of q_0 for the Star Trek sequence; part (a) of the figure is for q_0=$\{1,4,8\}$, and part (b) of the figure is for q_0=$\{8,16,22,31\}$. Note that for q_0=1, the frame size histogram is fairly symmetrical; as q_0 gets larger, the histogram gets more skewed with a longer tail relative to the mean. It is also interesting to note that when q_0 is small, the frame size histogram is very dependent on q_0, while for larger q_0 values (i.e., between q_0=16 and q_0=31), the histograms change very little. The reason is as follows. In DCT compression, most of the reduction in data rate comes from quantizing many DCT coefficients to zero, and then run-length encoding the DCT coefficients. Around q_0=16, most of the DCT coefficients that are close to zero have already been quantized to zero, and increasing the quantizer scale further does not

result in many more coefficients to be quantized to zero.

To characterize further how the traffic depends on q_0, in Figure 84, we show the average, maximum, minimum, and standard deviation of frame size as a function of q_0 for Star Trek. As the figure indicates, for small values of q_0, all the frame size statistics decrease sharply as q_0 is increased; as q_0 gets larger, the rate of decrease in frame size statistics gets smaller. For example, between q_0=1 and q_0=16, the average number of bits per frame differs by a factor of 19, and the maximum number of bits per frame differs by a factor of 9. By contrast, between q_0=16 and q_0=31, the average number of bits per frame differs by a factor of 1.25, and the maximum number of bits per frame differs by a factor of 1.4.

Now consider the frame size statistics for the other video sequences. In Figures 85 to 88, we show the number of bits per frame for the other four sequences, and for q_0=$\{$4,8,16,31$\}$. For all the sequences, changing the q_0 value mainly scales the number of bits per frame, without changing the relative frame sizes. For the Videoconferencing sequence, as expected, the variations in the frame sizes are less compared to the other sequences. In contrast, the Commercials sequence has larger variations compared to the other sequences; in particular, the second half of the sequence is encoded using significantly more bits for a given q_0 compared to the other sequences. To characterize the differences in frame size statistics further, in Figure 89 we show for the five video sequences the average and maximum number of bits per frame as a function of q_0. As the figure indicates, there is a significant difference between the Commercials sequence and all the others for all values of q_0. This is mainly because of the relatively large amount of motion and the sharp edges (which result in large amount of high frequency components in the spatial domain) for the second and third advertisements in the sequence. Furthermore, although not as significant, there is still some difference among the other four sequences, especially for q_0 values smaller than about 8. The differences in traffic characteristics among those four sequences can be seen more clearly in Figure 90, where we show the frame size histogram for the five sequences for q_0=8. Here too, the Commercials sequence is clearly distinguishable from the other four, as it has a much longer tail. Among the other four sequences, Star Trek and Raiders have a longer tail compared to the Terminator-2 and Videoconferencing sequences.

In Figure 91, we show the frame size autocorrelation for all five sequences, encoded at q_0=8[4]. One interesting observation is that the autocorrelation functions are very different

[4]For the sequences converted from film, we have replaced the size for a repeated frame with the average

from one sequence to another. The autocorrelation in the Commercials sequence persists for several hundreds of frames (i.e., tens of seconds). This is mainly because for this sequence, the successive scenes in the same advertisement have similar content, and this results in a similarity among the characteristics of hundreds of successive frames. The Raiders sequence also shows a strong autocorrelation, especially up to about 2–3 seconds of time lag (which is about the average scene length in this sequence). The autocorrelation function for the Videoconferencing sequence contains a periodic component with a period of 10 frames. In this sequence, the content does not vary too much; in particular, the background remains completely unchanged throughout the sequence. In this case, some of the variations from frame to frame are determined by the particular GOB that is being intracoded. Since the same GOB is intracoded every 10 frames, there is some correlation between frames that are separated from each other at multiples of 10. This effect is not seen in the other sequences because of the much greater variations in their content. As for other values of q_0, the autocorrelation functions are nearly identical to those given here; this is as expected since we have seen that changing the q_0 does not significantly change the relative sizes of the frames.

The above results indicate that both quality and traffic statistics vary with the video content for a given q_0. Now consider the quality and traffic taken together. In Figure 92, we show the average, maximum, and minimum values of $\hat{s}$ as a function of the average frame size (which is equal to the average bit rate divided by 30). As the average bit rate increases, the quality first sharply increases, then it reaches a plateau. To get a good quality at all times (i.e., a minimum of 4.2–4.3, and an average around 4.5), the average number of bits per frame needs to be around 30000–50000 (corresponding to the average bit rates in the range of 900-1500 kb/s). This corresponds to a q_0 range of 1–5. In Figure 93, we show the minimum quality as a function of the average frame size for all five sequences. For the average frame sizes smaller than about 60000 bits (corresponding to a data rate of 1.8 Mb/s), the quality varies significantly from one sequence to another for a given average frame size.

From the results given here, we can conclude that for a given q_0, both the resulting

of its preceding and succeeding frames' sizes, because otherwise a periodic autocorrelation component is introduced due to the correlation between the sizes of the repeated frames, which makes it more difficult to identify the long-term trends

data rate and quality depend on the video content; therefore, without apriori knowledge of the content (or if the content is highly variable in time), it is difficult to specify a clear methodology for selecting q_0. For small q_0 values (i.e., $q_0 \leq 5$), the quality is consistently very good, but the resulting traffic is highly variable, and has a large average. For larger q_0 values, the quality varies according to the content; the traffic also varies, although not as significantly as for the small q_0 values.

7.3 Results for Other Compression Schemes

A. MPEG-1

For MPEG-1, for both GS-1 and GS-2, the quality statistics are very similar to those obtained for H.261. As for the frame size statistics, in Figure 94, we show the number of bits per frame versus time for the Star Trek sequence as a representative example, encoded at q_0=8, using GS1. Comparing this figure with the corresponding figure for H.261 (i.e., Figure 82), we observe that in the MPEG encoded sequence, there is more of a short-term variation in the frame sizes due to the differently encoded frame types; in particular, the large spikes correspond to the I frames.

In Figure 95 we plot the frame size histogram for the same MPEG-encoded sequence, as well as the corresponding H.261-encoded sequence. The average number of bits per frame is very close between the two cases: 18.9 kbits in MPEG-1 as opposed to 20.9 kbits in H.261. For other q_0 values and other contents, similar results are observed. The slight difference in the average frame size is because of the B frames in MPEG, which are encoded more efficiently, as well as because of other improvements in the MPEG standard, such as the half-pixel accuracy in the motion estimation (as opposed to the one-pixel accuracy in H.261). In Figure 96, we show for the same sequence the frame size histograms individually for the I, P, and B frames. The figure indicates that on an average sense, the B frames are encoded using the least number of bits, then the P frames, and then the I frames.

As for the GS2, the quality statistics and the average and maximum values of the frame sizes are very similar to those for GS1.

B. Motion-JPEG

In Figure 100, we show $\hat{s}$ as a function of time for the Star Trek sequence, motion-JPEG encoded for q_0 values of 50,100,200, and 300 (roughly equivalent to q_0=8,16,32, and 48 for H.261 and MPEG-1). As the figure indicates, for q_0=50, the quality is nearly constant around 4.5; for q_0=300, the quality sometimes drops down to about 2.5. This behaviour is again similar to that observed in H.261 and MPEG. In Figure 101, we show the maximum, average, and minimum $\hat{s}$ versus q_0 for the Star Trek sequence. The figure indicates that these quality statistics decrease linearly with q_0 just as in H.261 and MPEG. In Figure 102, we show the maximum, average, and minimum $\hat{s}$ versus the average number of bits per frame. A comparison of this figure with Figure 92 indicates that for a given average frame size, the quality is lower for Motion-JPEG as compared to H.261 and MPEG, which is as expected given the less efficient compression in Motion-JPEG due to the lack of intercoding capability. For example, when the average frame size is equal to 50 kbits, the minimum quality attained by the Motion-JPEG sequence is about 4.2, whereas the minimum quality attained by the H.261 sequence is about 4.5.

In Figure 97, we show the frame size as a function of time for the Star Trek sequence, encoded using q_0=50. Since interframe coding is not employed in Motion-JPEG, the temporal complexity is irrelevant. Therefore, in the figure, there are no spikes corresponding to the scene cuts. Furthermore, some scenes with a relatively high spatial complexity, such as the 5th scene in the sequence, are encoded using a relatively large number of bits compared to the other scenes. In H.261 and MPEG, that particular scene does not require such a large number of bits relative to the other scenes; this is because the temporal complexity of the scene is relatively low, and interframe coding is able to reduce the number of bits significantly.

Comparing Figure 97 with Figure 94 reveals that the size of the I frames in the MPEG sequence and the corresponding frame sizes in the JPEG sequence are very similar. This is as expected since the I frames in MPEG-1 use the same default quantization matrix as in JPEG. This similarity is further confirmed by comparing the frame size histogram for the Motion-JPEG sequence (shown in Figure 98) with the I-frames histogram for the MPEG sequence (Figure 96).

In Figure 99, we show the maximum, average, and minimum frame sizes as a function of q_0 for the Star Trek sequence. For reasons similar to the H.261 and MPEG schemes,

here too for small q_0 values, the frame size decreases sharply as q_0 is increased; as q_0 gets larger, the rate of decrease becomes smaller.

It is also interesting to note that the relative traffic statistics for the five sequences are quite different from those for H.261 and MPEG, which is because in Motion-JPEG the temporal complexity of a sequence is not relevant. In Table 2 we show the average, standard deviation, maximum, and minimum frame sizes for the five sequences, Motion-JPEG encoded at q_0=50. Here too, the average frame size for the Commercials sequence is larger than the other four, but not by as large a margin as in H.261. Interestingly, in Motion-JPEG the second largest average frame size is attained by the Videoconferencing sequence, followed closely by the Terminator 2 and Raiders sequences, and the smallest average frame size is attained by the Star Trek sequence. Also, the peak-to-average frame size ratio for Motion-JPEG is around 1.2 to 2, which is not as large as that in H.261 and MPEG; the reason is that in Motion-JPEG all frames are intracoded, resulting in less dependency on the content.

8 Constant-Quality VBR Video Encoder Control Scheme

In Section 6, we have shown that for Constant Bit Rate (CBR) encoding, one can choose the data rate and the rate control buffer size appropriately to achieve a given quality objective; but this requires choosing these parameters large enough to accommodate the worst case, and therefore many scenes would be encoded using more bits than needed to achieve the given quality objective. Likewise, in Section 7, we have seen that with OL-VBR, for a small value of q_0, the quality achieved is quite good; however, the traffic produced is highly variable, and its average is quite high. The value of q_0 which would produce the fewest number of bits while meeting a given quality objective depends highly on the video content. Consequently, if a scheme is designed to maintain a desired quality objective at all times, such a scheme would produce fewer bits on average compared to CBR and OL-VBR schemes. In this section, we devise and characterize such a scheme. As in the previous sections, here too we first focus on the H.261 scheme. In Section 8.1, we describe the design of the Constant-Quality VBR (CQ-VBR) scheme. In Section 8.2 we characterize the quality for the CQ-VBR scheme, and show that it is indeed able to achieve a consistent level of quality. In Section 8.3, we examine the traffic resulting from using the CQ-VBR

scheme. We show that sometimes there are some short-term, but high-magnitude peaks in the produced traffic. In Section 8.4, we consider sending CQ-VBR streams over a circuit, and examine the resulting delays. In Section 8.5, we consider the MPEG-1 and motion-JPEG standards, and compare their results with those for H.261. In Section 8.6 we devise a modification to the CQ-VBR scheme where in addition to the quality, the peak rate of video is also controlled. This scheme, referred to as Joint Peak Rate and Quality Controlled VBR (JPQC-VBR), is particularly useful if the network cannot accommodate the large peaks produced by the CQ-VBR scheme while meeting the delay and quality requirements of the application. We show that with the JPQC-VBR scheme it is possible to reduce the peaks significantly without a severe degradation in quality. Finally, in Section 8.7, we compare the CQ-VBR scheme with the CBR and OL-VBR schemes from a traffic and quality point of view.

8.1 Design of the CQ-VBR Feedback Function

To encode video streams at a constant quality $\hat{s}_{target}$, we have devised a feedback control scheme, where we measure the quality $\hat{s}(k,w)$ at every sampling point k using the last w frames, and use the difference $(\hat{s}_{target}\text{-}\hat{s}(k,w))$ as feedback to adjust the quantizer scale q by means of an appropriate feedback function $q(k+1)=f(\hat{s}_{target}\text{-}\hat{s}(k,w))$. The block diagram of the encoder for this scheme is depicted in Figure 103. The design problem to be solved here is to choose the feedback function f and the quality estimation interval w appropriately so as to cause neither instability, nor too slow a response time. We have considered a feedback function of the PID (Proportional, Integral, Derivative) type, since this type of feedback function is known to be effective for a wide range of systems [27]. The PID feedback function is given by

$$\begin{aligned} q(k+1) &= K_p e(k) + K_p \frac{T}{T_I} \sum_{i=1}^{k} e(i) + q(0) + \\ & \qquad K_p \frac{T_D}{T}[e(k) - e(k-1)]. \end{aligned}$$

where $e(k) = \hat{s}(k,w) - \hat{s}_{target}$, and T is the sampling period of the system. Therefore, our design variables are the PID coefficients K_p, T_I, and T_D, the quality estimation interval w, and the sampling period T. Since smaller sampling periods result in better performance in digital control systems, we choose the sampling period to be as small as possible. Therefore,

for H.261, the sampling period we choose is equal to the frame interval.

In order to determine the PID coefficients, we have employed the "Ziegler-Nichols PID tuning using the stability limit" method [27]. The method works as follows (see [27] for details). The system is first controlled using only proportional control. The gain, K_p, is increased until continuous oscillations result, at which point the gain, K_u, and the oscillation period, P_u, are recorded. The PID gains are then determined as follows: $K_p = 0.6K_u$, $T_I = P_u/2$, and $T_D = P_u/8$. We have applied this procedure iteratively for various values of K_u, and for $\hat{s}_{target}$=$\{$3.5,4.0,4.5$\}$. In those experiments we have used 3 video sequences: Star Trek, Videoconferencing, and Commercials.

For H.261, we have determined that for all three sequences, $K_u \approx 20$,and $P_u \approx 4$, fairly independently of the particular sequence being used and the quality target chosen. As a representative example, in Figure 104, we plot the quantizer scale as a function of time for the frames 200 to 300, for proportional control at $\hat{s}_{target}$=4.0, w=3 frames, and K_p =$\{$15,18,20$\}$. It is clearly seen that for K_p=15 there are no sustained oscillations, while for K_p=20, there are such oscillations, particularly after frame number 260. For K_p=18, there are also some oscillations, but they are not as steady and periodic as for K_p=20. Therefore, we have chosen K_u=20. (We have also encoded the sequences by using the PID coefficients resulting from K_u=18; the results did not differ significantly from the case with K_u=20.)

As far as the quality estimation interval w is concerned, we have experimented with various values, and determined that w=3 frames gives the best results. Note that while we use only 3 frames to measure the quality for the feedback control purposes, when we present performance results of the CQ-VBR scheme, we still measure the quality at 1-second intervals (i.e., 30 frames) for the reasons explained in Section 4; namely, one second is small enough to capture variations according to the changes in the scene content, and large enough to correspond to the response time of the human visual system.

An important aspect of the CQ-VBR scheme is that it is possible to operate it in real-time, provided that $\hat{s}$ can be computed in real-time. Indeed, with some simplifications (e.g., replacing the Sobel filter with another filter that requires less computation), a software implementation which achieves real-time computation of $\hat{s}$ was created at the ITS using 80386-based personal computers [5]. (However, those simplifications result in some reduction in the accuracy of the measure; the exact degree of such reduction is not specified

by the authors.) Considering that a real-time software MPEG-1 encoder requires at least a 90 MHz Pentium processor, which is roughly an order-of-magnitude faster than a '386 machine, computing $\hat{s}$ appears to require a small fraction of the total processing power for doing real-time software encoding. Another example of the ratio of computing power between computing $\hat{s}$ and compressing a frame is that our implementation of $\hat{s}$ takes about 0.5 seconds per frame on a DECStation 5000/240; on the same platform encoding video sequences using H.261 and MPEG-1 take about 5 and 15 seconds per frame, respectively.

As far as hardware implementations which achieve real-time computation of $\hat{s}$ are concerned, we note that the most time-consuming operations are the computation of SI and TI, which involve Sobel filtering, pixel differencing of two successive frames, and computing standard deviation in the space domain. For all these operations, a frame can be divided into smaller regions and the operations can be performed concurrently in those regions. Therefore, the total number of operations required for computing $\hat{s}$ is considerably smaller than that required to encode a frame, and those operations can easily be parallelized; hence, it appears possible to provide a hardware implementation of $\hat{s}$ which operates in real-time.

8.2 Characterization of Quality for CQ-VBR

In this section, we begin with examining the videoconferencing sequence; the content of this sequence is fairly uniform, and thus we can expect quantizer scale not to vary too much in order to achieve a constant level of quality. In the following we show that this is indeed the case with the CQ-VBR scheme. Moreover, we determine how responsive the feedback function is, by choosing the initial value of q arbitrarily, and observing how quickly the appropriate range of q is reached.

In Figure 105, we show q as a function of time for videoconferencing, for $\hat{s}_{target}$ values of 4.0 and 4.5. As the figure indicates, the quantizer scale fluctuates around 25 for $\hat{s}_{target}$=4.0, and around 10 for $\hat{s}_{target}$=4.5. In order to illustrate how quickly the appropriate q range is reached, consider Figure 106, where the same plot is shown for the first 90 frames in the sequence. As seen in the figure, it takes only 10–15 frames to reach the appropriate q range. In this particular case, the initial q is chosen to be equal to 8; we have experimented with other values of initial q, and observed the same response time behavior in all cases.

In Figure 107, we show the quality as a function of time, again for videoconferencing, for $\hat{s}_{target}$=4.0 and 4.5. As the figure indicates, the quality level varies no more than $\pm$0.2

units around $\hat{s}_{target}$ at all times.

Of course, the more challenging case for CQ-VBR is when there are frequent scene changes, as it is the case with the other four video sequences under consideration. In particular, for the Star Trek sequence, the average scene lasts only about 2 seconds; thus the controller may have to readjust its parameters every 2–3 seconds. In Figures 108 to 115, we plot $\hat{s}$ versus time and q versus time for the Star Trek, Terminator 2, Raiders, and Commercials sequences. Also in Table 3, we give for all five sequences the average, standard deviation, minimum, and maximum values of $\hat{s}$ for $\hat{s}_{target}$=$\{4.0,4.5\}$. As the figures and the table indicate, the quality level remains within ± 0.3 units of $\hat{s}_{target}$ at all times; furthermore, the average level of quality is very close to the target value. This is the case for other values of $\hat{s}_{target}$ as well; for example, see Figure 116 where we show the average, minimum, and maximum values of $\hat{s}$ as a function of $\hat{s}_{target}$ for the same sequence. Here too, the average $\hat{s}$ follows $\hat{s}_{target}$ very closely. For all the values of $\hat{s}_{target}$ considered, the standard deviation of $\hat{s}$ was around 0.07–0.1 units. Therefore, even in cases where the scene content varies every few seconds, the CQ-VBR scheme is able to maintain a very consistent level of quality.

The variations in q in general follow the scene changes, although for some scenes there is a significant variation within the scenes as well; this is because of changes in the content within those scenes (e.g., zooming or panning of the camera).

8.3 Characterization of Traffic for CQ-VBR

In Figures 117 to 121, we show the number of bits per frame as a function of time for the five sequences, encoded at $\hat{s}_{target}$=$\{4.0,4.5\}$. It is interesting to note that in all cases except videoconferencing, the curves have some large "spikes" (i.e., bursts of short duration and of large magnitude compared to the remaining parts of the traffic). The spikes often occur due to a scene change, after which the feedback control takes some time to readjust. Some spikes also occur in the middle of a scene, but this is again due to the changes in the content within the scene. For example, such spikes occur around frame index 400 in Star Trek, where the video sequence contains a targeting computer display, and the target displayed on the screen is flashing. Despite such spikes, our constant-quality controller is able to produce a fairly consistent quality level as seen above. We have also experimented with other values of K_p, T_I, and T_D to determine if such spikes can be eliminated. However, we have found that for the values of PID coefficients which result in little or no spikes, the

system's response time becomes very slow, to the point that the quality can not be held constant across scene changes anymore. Therefore, the values of K_p, T_I, and T_D as given by the Ziegler-Nichols method are the most suitable ones for achieving the constant-quality objective. (However, as we will see in Section 8.6, it is possible to add a peak-rate controller to the system to reduce the magnitude of the peaks without compromising too much the resulting quality.)

In Figure 122, we show the frame size histograms for all five sequences, encoded at $\hat{s}_{target}$=4.5. As the figure indicates, the Commercials sequence exhibits a greater average and standard deviation of the frame sizes compared to the other sequences; the histograms for the other four sequences look relatively similar to each other. In Table 4, we show the average, standard deviation, maximum, and minimum values of frame sizes for $\hat{s}_{target}$=$\{$4.0,4.5$\}$. Among the five sequences, the Commercials sequence exhibits both the largest average and the largest variations in the traffic for both target quality values, with an average of 35 kbits per frame, a maximum frame size of 270.6 kbits, and a standard deviation of 25.7 kbits per frame for $\hat{s}_{target}$=4.5. For $\hat{s}_{target}$=4.5, the average and standard deviation of frame sizes for the Raiders sequence are also close to those for the Commercials; however, the Raiders sequence has a shorter tail, and its maximum frame size is 138.9 kbits. The Star Trek sequence has an almost as long a tail as the Raiders sequence, but smaller average and standard deviation values: 20.9 kbits and 13.8 kbits, respectively. For $\hat{s}_{target}$=4.0, the average frame size of the Raiders sequence is about 11.2 frames, which is about one-half of the average frame size of the Commercials sequence. For that $\hat{s}_{target}$ value, the Star Trek and Raiders sequences have very similar statistics. The Terminator 2 and Videoconferencing sequences have smaller average, standard deviation, and maximum values compared to the other three sequences for both $\hat{s}_{target}$ values.

The maximum-to-average frame size ratios range from about 1.6 for Videoconferencing to about 9 for Commercials. Among all five sequences, the average frame size varies by a factor of 2.4, and the maximum frame size varies by a factor of 13.

In Figure 123, we show the average frame size as a function of $\hat{s}_{target}$ for the five sequences. Particularly for the Commercials and Raiders (and to a smaller degree for the other sequences), the figure exhibits a knee around 4.4, beyond which the average number of bits per frame increases sharply.

In Figure 124, we show the frame size autocorrelation functions for all five sequences,

and for $\hat{s}_{target}$=4.5. The figure indicates that each sequence exhibits a different type of autocorrelation function. For example, for all the sequences except Raiders, the autocorrelation function drops below 0.4 within about 5-6 frames, while for the Raiders sequence, it takes about 20 frames. As another example, the autocorrelation for the Commercials sequence remains at 0.2 for at least 300 frame periods, while the others reach zero much earlier.

8.4 CQ-VBR Video Transmission over a Circuit

Now let us consider that a CQ-VBR encoded sequence is to be transmitted over a circuit of capacity C bits/second, and examine the delay characteristics. We assume again that the encoder outputs data one macroblock at a time. In Figure 125, we show $D(C,k)$ as a function of time for the Star Trek sequence, encoded at $\hat{s}_{target}$=4.5. We consider two values of C: 627 kb/s (i.e., the average rate for that sequence), and 1000 kb/s. It is clearly seen in the figure that for C=627 kb/s, the maximum delay is nearly 3 seconds, while for C=1000 kb/s, the maximum delay is reduced to about 500 ms[5].

In Figure 126, we show $\max_k\{D(C,k)\}$ versus C for the five sequences, encoded at $\hat{s}_{target}$=4.5. As the figure indicates, when C is about 2–3 times the average rate of the sequence, the maximum delay is relatively small (i.e., on the order of 100 ms). When C is decreased so as to be close to the average rate of the sequence, the delay becomes on the order of several seconds.

In addition to understanding the delay performance when the CQ-VBR video is transmitted over a circuit, these results are also useful in determining the peak rate that is to be negotiated during the call setup in an ATM network. If large delays are to be avoided, a peak rate of 2–3 times the average seems appropriate.

8.5 Results for Other Encoding Schemes

A. MPEG-1

In MPEG, the design of the feedback function depends on the GOP structure. The reason is that the B frames depend on the information in the future frames. Therefore, when

[5]Note that if the video is stored, it can be transmitted starting at a different point than the first frame. In that case, the delays would have been somewhat different. Investigation of that case is for further study.

there are B frames, the frames are encoded at a different order compared to how they are displayed. As a result, the quality metric $\hat{s}$ cannot be computed at every frame interval; instead, it can only be computed at those time instants where the last B frame before an I or P frame is encoded. Therefore, the maximum possible sampling rate of the controller is equal to the number of consecutive B frames plus one—for GS1 it is equal to one-third of the frame rate, and for GS2 it is equal to the frame rate. Since higher sampling rates result in better performance in a feedback controller, we use the above specified sampling rates in the design of our controller.

We have designed the appropriate PID feedback control function for MPEG-1 with these GOP structures by using the same approach as in H.261. In Figure 127, we show the quantizer scale q as a function of time for the videoconferencing sequence, proportionally controlled with various values of K_p; part (a) of the figure is for GS1, and part (b) is for GS2. It is clear from the figure that for GS1 the continuous oscillations start when the gain K_p is around 10, and for GS2 they start when K_p is around 13. Therefore, for GS1, K_u=10, and P_u=12 frame intervals, and for GS2, K_u=13, and P_u=4 frame intervals. Note that, when the sampling rate of a system is low, it is recommended to use a T_D coefficient larger than that suggested by the Ziegler-Nichols method [27]. For GS1, we have therefore experimented with various values of T_D, and found that $T_D = P_u/4$ gives the best results. Therefore, the PID coefficients for GS1 are given as $K_p = 6$, $T_I = 2T$, and $T_D = T$ (where T=100 ms), and for GS2, they are given as K_p=8, T_I=2T, and T_D=0.5T (where T=33 ms). We have also experimented with various values of w, and determined that $w = 3$ frames gives the best results for both GOP structures (similarly to the H.261 controller).

As far as the traffic and quality characterization of MPEG CQ-VBR sequences, first consider the GOP Structure 1. In Figure 128, we show $\hat{s}$ versus time for Star Trek, MPEG, GS1, CQ-VBR, $\hat{s}_{target}$=4.5. Clearly, despite the lower sampling rate, the CQ-VBR scheme is still able to maintain a consistent level of quality. In Figure 129, we plot the corresponding q versus time. Here, the q values change between 3 and 14, and variations in q are somewhat different from those in Figure 109, owing to the different feedback functions, and the differences between the H.261 and MPEG encoding schemes. In Figure 130, we plot the frame size versus time for the same sequence. Here too, there are some occasional spikes due to the changes in the content. What is also interesting is that the average number of bits per frame in this case is 24.5 kbits, as compared to 20.9 kbits in H.261. This is mainly

because of the slower sampling rate in the feedback function, which sometimes results in more bits than necessary to be encoded to achieve the required quality objective.

Now consider the GOP Structure 2. In Figure 131, we plot q versus time for Star Trek, MPEG, GS2, CQ-VBR, $\hat{s}_{target}$=4.5. Here, the q values range from 3 to 16, and generally follow a similar outlook to the q values for the GS1 case. In Figure 132, we show the corresponding frame size as a function of time. It is interesting to note that while the maximum frame size in GS2 is about the same as in GS1, the average frame size in GS2 is 19.8 kbits, which is smaller than the average frame size in GS1, and about the same as in H.261. This suggests that it is indeed the slow sampling rate in GS1 which causes the excess number of bits produced.

B. Motion-JPEG

For Motion-JPEG encoded sequences, we have also applied the same approach to design the CQ-VBR feedback function. We have determined that K_u=100 and $P_u = 8T$, where T =33 ms. Therefore, K_p=60, T_I=4T, and T_D=T. We have determined that w=3 gives the best results for this case as well. Also note that in Motion-JPEG, when q=20, the quality is near perfect at all times; if q is decreased beyond that point, the number of bits produced increases significantly without bringing in any benefit. Therefore, we have limited the q values to be greater than or equal to 20 at all times.

We have observed that for Motion-JPEG too, the quality can be held fairly consistent over time using the CQ-VBR scheme; furthermore, the quality statistics in general are very similar to those for H.261 and MPEG.

In Figure 133, we show the number of bits per frame versus time for the Star Trek sequence, CQ-VBR encoded at $\hat{s}_{target}$=4.5 using the Motion-JPEG scheme. For this sequence, the average number of bits per frame is 57.3 kbits (more than twice as large as in the corresponding H.261 and MPEG sequences), and the maximum number of bits per frame is 140 kbits, about the same as in the corresponding H.261 and MPEG sequences. The short-term peaks are not present in this case, and the variations in the frame sizes are observed mainly on a scene by scene basis.

8.6 Joint Peak Rate and Quality Controlled VBR

So far, we have seen that the CQ-VBR scheme is able to maintain a constant level of quality, but there are some spikes in the resulting traffic with a peak magnitude as large as 5-10 times the average rate of the encoded sequence, especially for H.261 and MPEG-1. These spikes typically last about 10-15 frames. Such spikes can be detrimental if the network does not have enough bandwidth or buffers to absorb them, or if the buffering of the excess bits would result in excessive delay. Thus, in some cases the peak rate of the traffic may have to be kept under a certain limit. In order to achieve this, we introduce a modification to the CQ-VBR scheme, referred to as *Joint Peak Rate and Quality Controlled VBR (JPQC-VBR)*. The block diagram for this scheme is depicted in Figure 134, and it can be described as follows. In addition to the CQ-VBR feedback loop, we consider another feedback loop which operates like the CBR feedback, with a given V and B. The two feedback loops operate concurrently and each of them produces a q value; then, the maximum of those two q values is selected. Therefore, as long as the bit rate required to achieve the target quality objective is less than V, the video is encoded according to the constant quality feedback as before. When it is greater than V, the CQ-VBR feedback is disabled, and the CBR feedback is activated, which aims to maintain the target bit rate at V.

In this scheme, as a result of the short-term variability in the CBR encoded video traffic, there would still be some frames which are produced at a rate somewhat greater than V. The size of these frames depend on the video content, and on B. Therefore, by choosing V and B appropriately, a maximum delay objective $\max_k\{D(C,k)\}$ may be met at all times when the JPQC-VBR encoded video is sent over a circuit with bandwidth C. Clearly, V must be chosen such that $V \leq C$. An obvious choice of V and B are $V{=}C$, and $B{=}C \times \max_k\{D(C,k)\}$. One may also choose a smaller V and a larger B to achieve the same delay objective; however, we have observed that in this case the resulting quality exhibits more deviations from $\hat{s}_{target}$. For example, consider that the Star Trek sequence is to be encoded with $\hat{s}_{target}{=}4.5$, under the constraint that $C{=}1536$ kb/s, and $\max_k\{D(C,k)\}$ $\leq$100 ms. We have encoded this sequence using the JPQC-VBR scheme, with the (V,B) pairs of (1536 kb/s,153.6 kbits), (1024 kb/s,1024 kbits), and (768 kb/s,1536 kbits). (The B values for $V{=}1024$ kb/s and $V{=}768$ kb/s are experimentally found such that they are the largest possible values for which the delay objective is met.) In Table 5, we show the corresponding average, standard deviation, minimum, and maximum values of quality. In the table,

the quality statistics for the CQ-VBR encoded sequence are also shown for comparison purposes. The average and the standard deviation of quality do not change very much with the particular values of V and B; furthermore, these quality statistics are very close to those for the CQ-VBR case. However, the minimum quality decreases from 4.20 for (V=1536 kb/s,B=153.6 kbits) to 4.02 for (V=768 kb/s,B=1536 kbits).

In Figure 135, we show the number of bits per frame as a function of time for the JPQC-VBR scheme, for $\hat{s}_{target}$=4.5, V=1536 kb/s, and B=153.6 kbits. Comparing this figure with Figure 117 (b), we see that the peaks in the traffic have a magnitude about one-half of those in the CQ-VBR case. However, other than the portions with the large peaks, the two sequences look very similar. Furthermore, the average number of bits per frame is equal to 20900 for the CQ-VBR, and it is equal to 20570 for the JPQC-VBR; thus, with the JPQC-VBR scheme the average rate is decreased only by 1.6%, while the peak rate is decreased by a factor of two.

In Figure 136, we plot $\max_k\{D(C,k)\}$ versus C for the Star Trek sequence, encoded using the JPQC-VBR scheme with $\hat{s}_{target}$=4.5, V=1536 kb/s, and B=153.6 kbits, as well as the CQ-VBR scheme with $\hat{s}_{target}$=4.5. As the figure indicates, the maximum delay at C=1.5 Mb/s is very small (equal to 13 ms) for the JPQC-VBR scheme, while for the CQ-VBR scheme it is equal to about 180 ms. Therefore, the JPQC-VBR scheme is indeed able to reduce the peaks in the traffic, and therefore the associated delays, while still maintaining the quality at a consistent level.

In Table 6, we show for all five sequences the average, standard deviation, minimum, and maximum values of quality for the JPQC-VBR scheme, for $\hat{s}_{target}$=4.5, V=1536 kb/s, and B=153.6 kbits. Comparing this table with Table 3, we observe that the average quality remains about the same for all the sequences considered, despite the peak rate control. However, for the Commercials sequence, the minimum quality decreases from 4.33 to 3.98. For the other sequences, the decrease in minimum quality is insignificant. In fact, for the Videoconferencing sequence, there is no change in any of the quality statistics; this is because the number of bits per frame for this sequence is already relatively small, thus rarely requiring peak rate control.

In Table 7, we show for all five sequences the average, standard deviation, minimum, and maximum values of frame sizes for the JPQC-VBR scheme, again for $\hat{s}_{target}$=4.5, V=1536 kb/s, and B=153.6 kbits. Comparing this table with Table 4, it can be seen that

the average frame size remains very close to the CQ-VBR case for all the sequences, but the peaks are significantly reduced.

As far as MPEG encoded sequences are concerned for the JPQC-VBR scheme, we have found similar results.

8.7 Comparison of CBR, OL-VBR, and CQ-VBR Schemes

The criteria by which the performance of various encoder control schemes is compared depends on the scenario considered. For example, if the encoded video is first stored and then played back locally, then two appropriate criteria of comparison would be the quality level of the video, and the total number of bits required to store the encoded sequence (or equivalently, the average data rate). As another example, if several video streams are statistically multiplexed to be sent over, say, an ATM network, then appropriate performance criteria may be the quality of video at the receiver (which is now affected by both the quality degradations due to video encoding, and due to cell losses in the ATM network), and the number of video streams that can be multiplexed over a channel of a given bandwidth, under a given delay constraint. In this case, one may define a statistical multiplexing gain as the ratio of the number of VBR streams to the number of CBR streams that the multiplexer can accommodate given the type of video content, a certain end-to-end delay constraint, a minimum level of video quality that should be maintained at all times, and a certain network capacity. A full comparison of the encoder control schemes in terms of statistical multiplexing gain is out of scope of this report. However, we note that the average data rate is an interesting performance measure for this scenario as well, since the ratio of the VBR and CBR average data rates for a given minimum level of video quality represents an upper limit on the statistical multiplexing gain. Furthermore, we also consider here the simple multiplexing scenario that we had examined earlier for CBR.

Given these considerations, here we first compare the quality statistics for CBR, OL-VBR, and CQ-VBR schemes when in each scheme the resulting average data rate is the same. Then we compare the resulting average data rates for the three encoder control schemes given the constraint that the quality must be at least equal to a given $\hat{s}_{min}$ at all times. Finally, we give a comparison of delays encountered by the CBR and CQ-VBR streams in the simple statistical multiplexing scenario.

In Table 8 we show for the five sequences the average, standard deviation, minimum, and

maximum $\hat{s}$ when the sequences are CBR and OL-VBR encoded, using the same average rate as their CQ-VBR counterparts. For the CBR sequences, a rate control buffer size B=384 kbits is used, which is large enough so that any larger B would not have resulted in a better quality. Part (a) of the figure is for the same average rate as the CQ-VBR sequences with $\hat{s}_{target}$=4.0, and part (b) is for the same average rate as as the CQ-VBR sequences with $\hat{s}_{target}$=4.5.

Compared to the CQ-VBR sequences encoded at $\hat{s}_{target}$=4.0, the corresponding CBR and OL-VBR sequences exhibit a quite larger variation in quality. In particular, the minimum $\hat{s}$ values for the Commercials, Raiders, and Star Trek sequences are very low for CBR and OL-VBR, and accordingly, the span between the minimum and maximum $\hat{s}$ values are significantly higher than those for CQ-VBR. By contrast, for the Videoconferencing sequence, the $\hat{s}$ statistics are very similar for all three schemes. This is as expected, since the contents of this sequence do not vary significantly over time.

For the CQ-VBR sequences encoded at $\hat{s}_{target}$=4.5, the corresponding CBR and OL-VBR sequences again exhibit a larger variation, although not as large as those in part (a) of the figure. In this case, smaller quantizer scales are used in order to achieve a better quality; for such small quantizer scales the variations in quality are smaller. However, it is also important to note that the average frame size required to achieve a quality level of 4.5 varies by a factor of 2.5 from the Videoconferencing sequence to the Commercials sequence. If the CBR scheme is used for encoding and transmitting, say, a TV program, then a high bit rate must be chosen to ensure that all the scenes are encoded without excessive quality degradation; the same bit rate must also be used for those scenes which could be encoded at a lower bit rate and still incur a small quality degradation. By contrast, the CQ-VBR scheme allows the bit rate to be automatically adjusted; thus the given quality objective is maintained using only as many bits as required. For example, if the five sequences considered here were to be sent back-to-back using the CQ-VBR scheme, then the resulting average rate would be about 700 kb/s. On the other hand, in order to maintain a similar level of quality at all times using the CBR scheme, the sequences would have to be encoded at a rate of about 1.8 Mb/s.

We now compare the average rates resulting from encoding the CBR, OL-VBR, and CQ-VBR sequences such that they maintain a given $\hat{s}_{min}$. In Table 9, we show for the CQ-VBR the $\hat{s}_{min}$ and the resulting average rate, as well as the corresponding average

rates for the CBR and OL-VBR schemes for the same $\hat{s}_{min}$ (± 0.1 quality impairment units). Part (a) of the table is for the CQ-VBR target quality $\hat{s}_{target}$=4.0, and part (b) is for $\hat{s}_{target}$=4.5. Part (a) of the table indicates that the average rates between CQ-VBR and CBR differ by a factor of 2 for the Commercials sequence; they differ by about 1.5 for Raiders and Star Trek sequences, and there is very little difference for the Terminator-2 and Videoconferencing sequences. The OL-VBR average rates are somewhere in-between those of CQ-VBR and CBR. Therefore, for sequences such as Videoconferencing or Terminator-2, where the content does not vary significantly over time, no statistical multiplexing gain of VBR over CBR should be expected. On the other extreme, for the Commercials sequence, a statistical multiplexing gain up to a factor of 2 can be achieved. Therefore, the variability of content is very important in determining which video encoder control scheme is the most appropriate. As for the part (b) of the table, there is a factor of 1.7 difference between CQ-VBR and CBR average data rates for the Commercials sequence, but for the other four sequences the average data rates do not differ significantly from one encoder control scheme to another. In fact, for the Raiders, the CBR and OL-VBR encoded sequences have a smaller average rate compared to the CQ-VBR encoded sequence; this is because $\hat{s}_{min}$=4.2 for the CBR and OL-VBR sequences, which results in a much smaller average rate compared to $\hat{s}_{min}$=4.3. Here too, we reiterate the point that if these sequences were to be combined into a single sequence, they could be sent at a rate about 2.5 times smaller using the CQ-VBR scheme compared to the CBR scheme.

Now consider that a number of CQ-VBR and CBR streams are statistically multiplexed over a circuit of bandwidth W, where we make the same assumptions as in Section 6 about the operation of the system. As an example case, we consider the Commercials sequence, CQ-VBR encoded at $\hat{s}_{target}$=4.5, and the corresponding CBR sequence which gives the same $\hat{s}_{min}$ (i.e., V=1800 kb/s, B=768 kbits). We consider multiplexing a number N_v of each type of sequence over the network, and compare the resulting delays. The maximum delay histograms are shown in Figure 137 for the case of $W = 2000\ N_v$ kbits/s, and N_v=$\{1,2,4,8,16\}$. It is interesting that until N_v=16, the maximum delay for the CBR are somewhat smaller compared to CQ-VBR; beyond that point, the maximum delay for the CQ-VBR becomes smaller.

9 Conclusions

In this report, we have characterized the quality, delay (when being transmitted over a circuit), and traffic for CBR, OL-VBR, and CQ-VBR encoded video for several video contents with different spatial and temporal characteristics, encoded using the H.261, MPEG-1, and Motion-JPEG standards. As far as video quality is concerned, we have used a quantitative video quality measure developed at ITS, which correlates well with subjective evaluations.

As far as CBR encoded sequences are concerned, the main issue is to select the target bit rate V, and the rate control buffer size B appropriately so that the applications' quality and delay requirements can be met. We have determined that for a given V, increasing B increases the quality up to a certain point, beyond which it remains fairly constant. On the other hand, increasing B also increases the rate control buffer delay. Therefore, it does not pay off to increase the buffer size beyond the point where the quality reaches its plateau. Likewise, for a given B, increasing V also increases the quality up to a certain point, beyond which the quality remains constant. Accordingly, there is a trade-off between B and V to achieve the same quality. For some video sequences, the equal quality contours in the B–V space exhibit a very sharp knee; then, a good choice of B and V is at the knee, achieving near-minimum values for both B and V. We have also shown that the quality for given V and B depends significantly on the type of video content.

We have also considered transmission of CBR-encoded video over a circuit, where the main issue is characterization of delay. We have first considered that the circuit bandwidth is equal to V. We have shown that for a given V, the delay increases in a nearly linear fashion as B is increased. Furthermore, as V is increased, a greater B can be used while still meeting the delay constraint. We have also shown that there is an optimum (B, V) pair which meets given quality and delay objectives while producing the fewest number of bits; however, this optimum pair depends significantly on the video content.

We have also shown that when the end-to-end delay requirement is very stringent, using a rate V smaller than the bandwidth of the circuit over which the CBR stream can result in a better quality by allowing a greater buffer size to be used.

Finally for CBR, we have presented some traffic statistics, and shown that the fluctuations in the bit rate are relatively small and short-lived; this suggests that statistical multiplexing of CBR streams in order to reduce the end-to-end delay would be beneficial.

Indeed, we have simulated a simple scenario where multiple CBR streams are statistically multiplexed over a circuit, and shown that even when the number of CBR streams being multiplexed is as small as 2, there is a substantial reduction in the delay.

The main problem with the CBR scheme is that it is not possible to determine the appropriate V and B values to maintain a target quality level without any apriori knowledge of the video sequence being encoded. This may be overcome by classifying video sequences into some appropriate categories, and determining the B and V values for each category. Alternatively, a conservative approach is to choose B and V large enough to accommodate even a worst case; this is of course fairly inefficient in terms of network resources.

As far as OL-VBR encoded sequences are concerned, we have determined that for small q_0 values (i.e., up to 8–10), the quality is typically very good; when the q_0 is increased, the average and minimum values of quality decrease linearly, but at a rate that depends on the content. Furthermore, the variability in quality increases as q_0 is increased. On the other hand, the traffic is highly variable according to the content for small q_0 values. For larger q_0 values, the traffic is still variable, but to a smaller extent. Thus, for OL-VBR, it is not possible to select an appropriate q_0 value to meet certain quality and traffic rate objectives. If the quality is the main objective, q_0 may be chosen small enough (e.g., 3 or 4), but the resulting traffic rate becomes large, and highly variable.

As far as CQ-VBR sequences are concerned, we have demonstrated that indeed the quality can be maintained at a very consistent level at all times, even when there are frequent scene changes. We have shown that at the same average rate, the CQ-VBR scheme can maintain a better quality compared to the CBR and OL-VBR schemes, especially when the scene content is highly variable in time. Furthermore, for a given minimum level of quality, the CBR scheme needs to use a data rate up to twice as much as the average CQ-VBR rate; this suggests that up to a factor of two statistical multiplexing gain can be obtained using the CQ-VBR scheme. However, the CQ-VBR traffic occasionally contains bursts of relatively high magnitude (5–10 times the average) but short duration (5–15 frames). We have therefore devised the Joint Peak Rate and Quality Controlled VBR scheme, where in addition to the quality, the peak rate of the traffic is also controlled, by means of a rate control buffer as in CBR. We have shown that with the JPQC scheme, it is possible to achieve near-constant video quality while keeping the peak rate within 2–3 times the average rate.

Note that the same approach taken here to achieve constant-quality video encoding could also be applied to MPEG-2. However, the ITS metric is not appropriate for MPEG-2, as it has been calibrated and validated by subjective viewers for a level of quality achieved by MPEG-1 and H.261. Therefore, another quality measure which can work for MPEG-2 is needed, given the better quality achieved and the higher viewer expectation. Such a measure has recently been developed, and is presented in [28] (referred to as MPQM). The suitability of this measure for assessing MPEG-2 quality is examined in [29]. Also note that MPEG-2 presents a richer set of parameters controlled, thus allowing a more precise control of quality. We are currently working on designing a CQ-VBR scheme for MPEG-2 using the MPQM, and considering a variety of parameters to be controlled. As a first step, we have designed the feedback function for the case where again the quantizer scale q is the controlled parameter [30]. For the H.261, MPEG-1, and Motion-JPEG schemes, the MPQM scheme may also be used for achieving constant quality; we leave this as future work.

References

[1] "Video CODEC for Audiovisual Services at p×64 kbit/s," ITU-T Recommendation H.261, (Geneva, 1990).

[2] "ISO/IEC 11172, Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s," International Organization for Standardization (ISO), Nov. 1991.

[3] "ISO/IEC JTC1/SC2/WG10, JPEG Draft International Standard DIS 10918-1," International Organization for Standardization (ISO), Oct. 1991.

[4] "Draft ANSI Standard Specification of Video Performance Terms and Definitions," ANSI T1A1.5/94-137 R1, Oct. 3, 1994.

[5] A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Voran, and S. Wolf, "An Objective Video Quality Assessment System Based on Human Perception," in *SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, (San Jose, CA), pp. 15–26, Feb. 1993.

[6] İ. Dalgıç and F. A. Tobagi, "Performance Evaluation of Networks Carrying Constant and Variable Bit Rate Video Traffic," Submitted to *IEEE JSAC*, special issue on Real-Time Video Services in Multimedia Networks.

[7] İ. Dalgıç and F. A. Tobagi, "Performance Evaluation of Ethernets and ATM Networks Carrying Multimedia Traffic," Computer Systems Laboratory Technical Report, Stanford University (in preparation).

[8] A. R. Reibman and A. W. Berger, "On VBR video teleconferencing over ATM networks," in *IEEE GLOBECOM '92*, pp. 314–319, 1992.

[9] D. P. Heyman, A. Tabatabai, and T. Lakshman, "Statistical Analysis and simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Trans. Circ. and Sys. Video Tech.*, vol. 2, pp. 49–59, Mar. 1992.

[10] D. Heyman and T. Lakshman, "Source models for VBR broadcast-video traffic," in *IEEE INFOCOM '94*, pp. 664–671, 1994.

[11] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission," *IEEE Communications Magazine*, vol. 32, pp. 54–66, May 1994.

[12] E. Knightly and H. Zhang, "Traffic Characterization and Switch Utilization Using a Deterministic Bounding Interval Dependent Traffic Model," in *IEEE INFOCOM '95*, (Boston, MA), pp. 1137–1145, Apr. 1995.

[13] M. Krunz, R. Sass, and H. Hughes, "Statistical Characteristics and Multiplexing of MPEG Streams," in *IEEE INFOCOM '95*, pp. 455–462, 1995.

[14] D. Reininger and D. Raychaudhuri, "Bit-rate Characteristics of a VBR MPEG Video Encoder for ATM Networks," in *IEEE ICC '93*, pp. 517–521, 1993.

[15] O. Rose, "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems," Research Report Series 101, University of Wurzburg, Institute of Computer Science, Feb. 1995.

[16] M. Grasse, J. Arnold, and M. Frater, "Statistics of Variable Bit Rate Video Coders," in *Sixth International Workshop on Packet Video*, pp. D.5.1–D.5.4, 1994.

[17] M. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," in *ACM Sigcomm'94*, (London, UK), Oct. 1994.

[18] H. Gaggioni, "The Evolution of Video Technologies," *IEEE Communications Magazine*, vol. 25, pp. 20–36, Nov. 1987.

[19] "ISO/IEC 13818, Generic Coding of Moving Pictures and Associated Audio Information," International Organization for Standardization (ISO), 1994.

[20] "The PVRG MPEG-1, H.261, and JPEG encoders are available via anonymous ftp from havefun.Stanford.EDU."

[21] D. L. Mills, "Network Time Protocol (Version 2) — Specification and Implementation," Network Working Group Request for Comments RFC 1119, University of Delaware, Sept. 1989.

[22] "CCIR Recommendation 500-3, Method for the Subjective Assessment of the Quality of Television Pictures," CCIR, 1986, XVI'th Plenary Assembly, Volume XI, Part 1.

[23] A. K. Jain, *Fundamentals of Digital Image Processing.* Prentice Hall, Englewood Cliffs, NJ 07632, 1989.

[24] "Description of the Reference Model 8," CCITT SG XV. Spec. Group on Coding for Visual Telephony, May 1989.

[25] "MPEG Video Simulation Model Three (SM3)," ISO-IEC/JTC1/SC2/WG8, 1990.

[26] A. C. Luther, *Digital Video in the PC Environment.* Intertext Publications, McGraw-Hill Book Company, New Yor, NY, second ed., 1991.

[27] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems (second edition).* Addison-Wesley Publishing Company, 1990.

[28] C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System," in *Proceedings of the IS&T Symposium on Electronic Imaging: Science and Technology,* Digital Video Compression: Algorithms and Technologies 1996, (San Jose, CA), The Society for Imaging Science and Technology, Jan. 1996.

[29] A. Basso, İ. Dalgıç, F. A. Tobagi, and C. J. van den Branden Lambrecht, "Study of MPEG-2 Coding Performance based on a Perceptual Quality Metric," in *Proceedings of the 1996 Picture Coding Symposium*, (Melbourne, Australia), Mar. 1996.

[30] A. Basso, İ. Dalgıç, F. A. Tobagi, and C. J. v. d. B. Lambrecht, "A Feedback Control Scheme for Low Latency Constant Quality MPEG-2 Video Encoding," submitted for publication in proceedings of *EOS/SPIE Digital Compression Technologies and Systems for Video Communications*, Oct. 7-11,1996 Berlin, Germany.

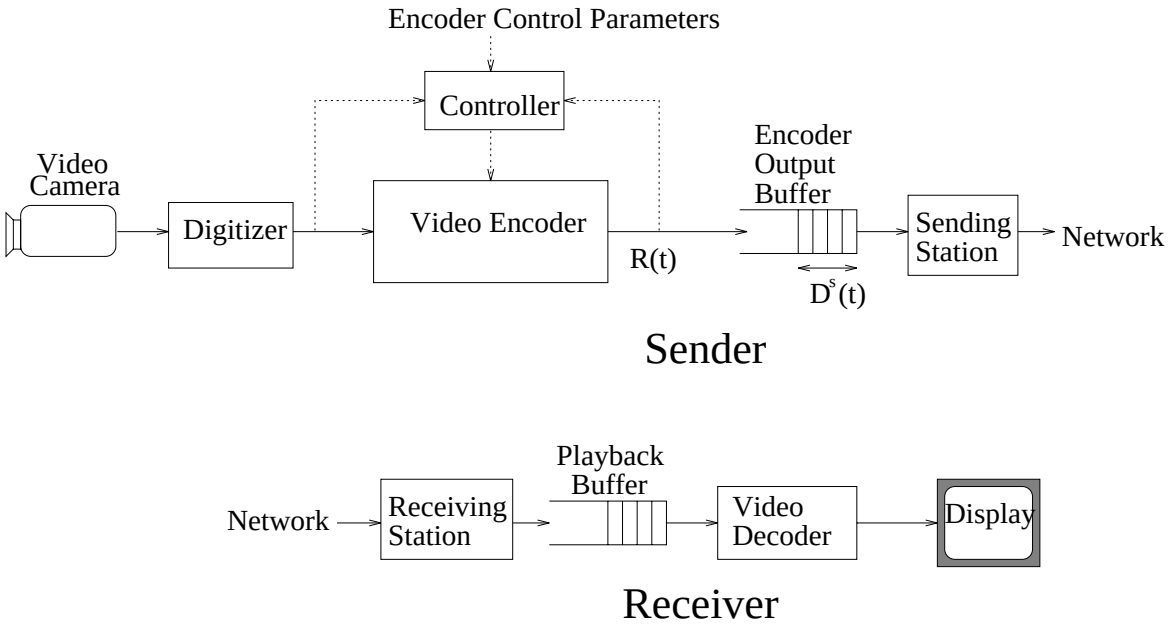


Figure 1: Block diagram of the system under consideration.

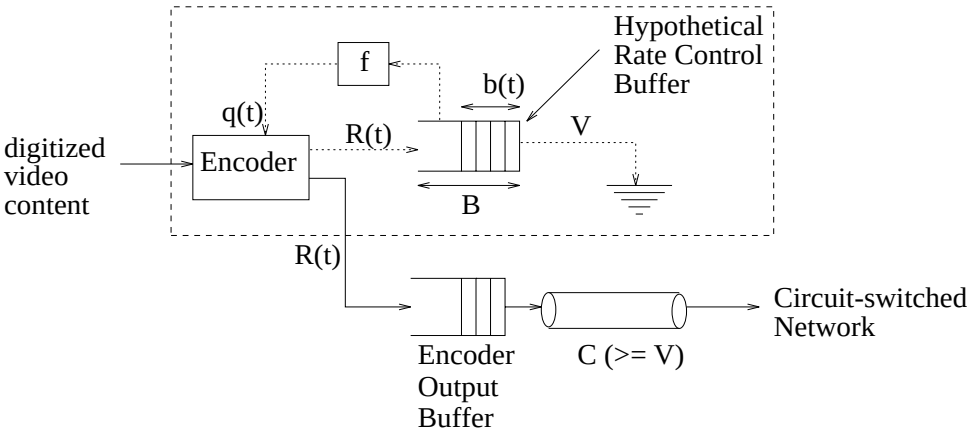


Figure 2: Block diagram of an encoder controlled according to the CBR scheme.

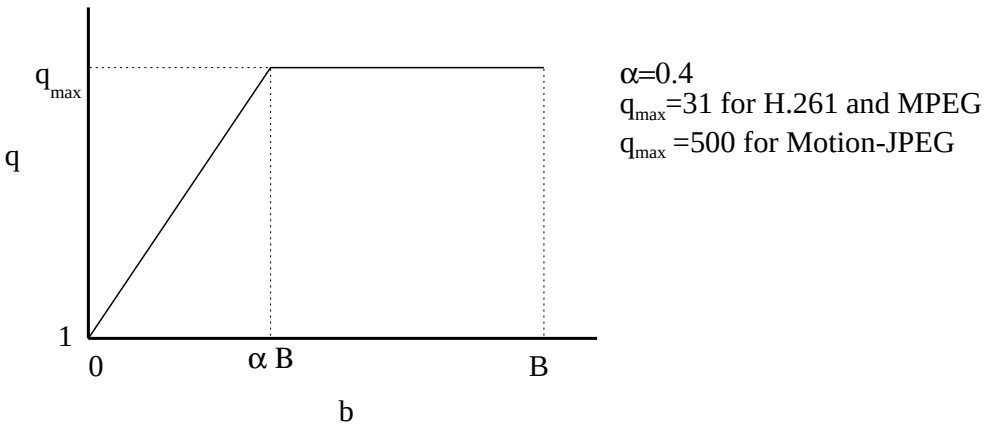
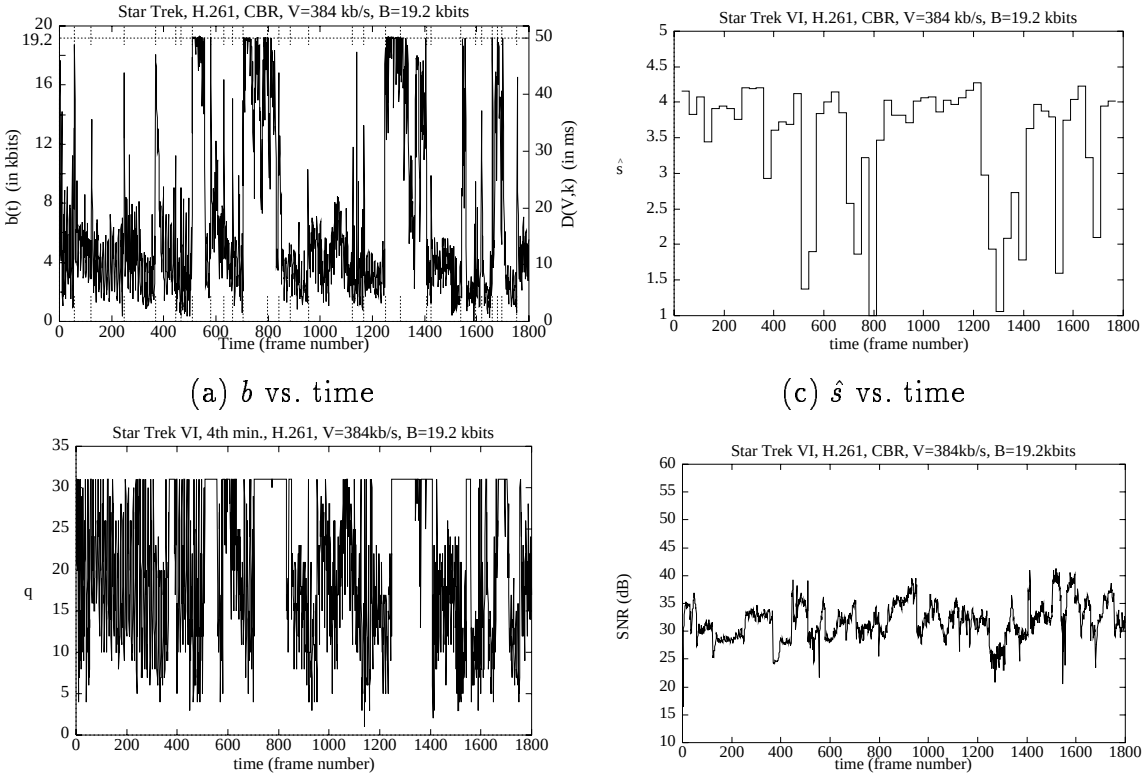


Figure 3: CBR feedback function



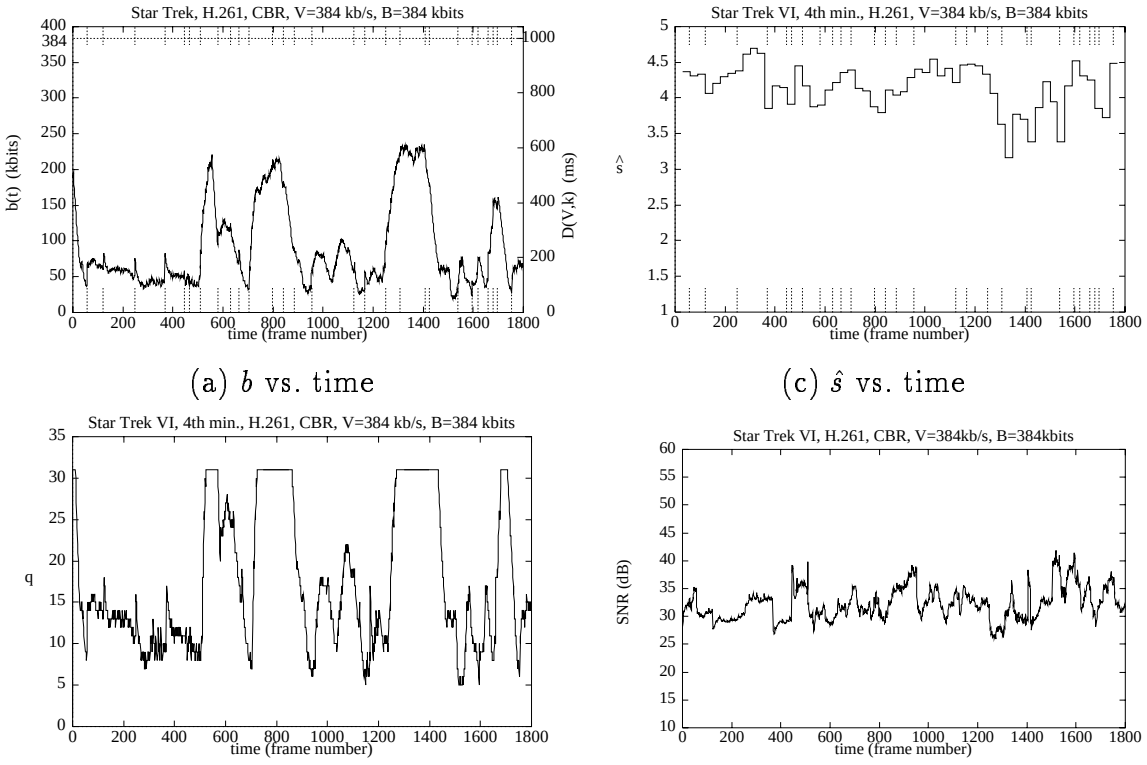
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

(d) SNR vs. time

Figure 4: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=384 kb/s, B=19.2 kbits.



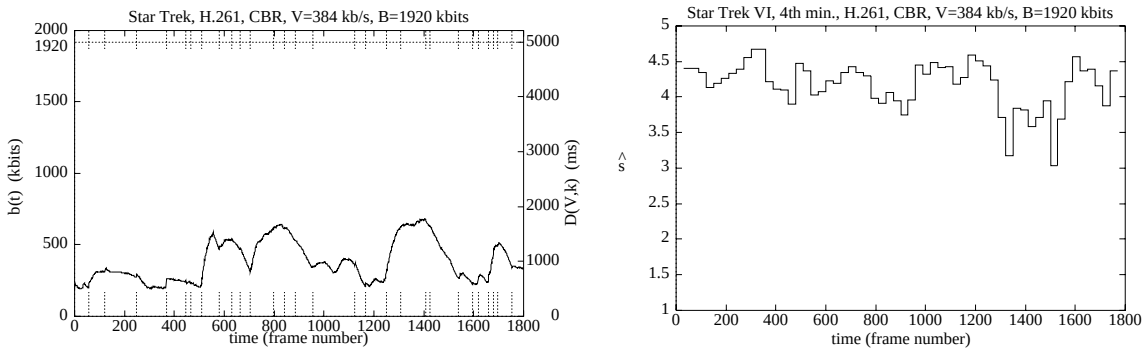
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

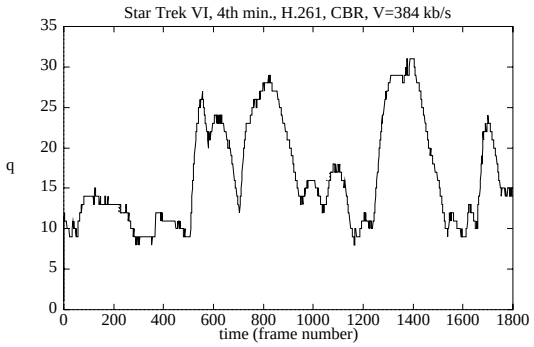
(d) SNR vs. time

Figure 5: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=384 kb/s, B=384 kbits.

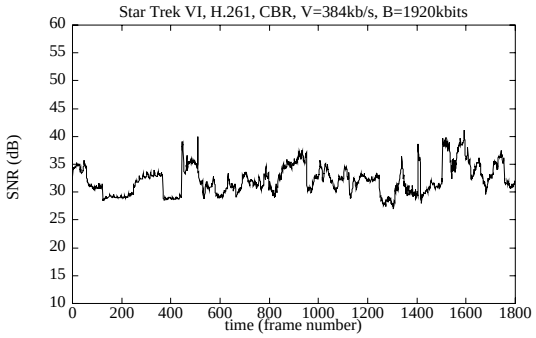


(a) b vs. time

(c) $\hat{s}$ vs. time

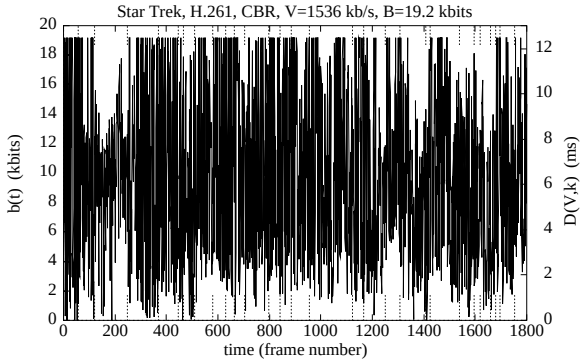


(b) q vs. time

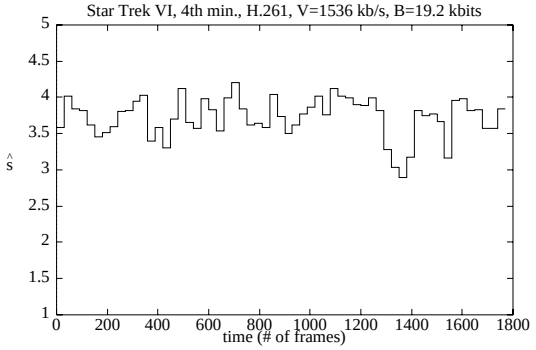


(d) SNR vs. time

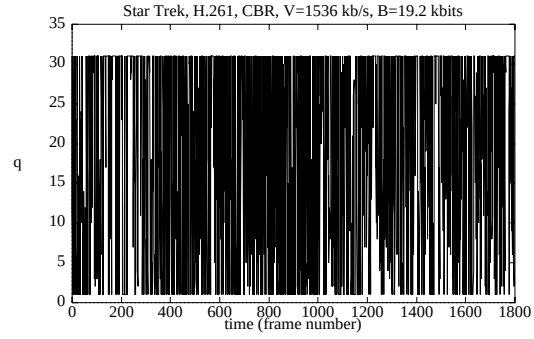
Figure 6: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=384 kb/s, B=1920 kbits.



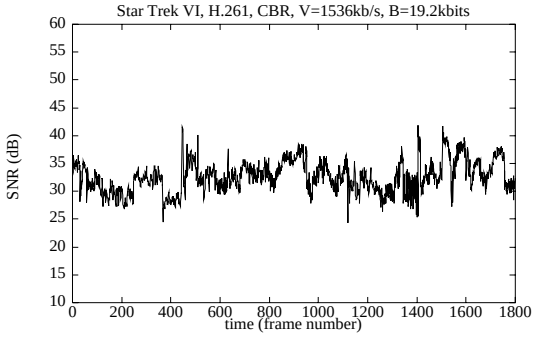
(a) b vs. time



(c) $\hat{s}$ vs. time

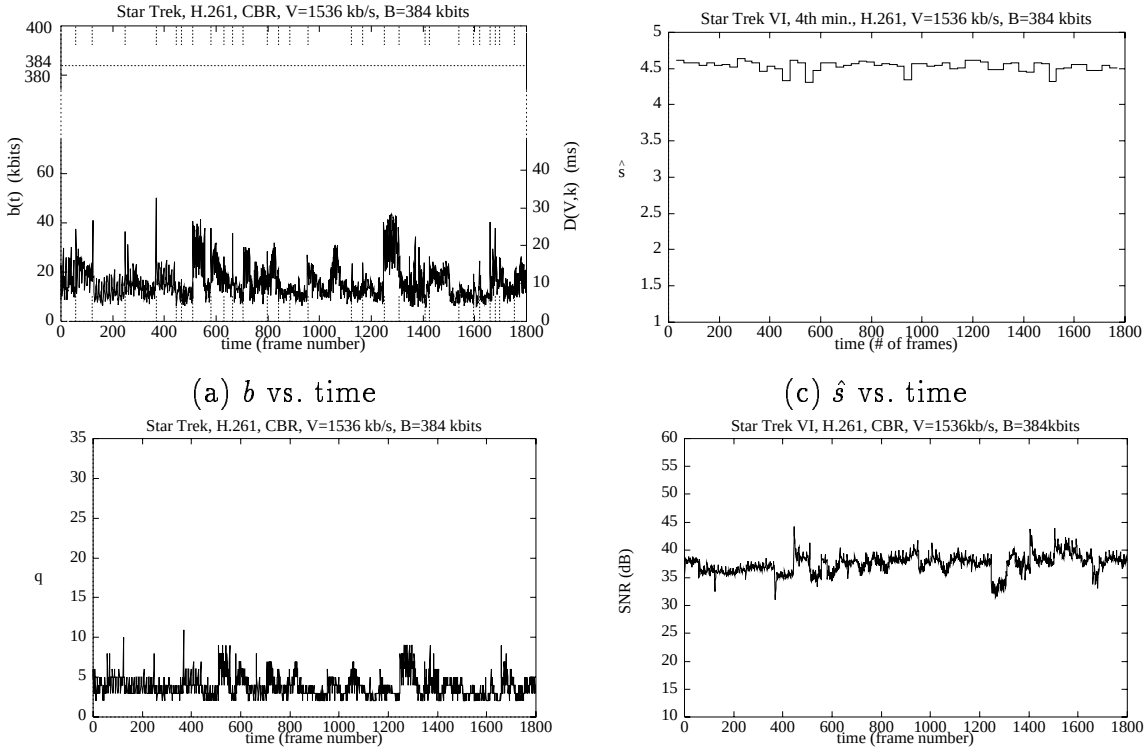


(b) q vs. time



(d) SNR vs. time

Figure 7: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=1536 kb/s, B=19.2 kbits.



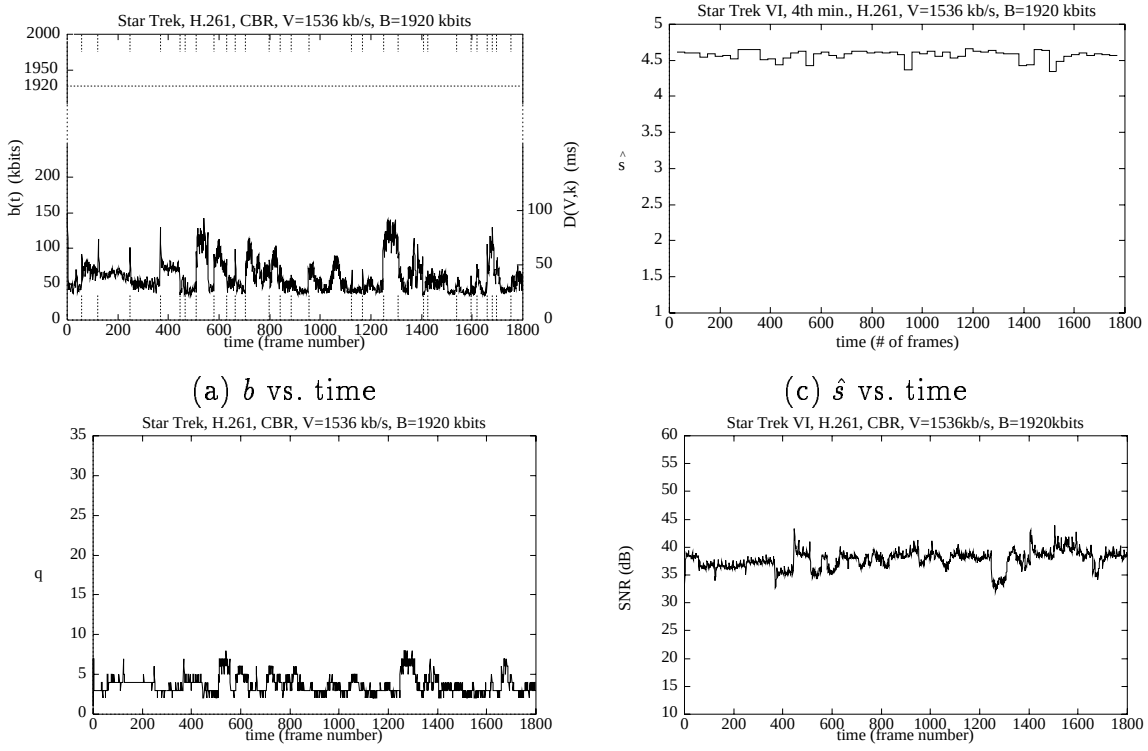
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

(d) SNR vs. time

Figure 8: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=1536 kb/s, B=384 kbits.



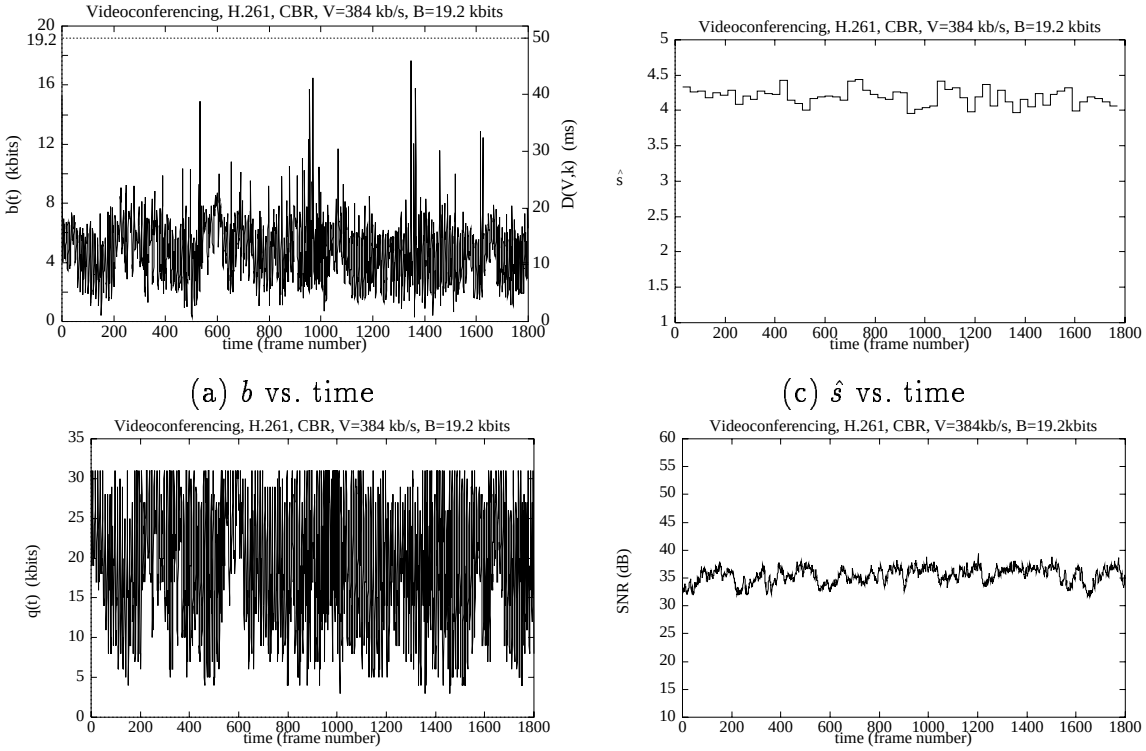
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

(d) SNR vs. time

Figure 9: b, q, $\hat{s}$, and SNR vs. time for the Star Trek sequence, H.261, CBR, V=1536 kb/s, B=1920 kbits.



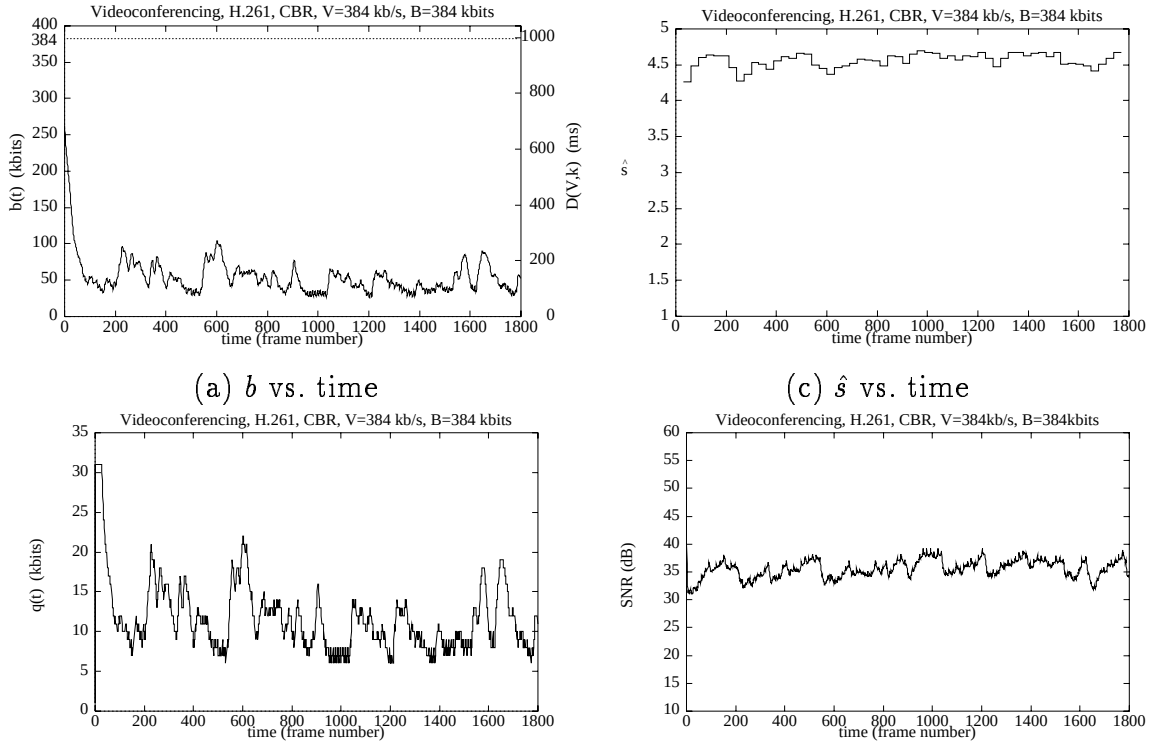
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

(d) SNR vs. time

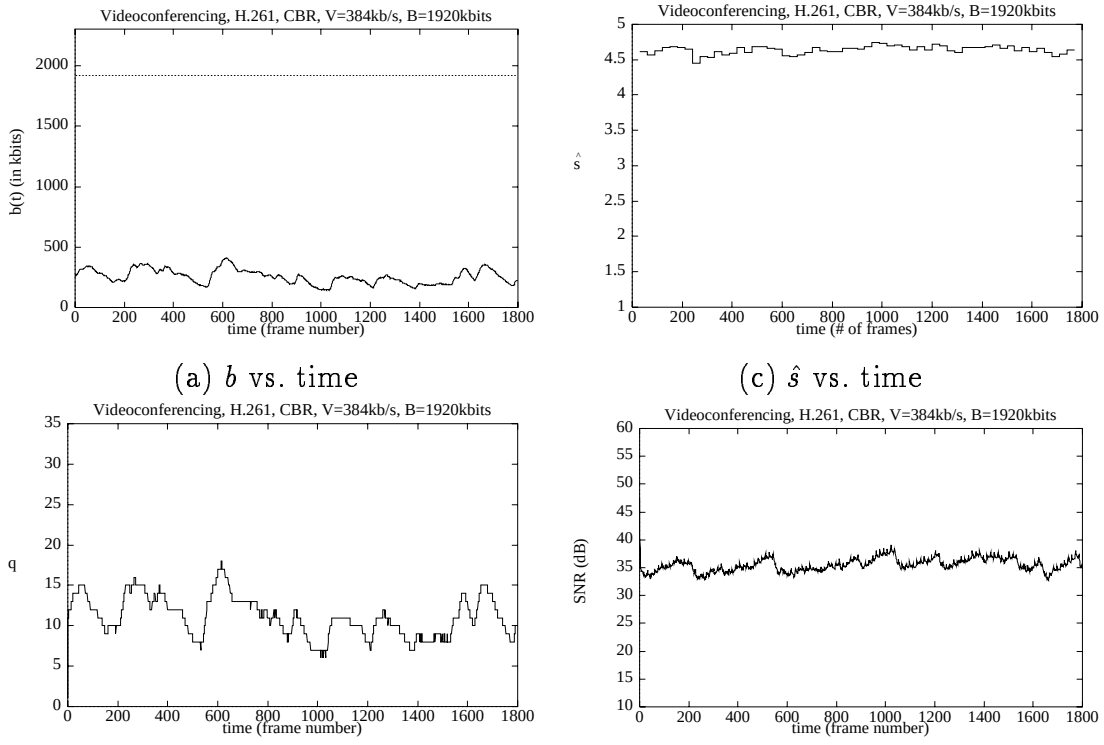
Figure 10: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=384 kb/s, B=19.2 kbits.



(a) b vs. time (c) $\hat{s}$ vs. time

(b) q vs. time (d) SNR vs. time

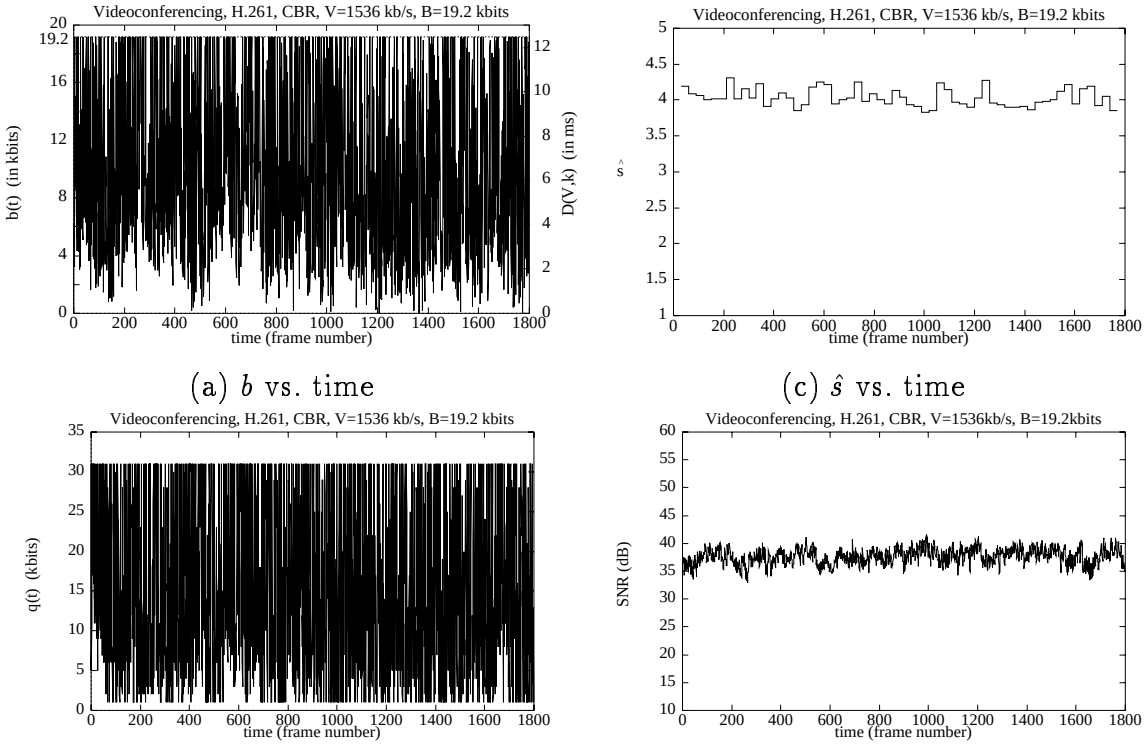
Figure 11: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=384 kb/s, B=384 kbits.



(a) b vs. time (c) $\hat{s}$ vs. time

(b) q vs. time (d) SNR vs. time

Figure 12: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=384 kb/s, B=1920 kbits.



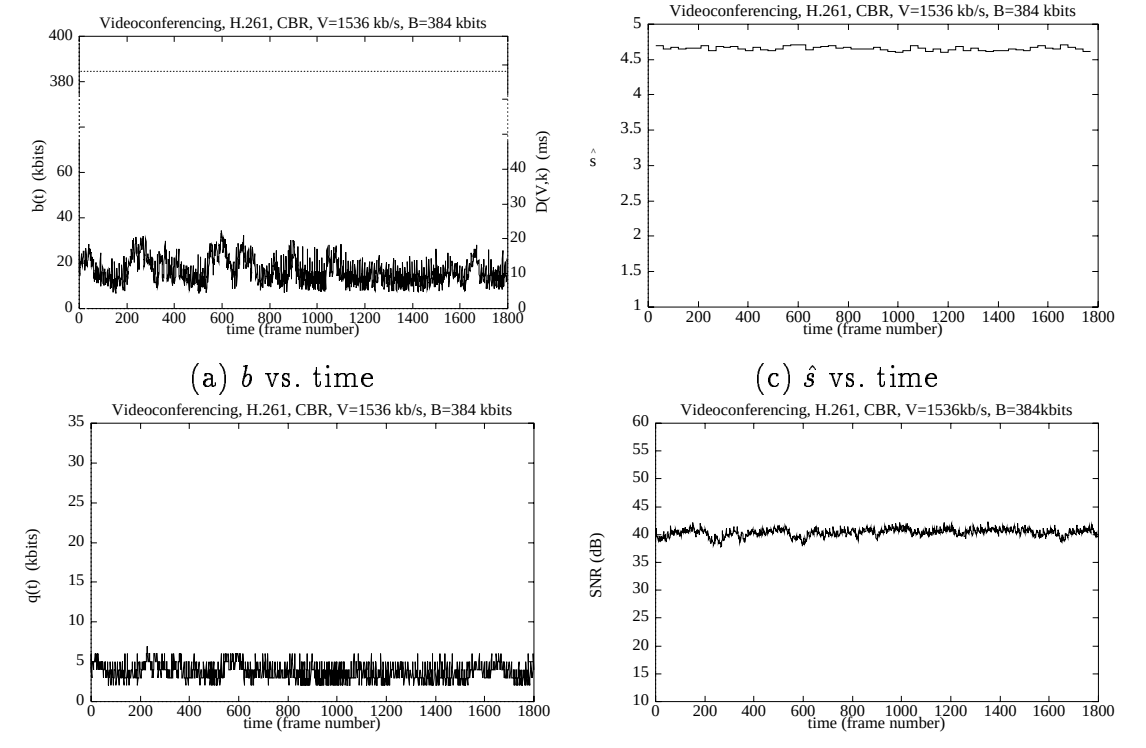
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

(d) SNR vs. time

Figure 13: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=1536 kb/s, B=19.2 kbits.



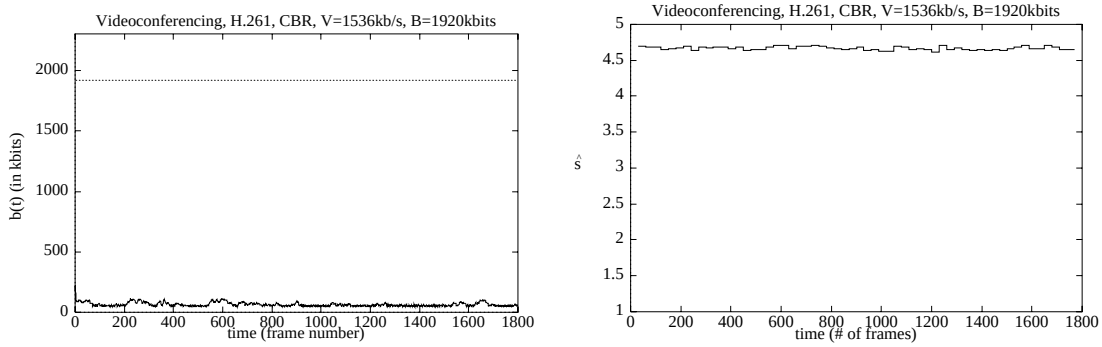
(a) b vs. time

(c) $\hat{s}$ vs. time

(b) q vs. time

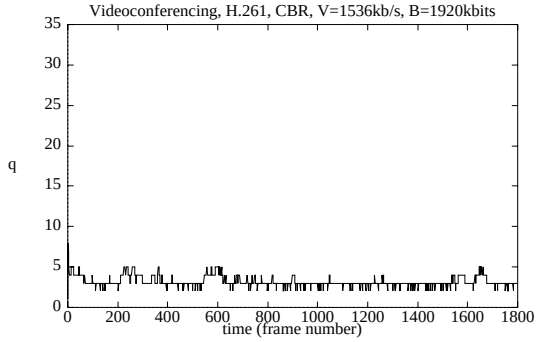
(d) SNR vs. time

Figure 14: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=1536 kb/s, B=384 kbits.

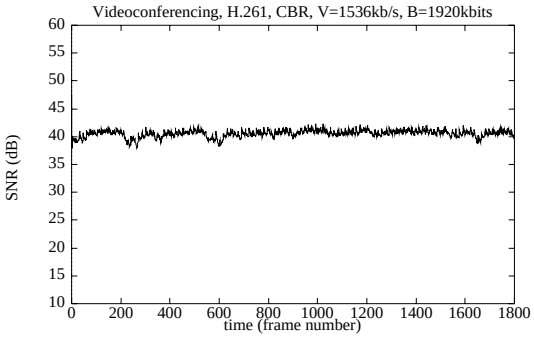


(a) b vs. time

(c) $\hat{s}$ vs. time

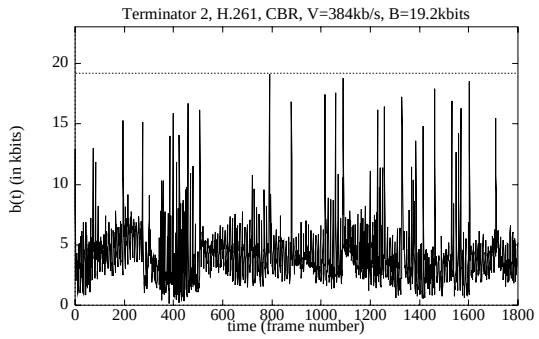


(b) q vs. time

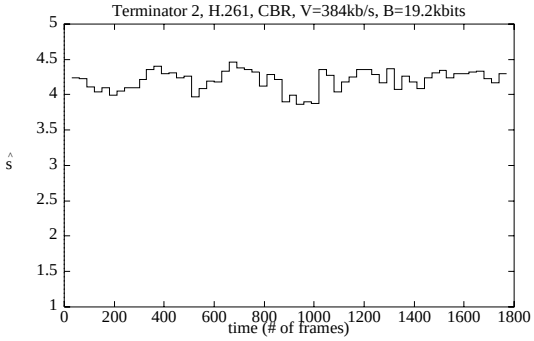


(d) SNR vs. time

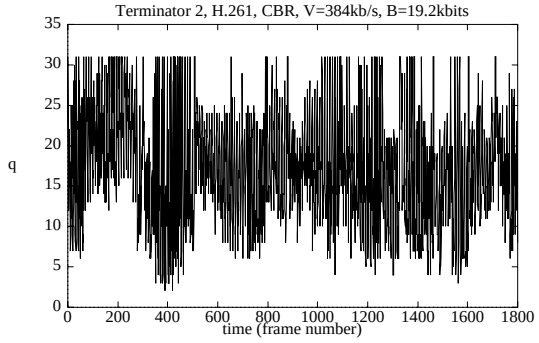
Figure 15: b, q, $\hat{s}$, and SNR vs. time for the Videoconferencing sequence, H.261, CBR, V=1536 kb/s, B=1920 kbits.



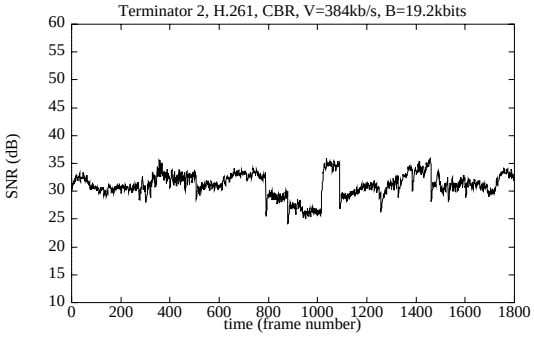
(a) b vs. time



(c) $\hat{s}$ vs. time

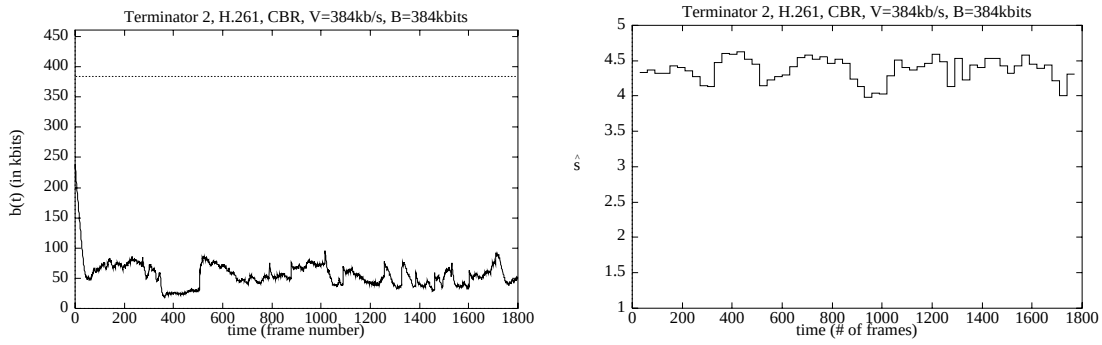


(b) q vs. time



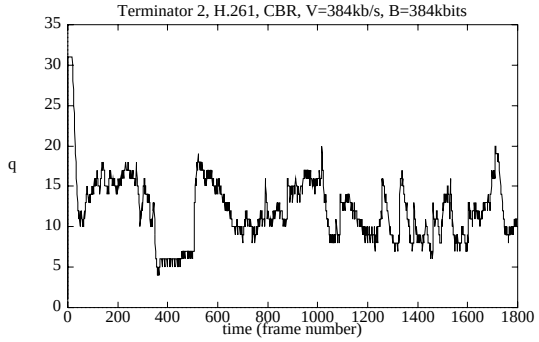
(d) SNR vs. time

Figure 16: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=384 kb/s, B=19.2 kbits.

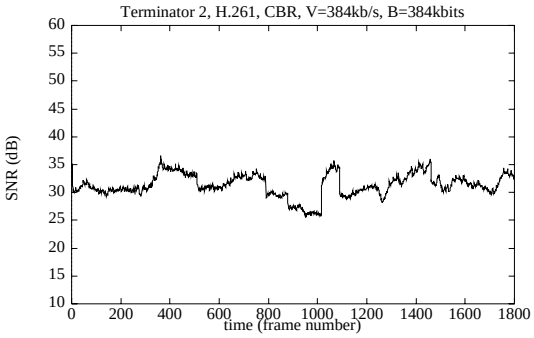


(a) b vs. time

(c) $\hat{s}$ vs. time

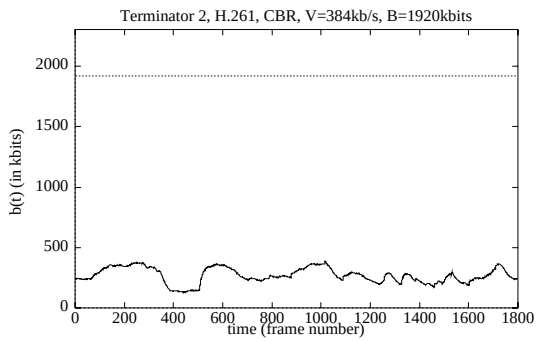


(b) q vs. time

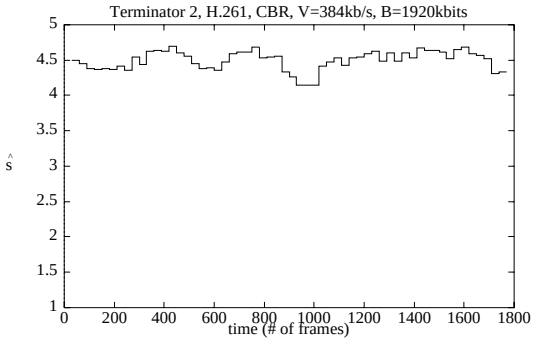


(d) SNR vs. time

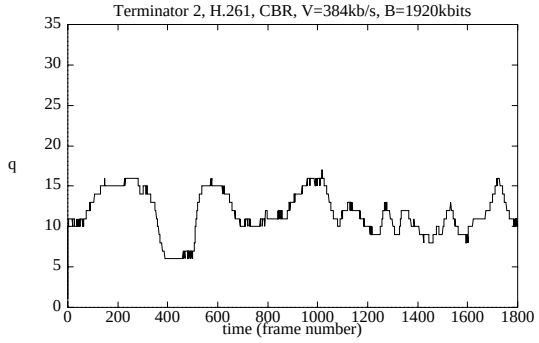
Figure 17: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=384 kb/s, B=384 kbits.



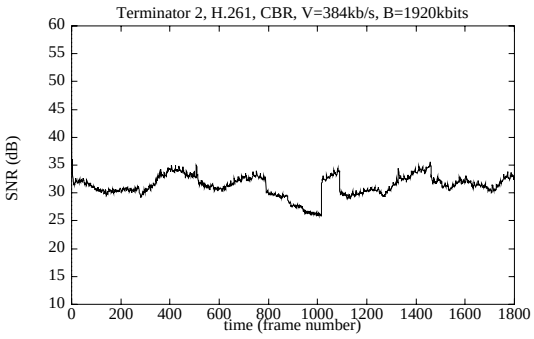
(a) b vs. time



(c) $\hat{s}$ vs. time

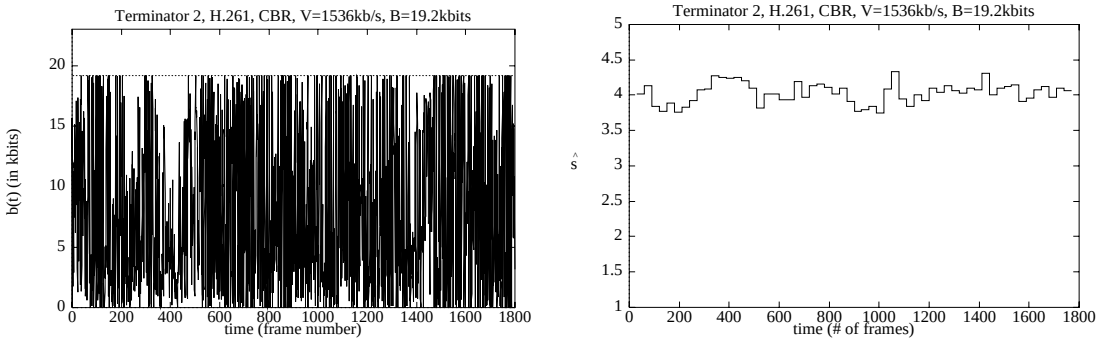


(b) q vs. time



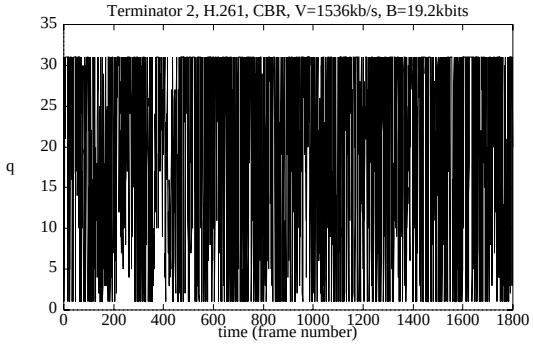
(d) SNR vs. time

Figure 18: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=384 kb/s, B=1920 kbits.

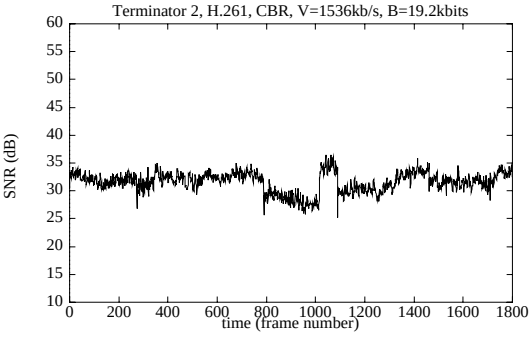


(a) b vs. time

(c) $\hat{s}$ vs. time

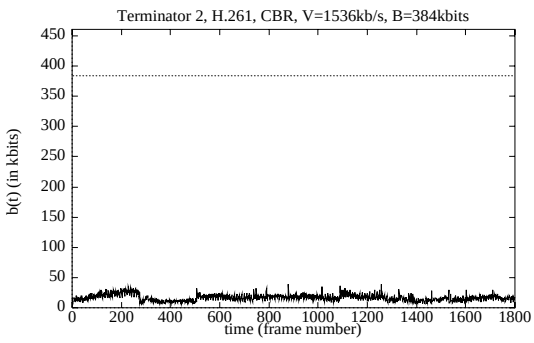


(b) q vs. time

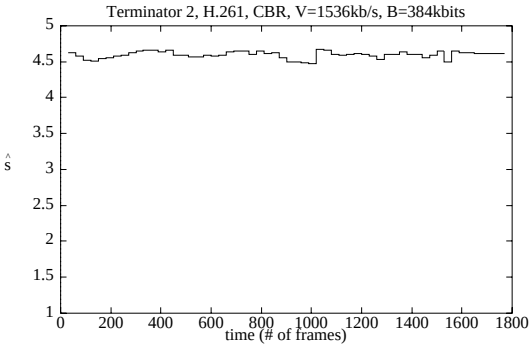


(d) SNR vs. time

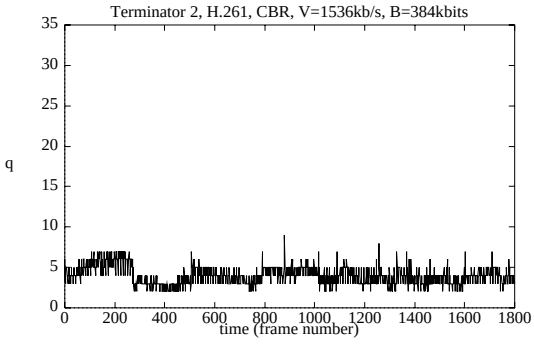
Figure 19: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=1536 kb/s, B=19.2 kbits.



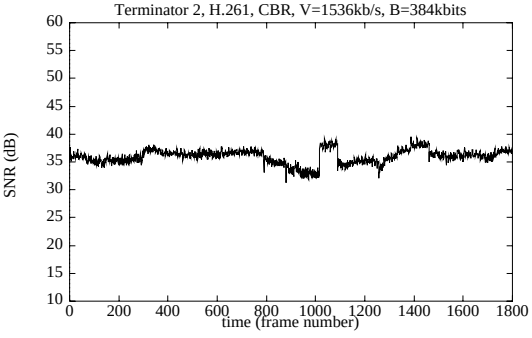
(a) b vs. time



(c) $\hat{s}$ vs. time

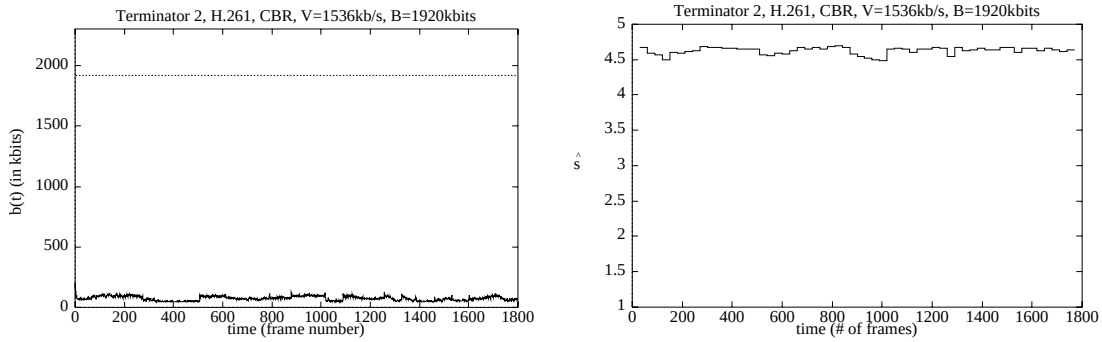


(b) q vs. time



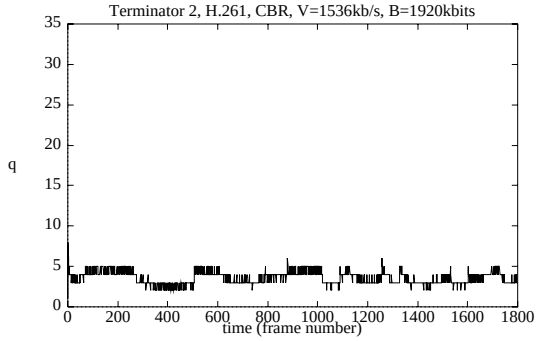
(d) SNR vs. time

Figure 20: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=1536 kb/s, B=384 kbits.

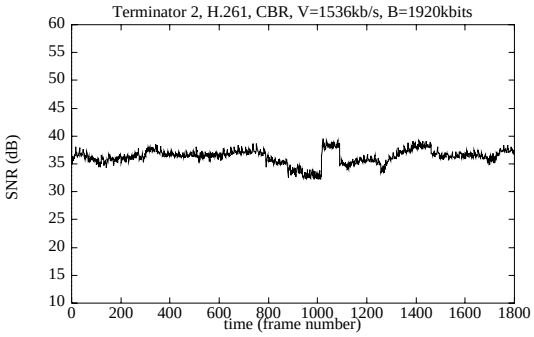


(a) b vs. time

(c) $\hat{s}$ vs. time

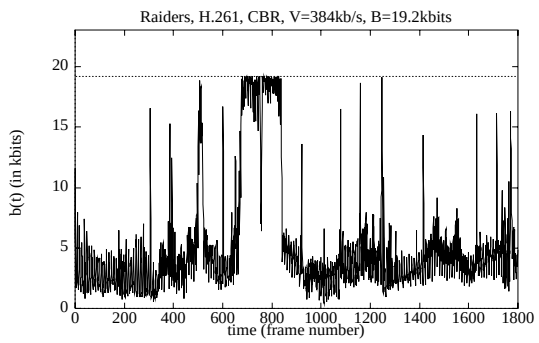


(b) q vs. time

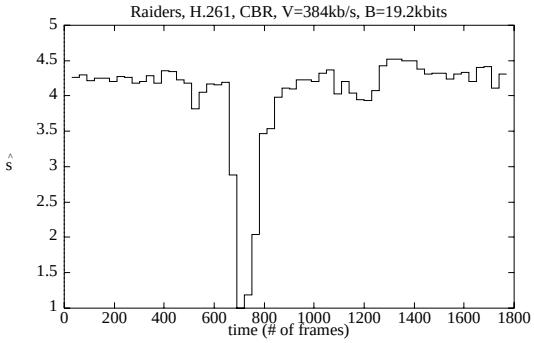


(d) SNR vs. time

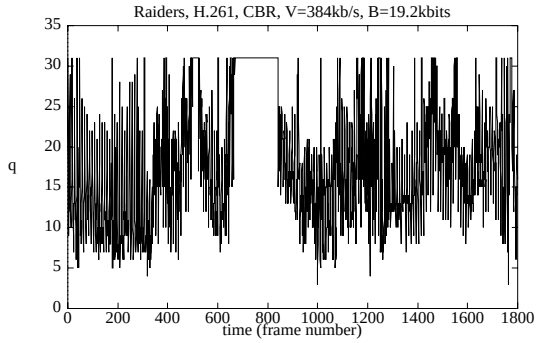
Figure 21: b, q, $\hat{s}$, and SNR vs. time for the Terminator-2 sequence, H.261, CBR, V=1536 kb/s, B=1920 kbits.



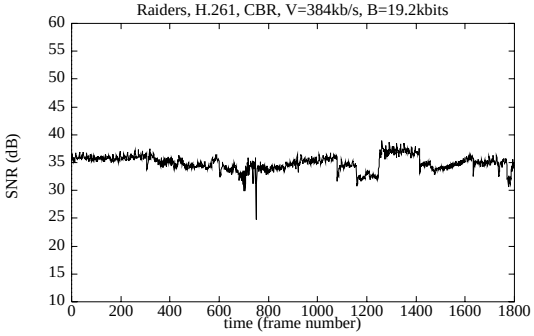
(a) b vs. time



(c) $\hat{s}$ vs. time

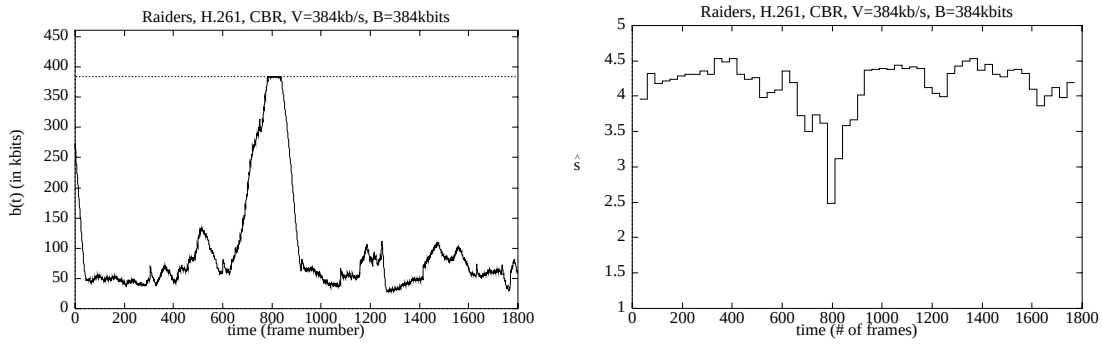


(b) q vs. time



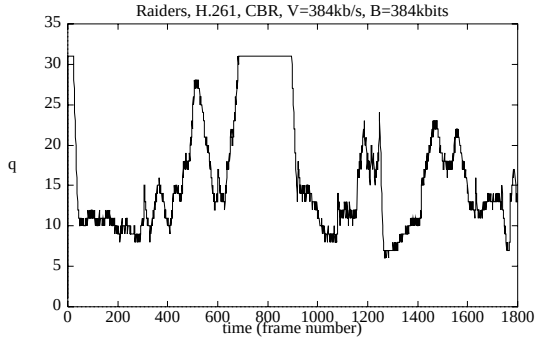
(d) SNR vs. time

Figure 22: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=384 kb/s, B=19.2 kbits.

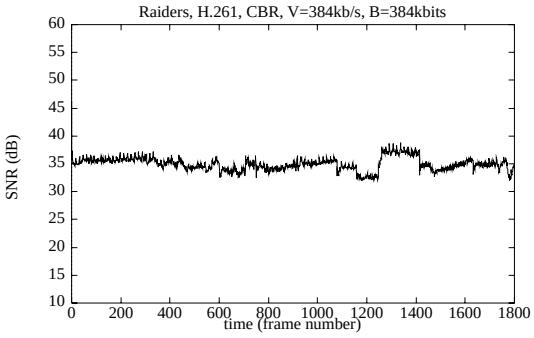


(a) b vs. time

(c) $\hat{s}$ vs. time

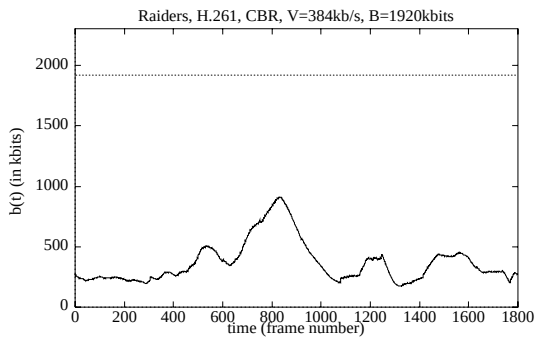


(b) q vs. time

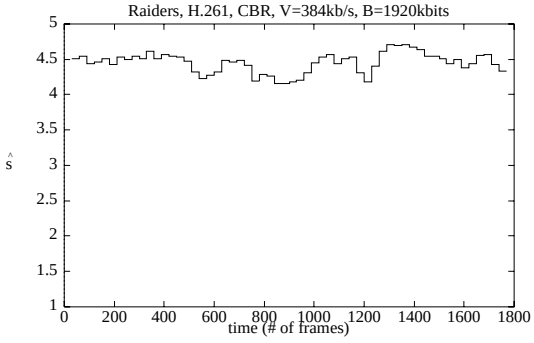


(d) SNR vs. time

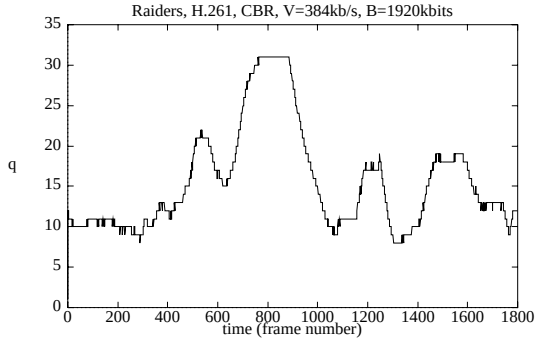
Figure 23: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=384 kb/s, B=384 kbits.



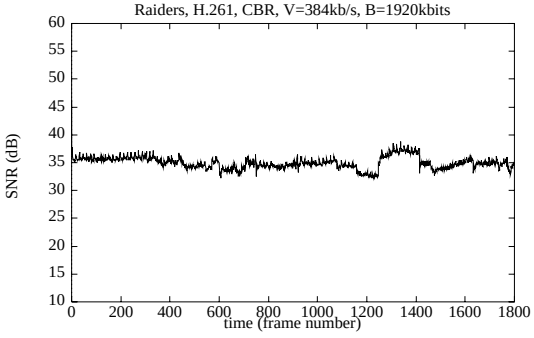
(a) b vs. time



(c) $\hat{s}$ vs. time

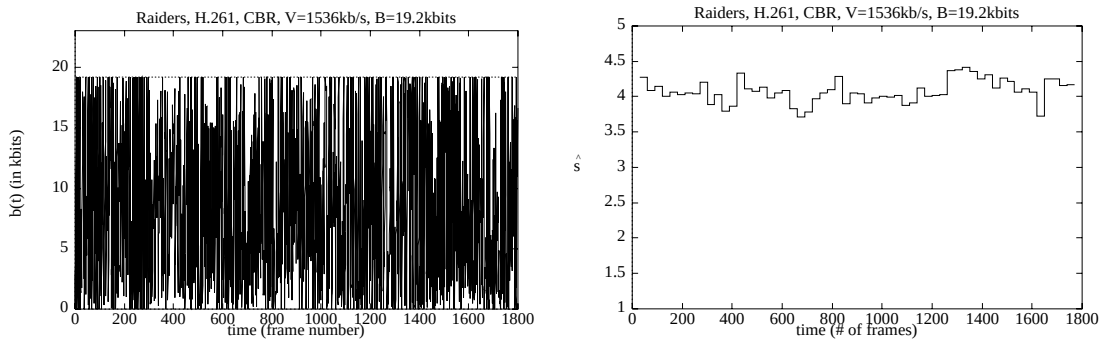


(b) q vs. time



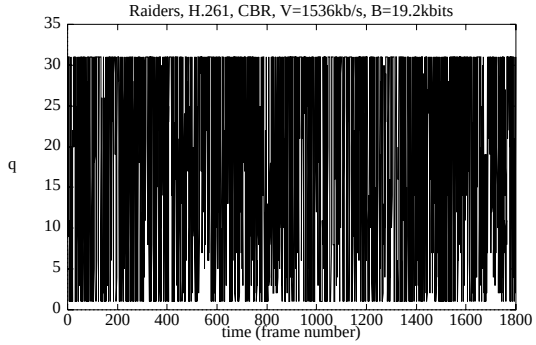
(d) SNR vs. time

Figure 24: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=384 kb/s, B=1920 kbits.

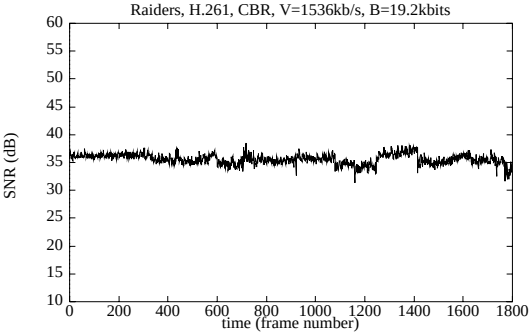


(a) b vs. time

(c) $\hat{s}$ vs. time

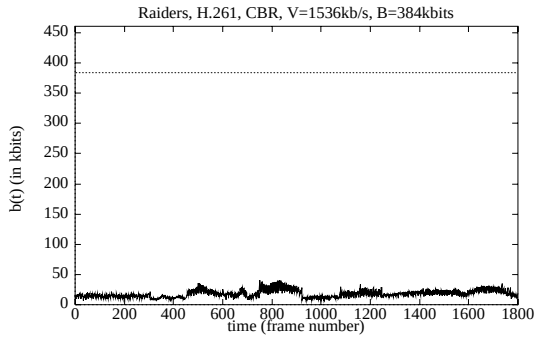


(b) q vs. time

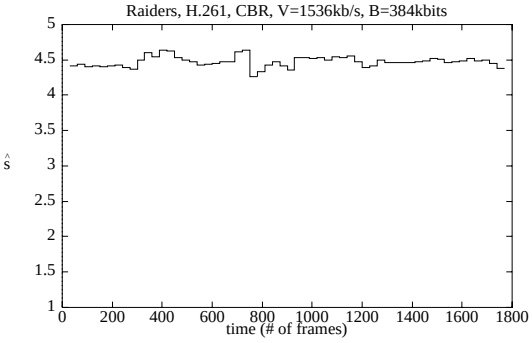


(d) SNR vs. time

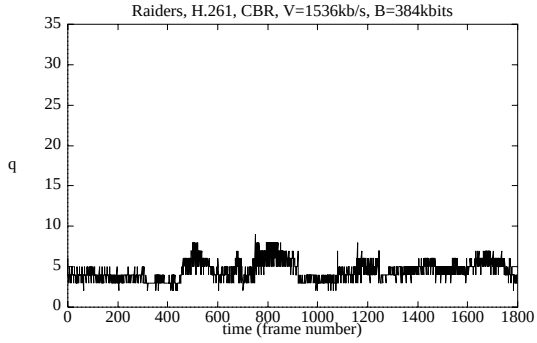
Figure 25: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=1536 kb/s, B=19.2 kbits.



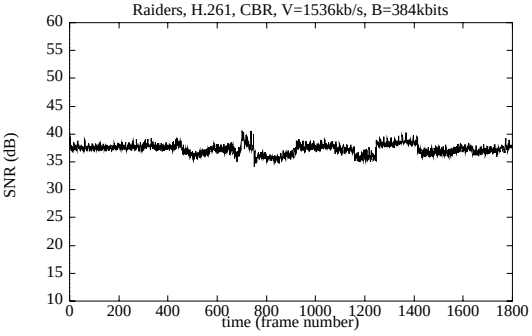
(a) b vs. time



(c) $\hat{s}$ vs. time

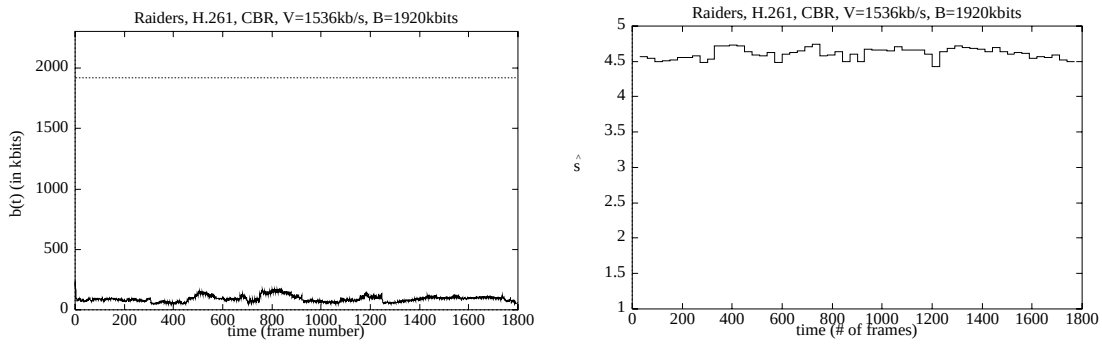


(b) q vs. time



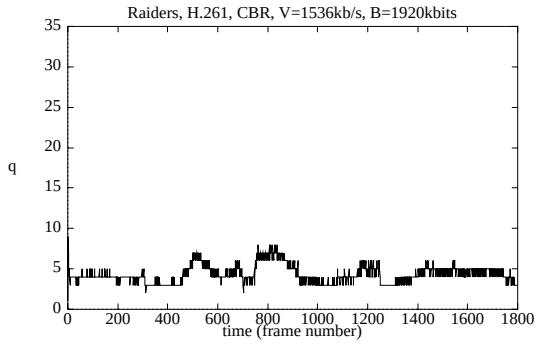
(d) SNR vs. time

Figure 26: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=1536 kb/s, B=384 kbits.

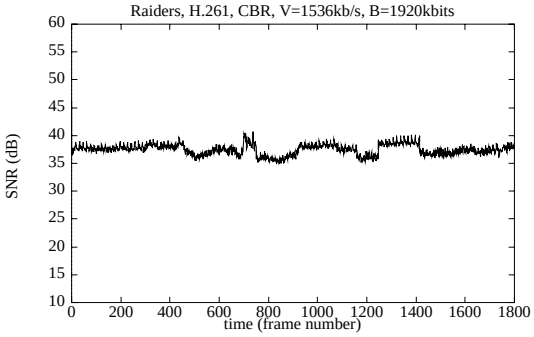


(a) b vs. time

(c) $\hat{s}$ vs. time

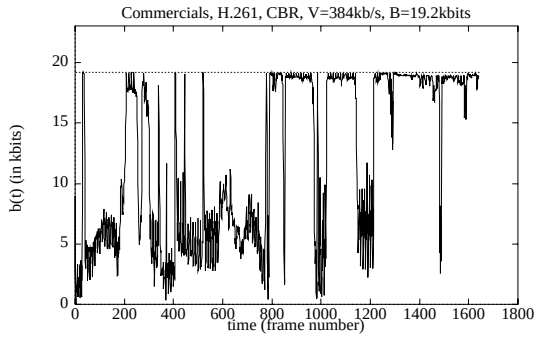


(b) q vs. time

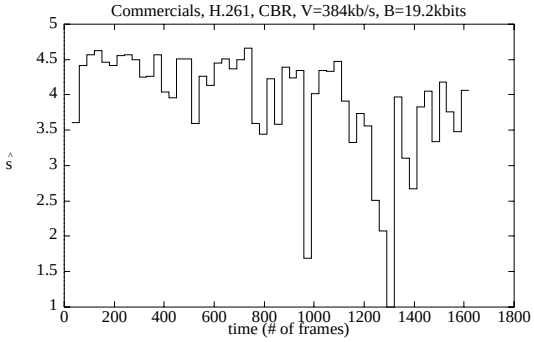


(d) SNR vs. time

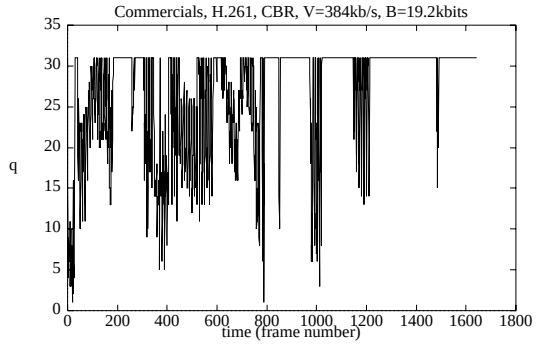
Figure 27: b, q, $\hat{s}$, and SNR vs. time for the Raiders sequence, H.261, CBR, V=1536 kb/s, B=1920 kbits.



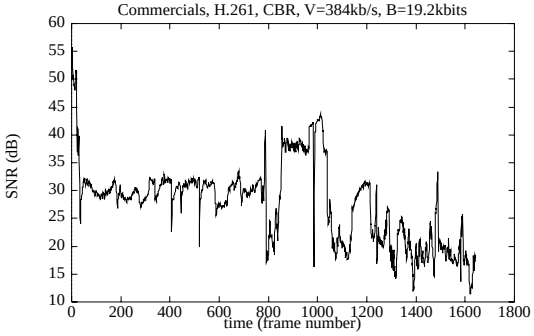
(a) b vs. time



(c) $\hat{s}$ vs. time

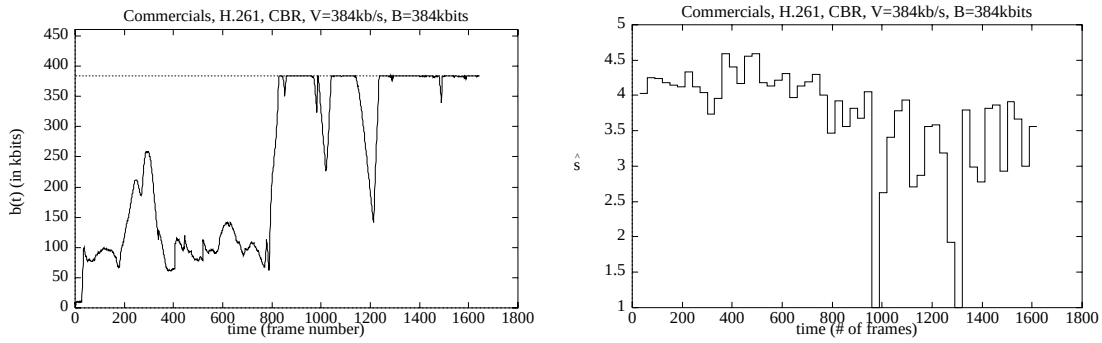


(b) q vs. time



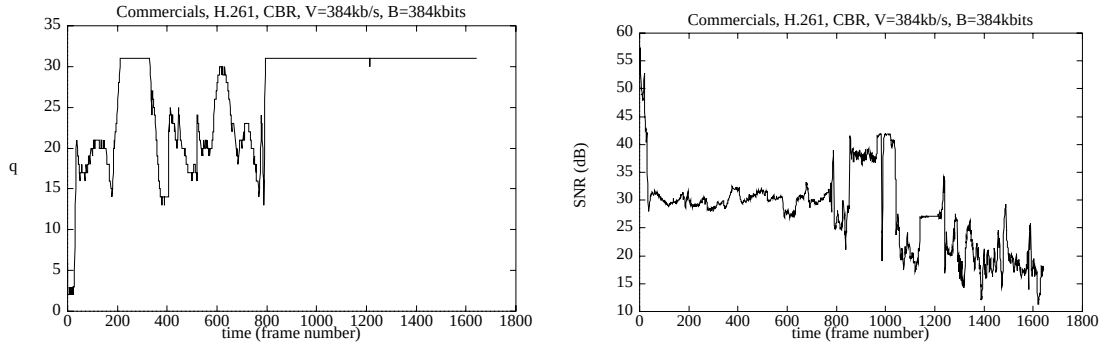
(d) SNR vs. time

Figure 28: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=384 kb/s, B=19.2 kbits.



(a) b vs. time

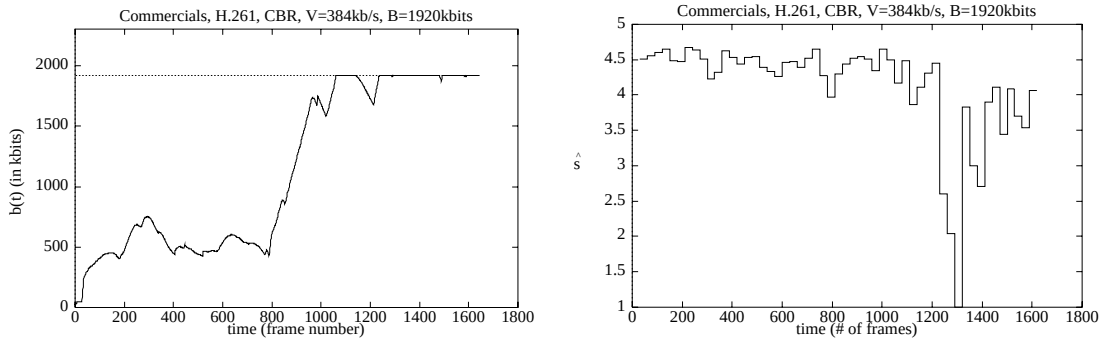
(c) $\hat{s}$ vs. time



(b) q vs. time

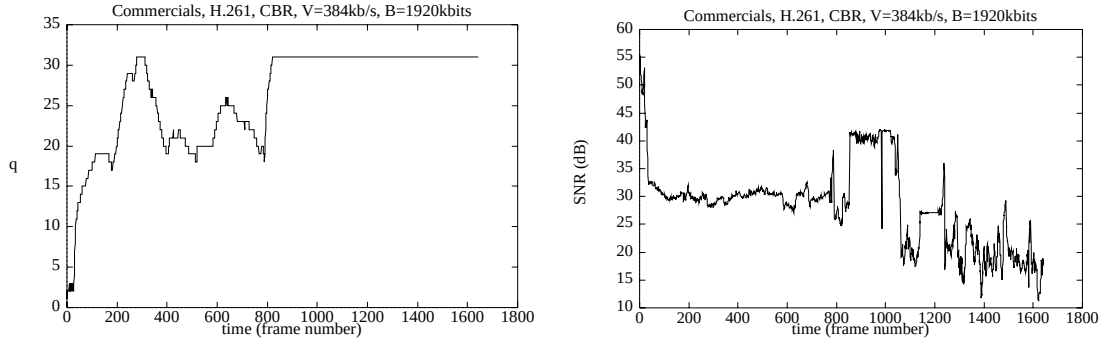
(d) SNR vs. time

Figure 29: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=384 kb/s, B=384 kbits.



(a) b vs. time

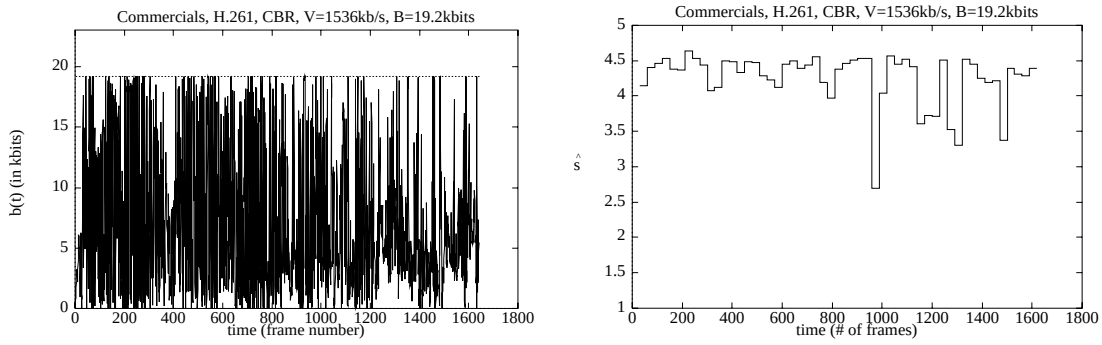
(c) $\hat{s}$ vs. time



(b) q vs. time

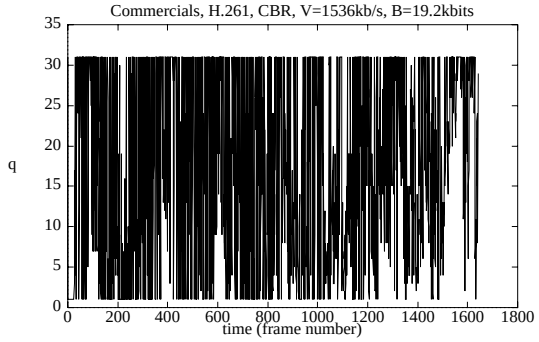
(d) SNR vs. time

Figure 30: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=384 kb/s, B=1920 kbits.

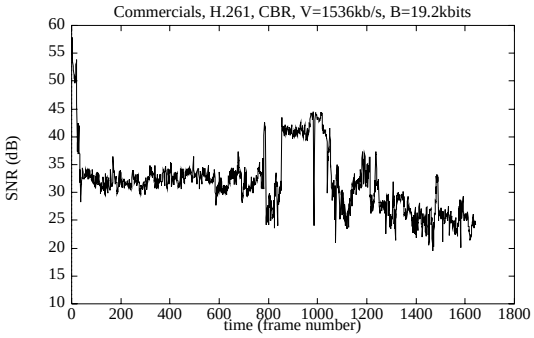


(a) b vs. time

(c) $\hat{s}$ vs. time

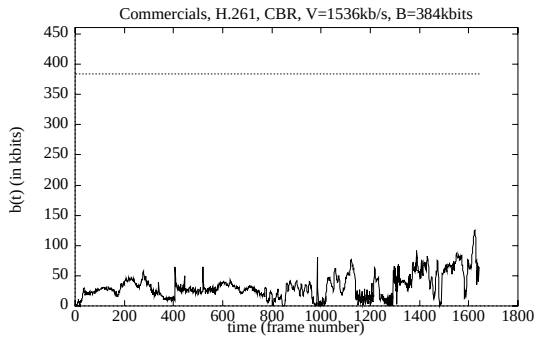


(b) q vs. time

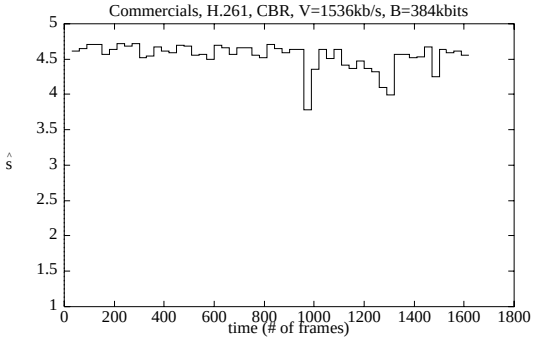


(d) SNR vs. time

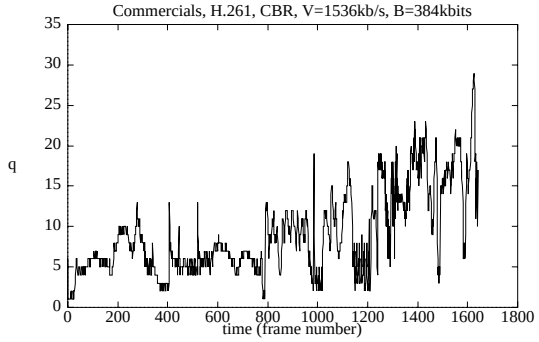
Figure 31: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=1536 kb/s, B=19.2 kbits.



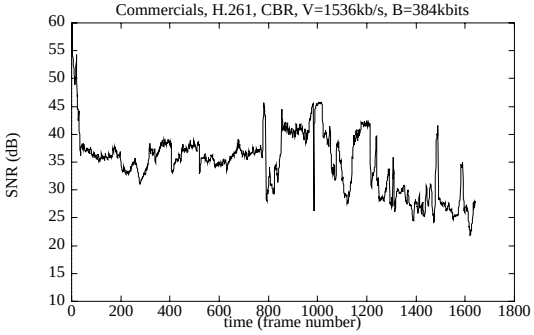
(a) b vs. time



(c) $\hat{s}$ vs. time

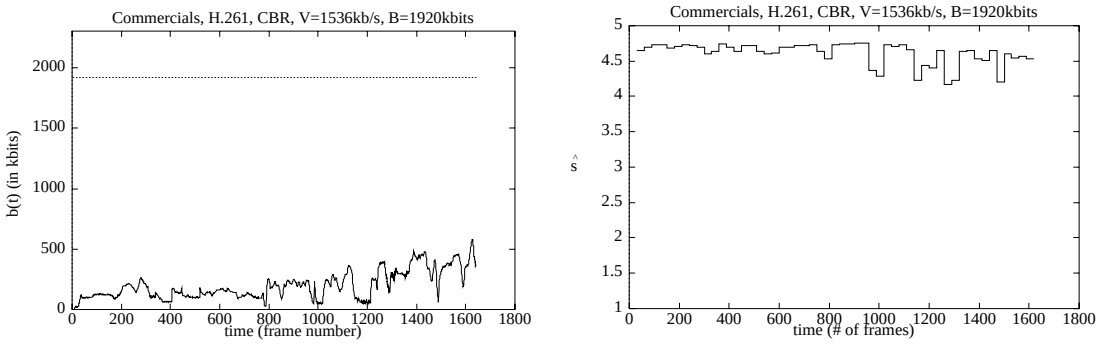


(b) q vs. time



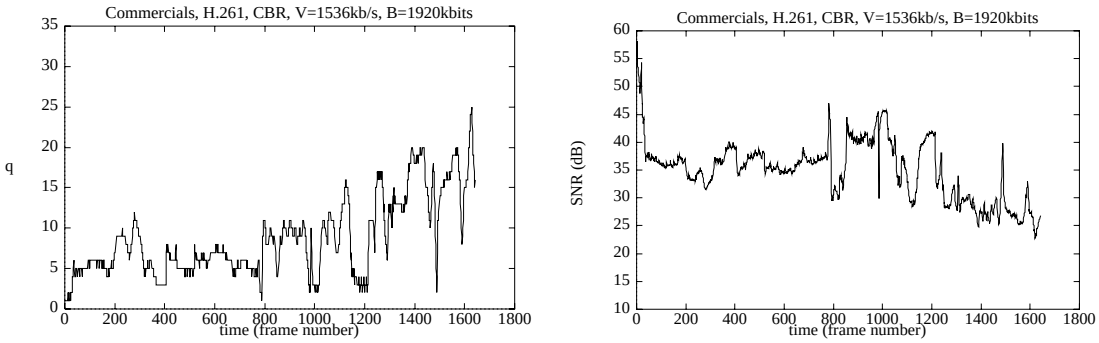
(d) SNR vs. time

Figure 32: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=1536 kb/s, B=384 kbits.



(a) b vs. time

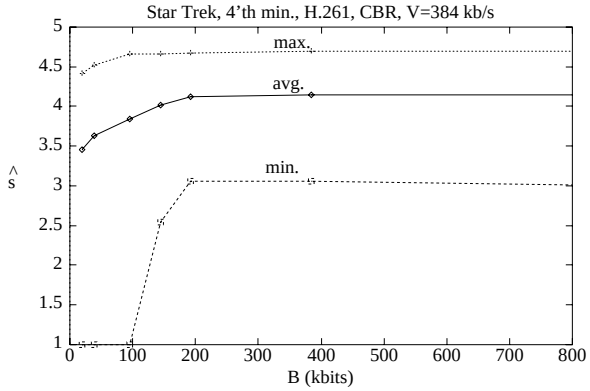
(c) $\hat{s}$ vs. time



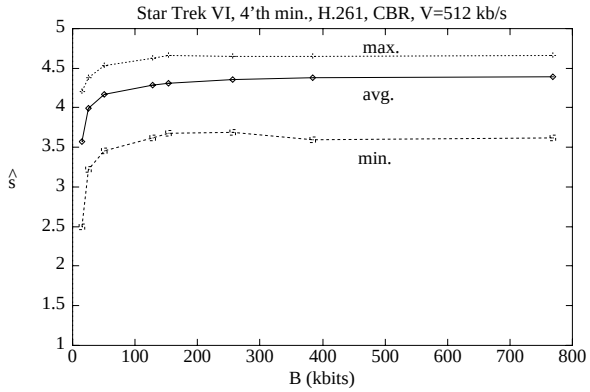
(b) q vs. time

(d) SNR vs. time

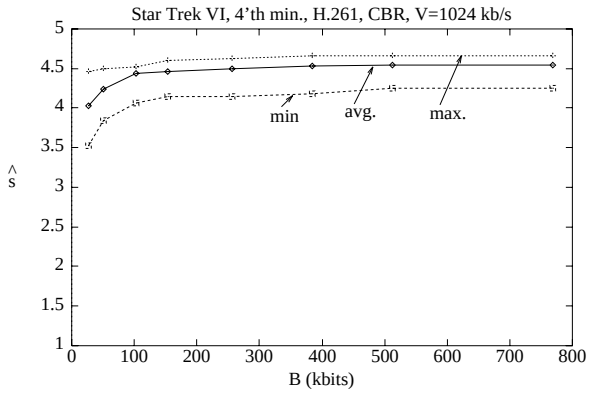
Figure 33: b, q, $\hat{s}$, and SNR vs. time for the Commercials sequence, H.261, CBR, V=1536 kb/s, B=1920 kbits.



(a) V=384 kb/s.

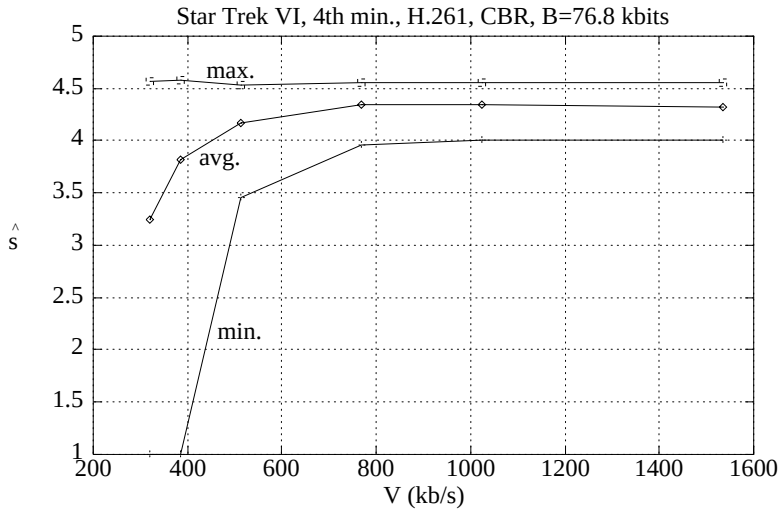


(b) V=512 kb/s.

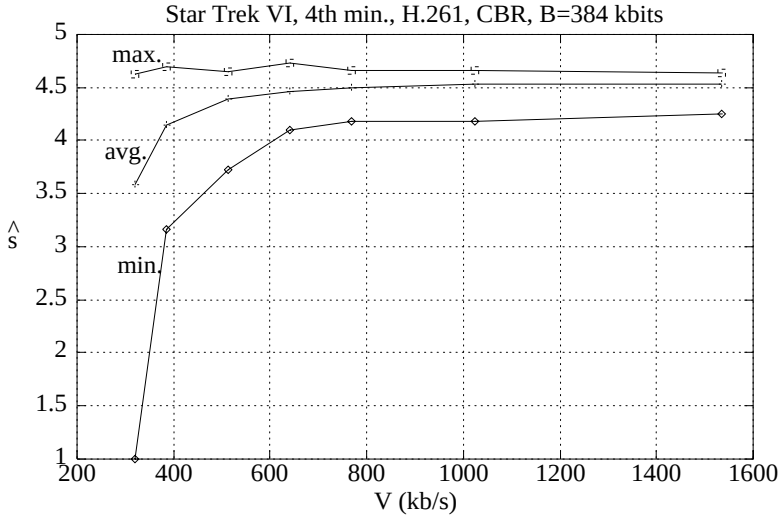


(c) V=1024 kb/s.

Figure 34: Maximum, average, and minimum $\hat{s}$ as a function of B for given V for the Star Trek sequence.



(a) B=76.8 kbits



(b) B=384 kbits

Figure 35: Maximum, average, and minimum $\hat{s}$ versus V for the Star Trek sequence, H.261, CBR.

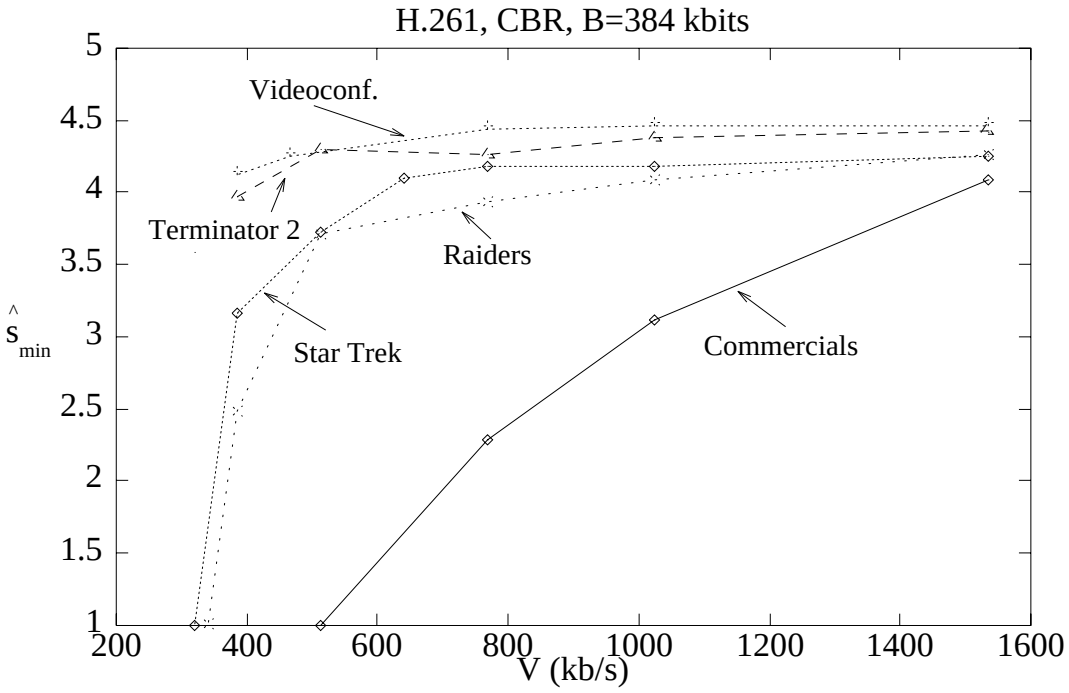


Figure 36: $\hat{s}_{min}$ vs. V for various sequences, H.261, CBR.

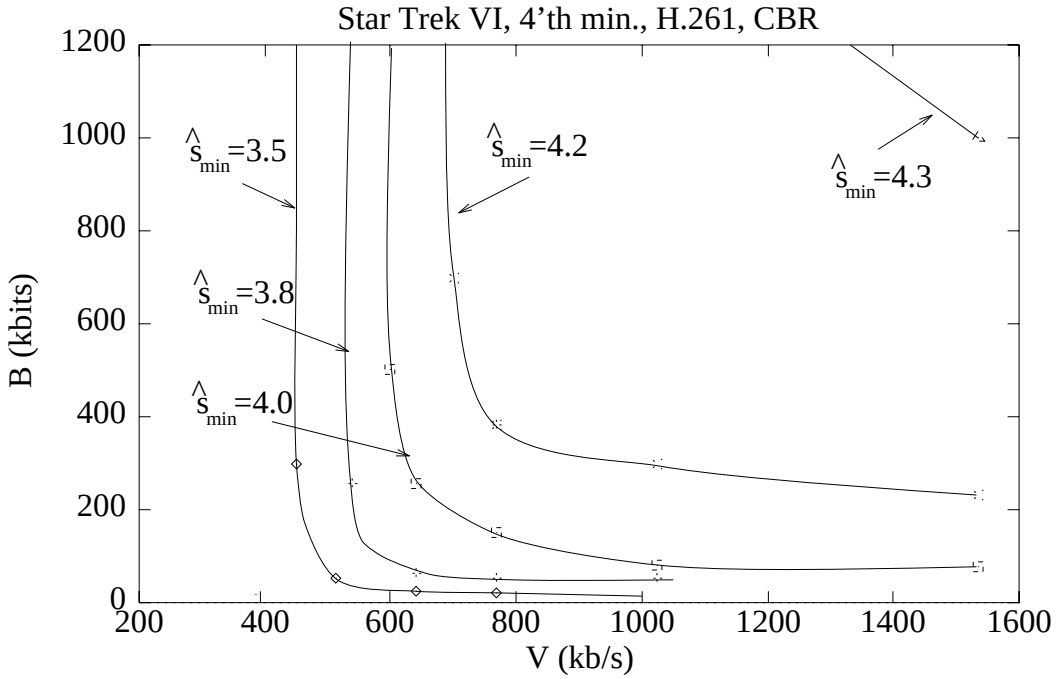


Figure 37: Equal $\hat{s}_{min}$ contours for the Star Trek sequence, H.261, CBR.

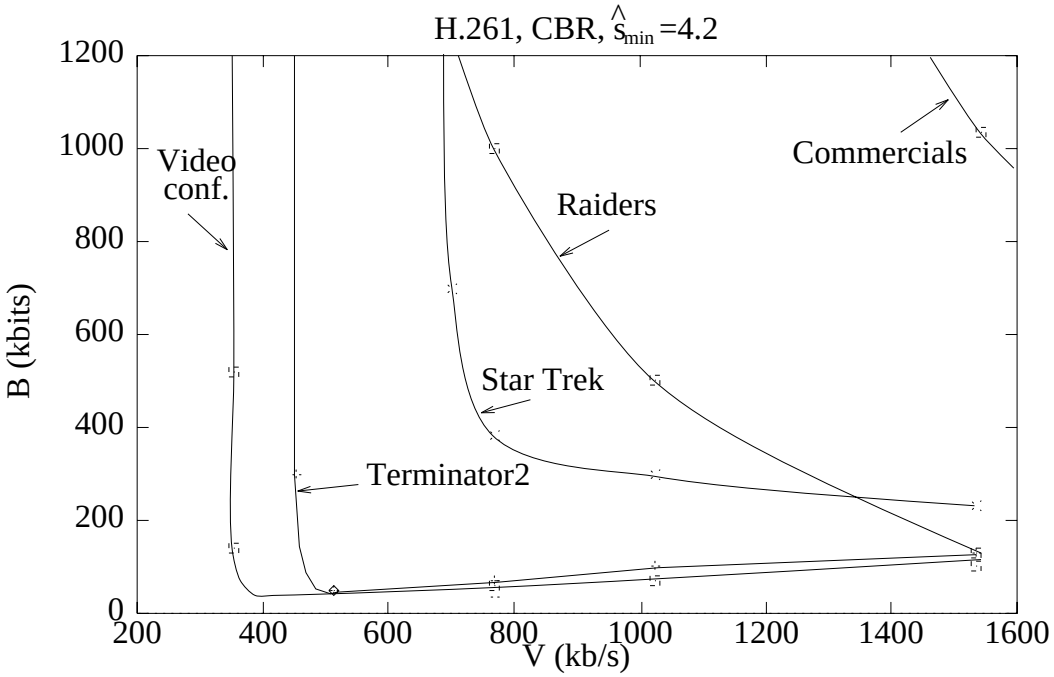
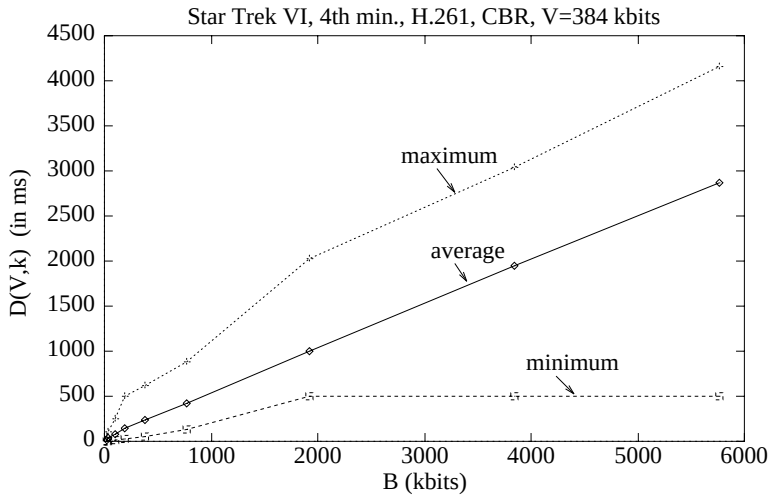
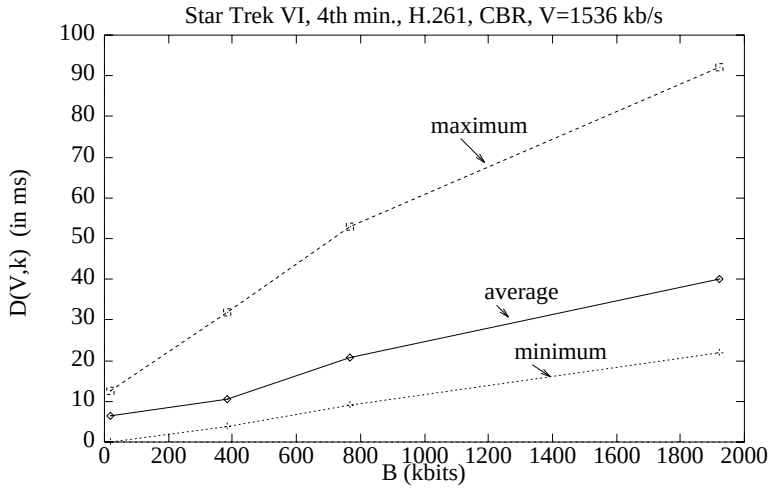


Figure 38: $\hat{s}_{min}$=4.2 contours for various sequences, H.261, CBR.



(a) V=384 kb/s



(b) V=1536 kb/s

Figure 39: Maximum, average, and minimum D_r versus B for the Star Trek sequence, H.261, CBR.

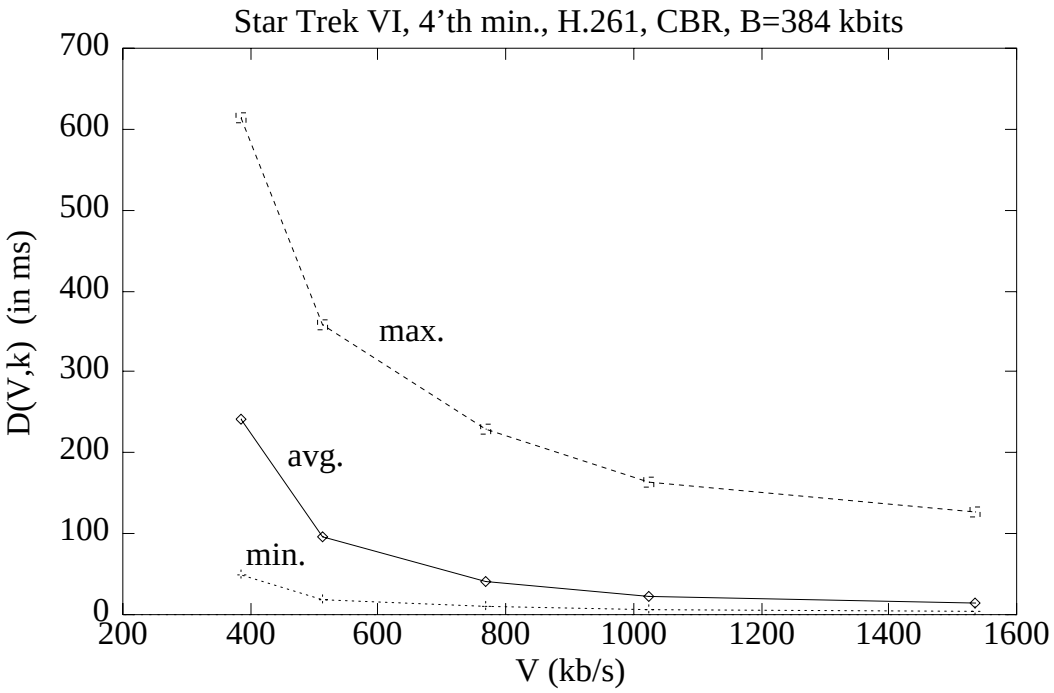


Figure 40: Maximum, average, and minimum $D(V,k)$ versus V for the Star Trek sequence, H.261, CBR, B=384 kbits.

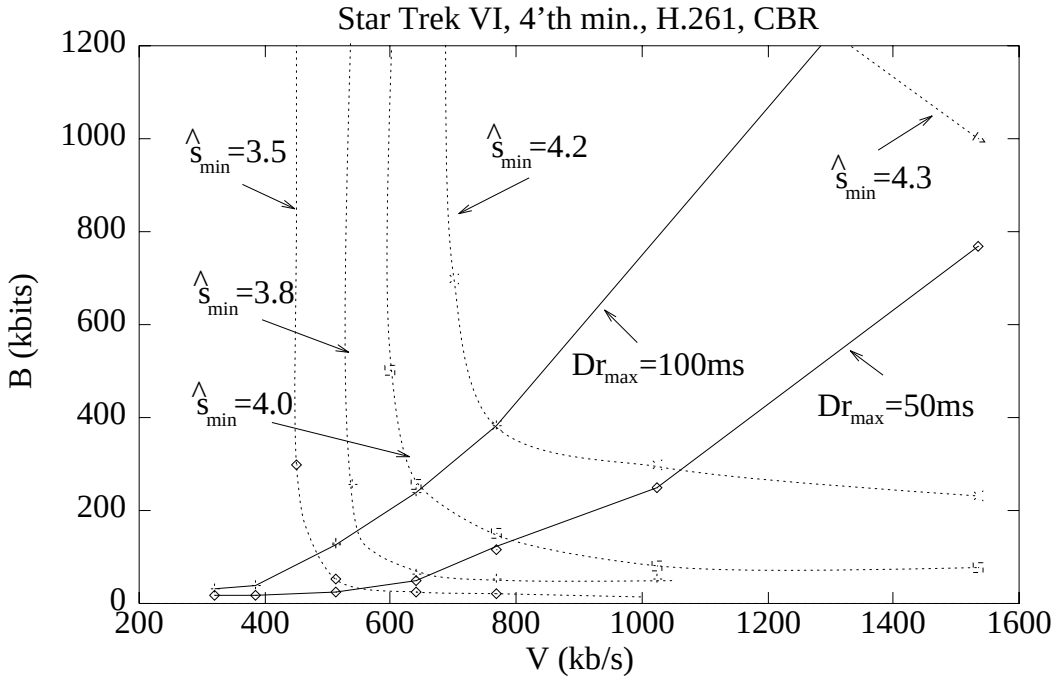


Figure 41: Equal $\max_k\{D(V,k)\}$ and $\hat{s}_{min}$ contours for the Star Trek sequence, H.261, CBR.

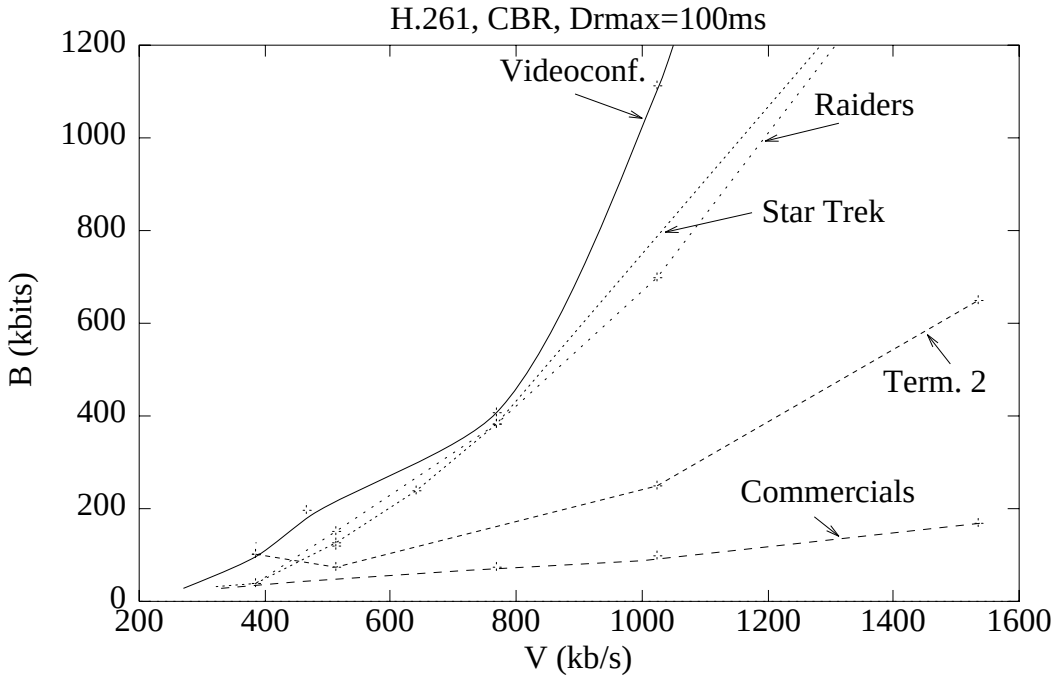


Figure 42: $\max_k\{D(V,k)\}$=100 ms contours for various sequences, H.261, CBR.

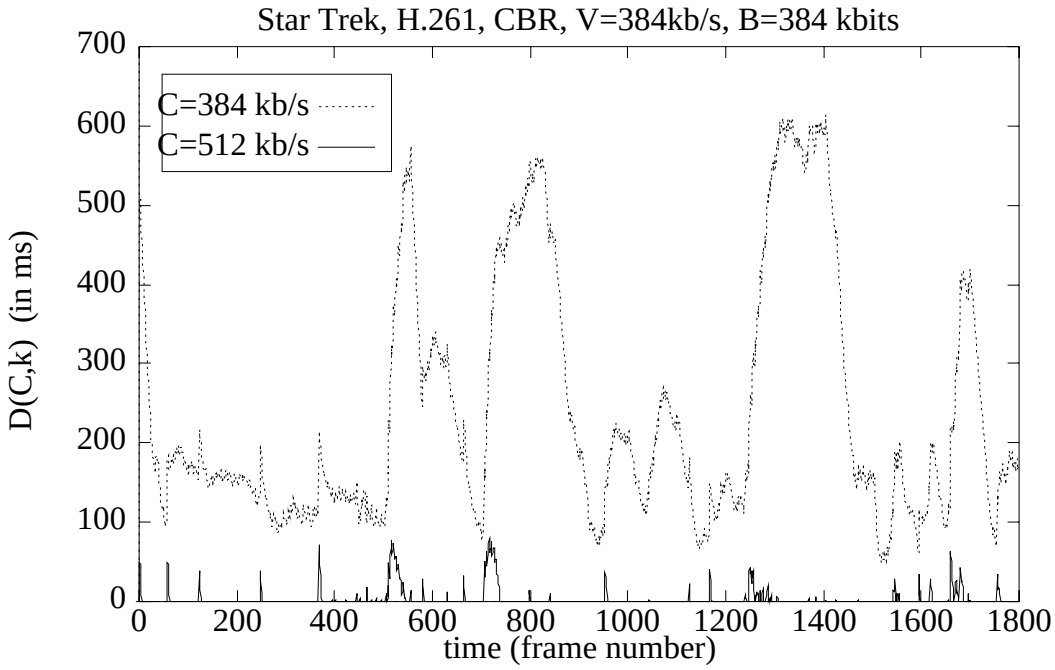


Figure 43: $D(C,k)$ versus time for the Star Trek sequence, H.261, CBR, V=384 kb/s, $B = 384$ *kbits*, C=$\{$384,512$\}$ kb/s.

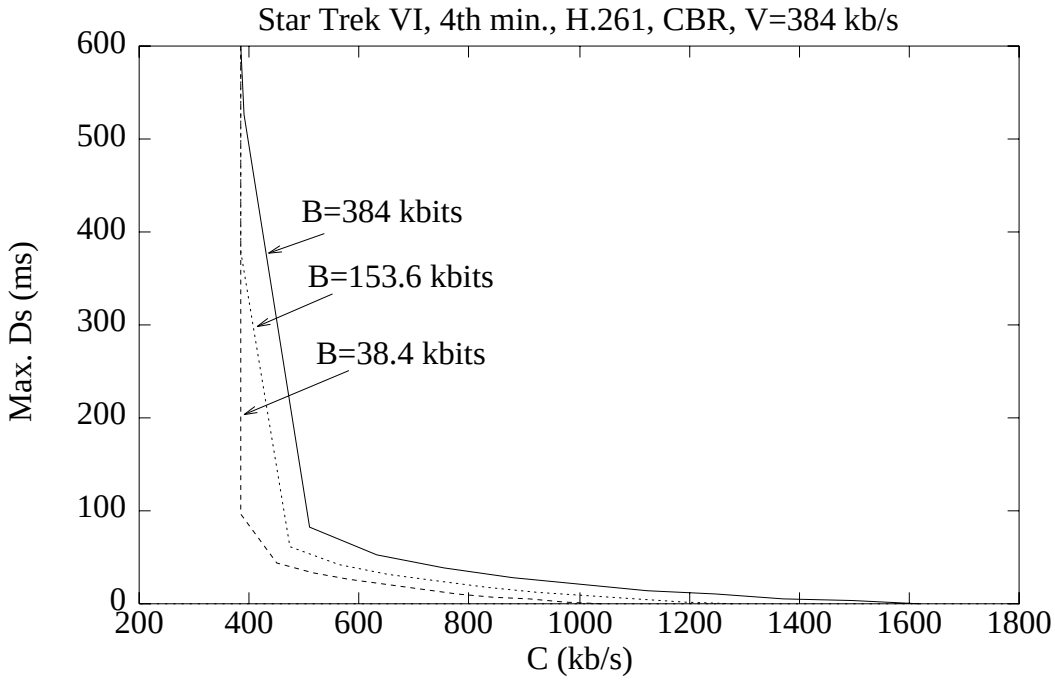


Figure 44: $\max_k\{D(C,k)\}$ versus C for the Star Trek sequence, H.261, CBR, V=384 kb/s, various values of B.

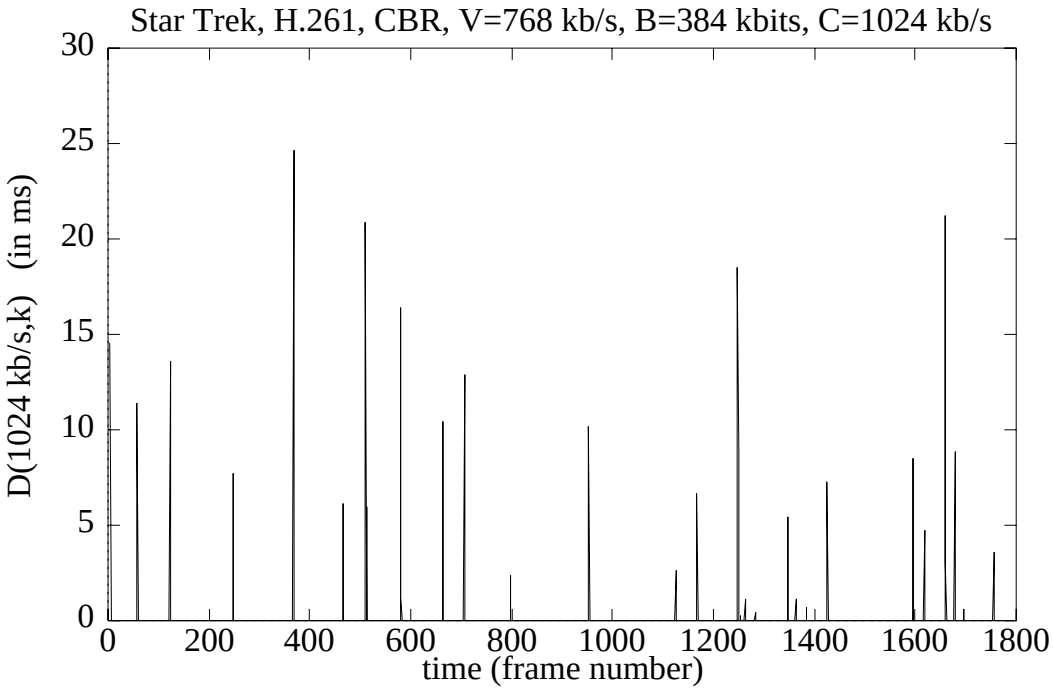
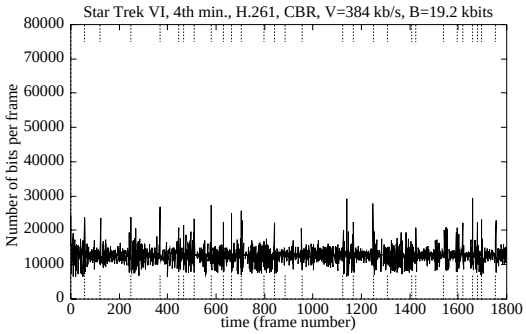
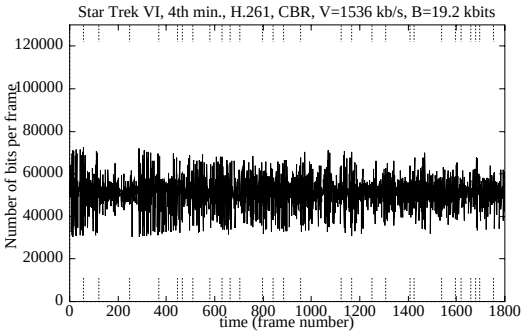


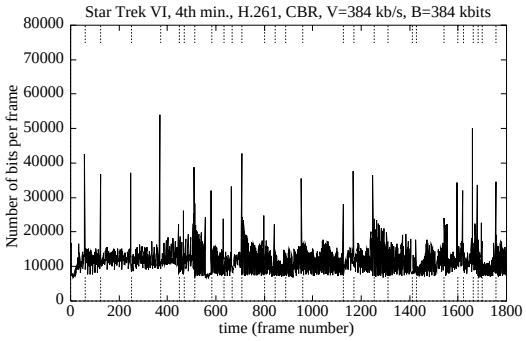
Figure 45: $D(C,k)$ versus time for the Star Trek sequence, H.261, CBR, V=768 kb/s, B=384 kbits, C=1024 kb/s.



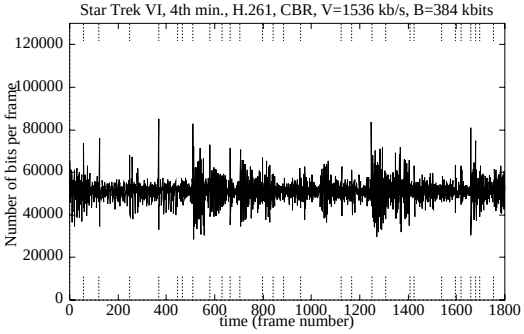
(a) V=384 kb/s, B=19.2 kbits.



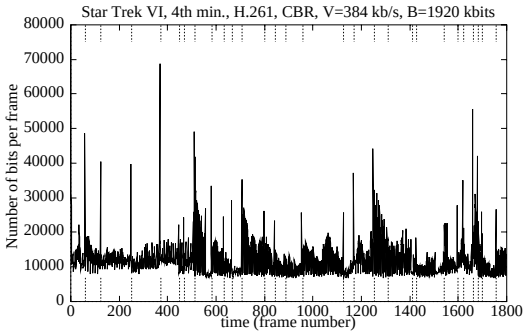
(d) V=1536 kb/s, B=19.2 kbits.



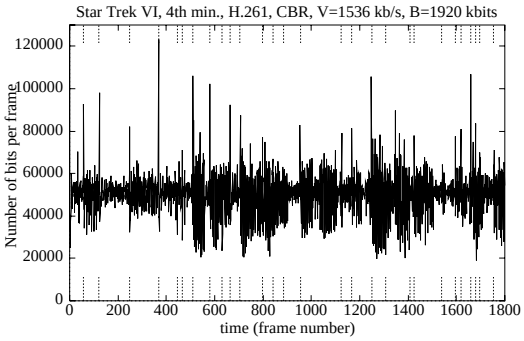
(b) V=384 kb/s, B=384 kbits.



(e) V=1536 kb/s, B=384 kbits.



(c) V=384 kb/s, B=1920 kbits.



(f) V=1536 kb/s, B=1920 kbits.

Figure 46: Number of bits per frame for the Star Trek sequence, H.261, CBR.

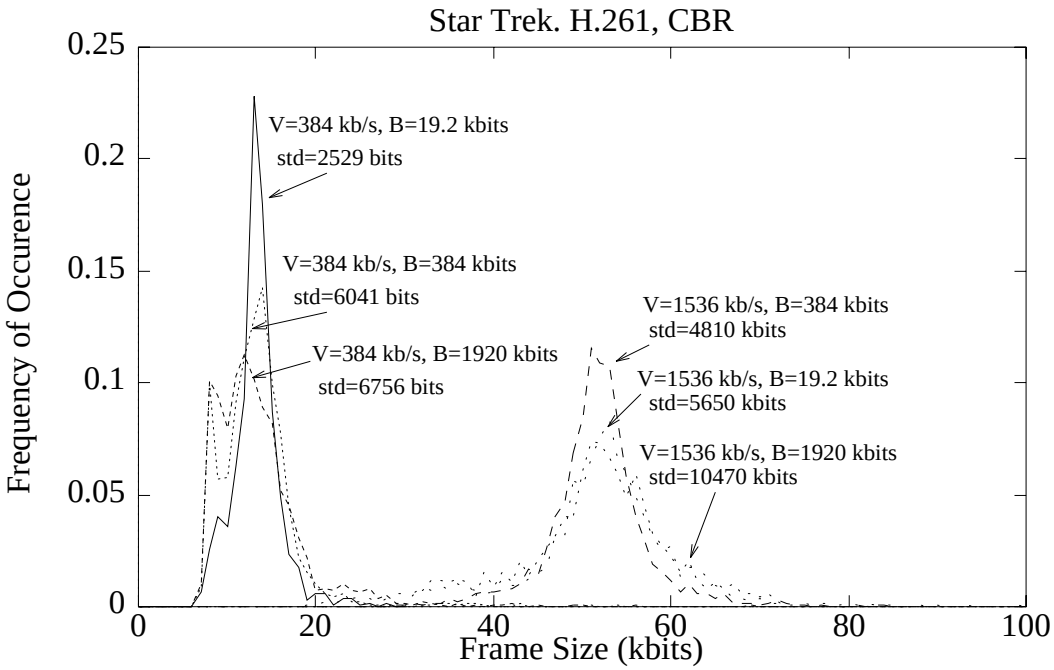
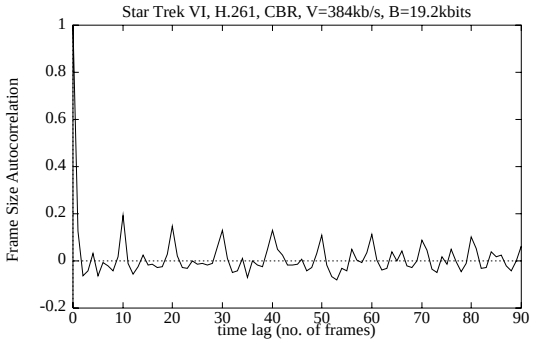
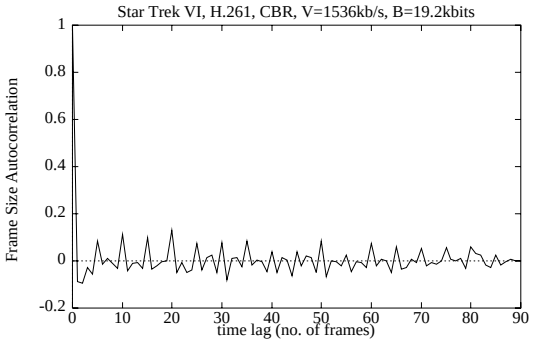


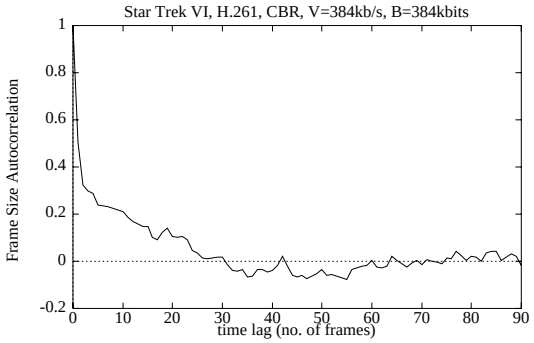
Figure 47: Frame size histogram for Star Trek, H.261, CBR.



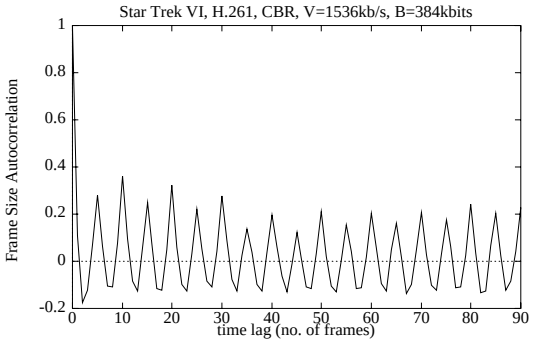
(a) V=384 kb/s, B=19.2 kbits



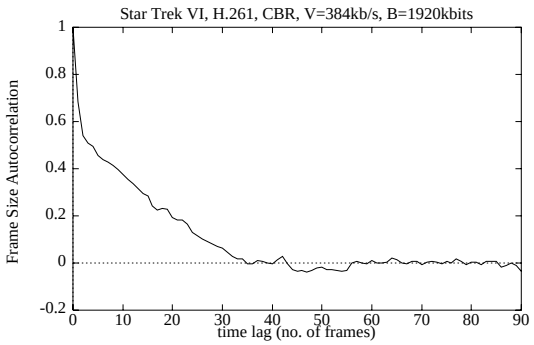
(d) V=1536 kb/s, B=19.2 kbits



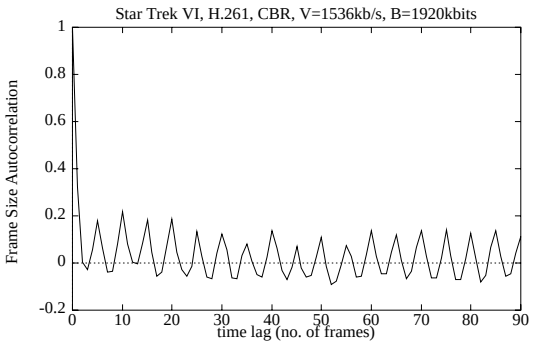
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits

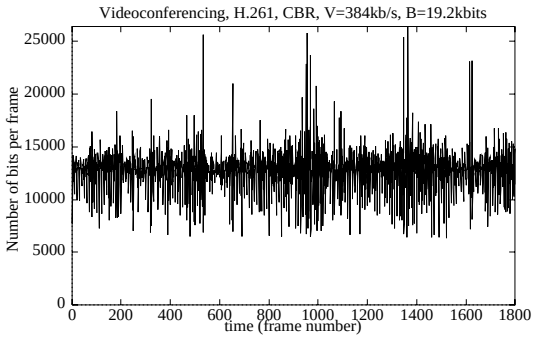


(c) V=384 kb/s, B=1920 kbits

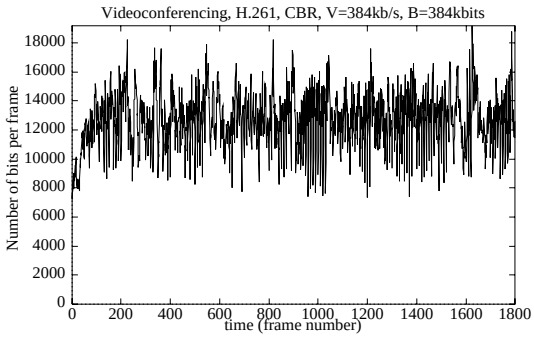


(f) V=1536 kb/s, B=1920 kbits

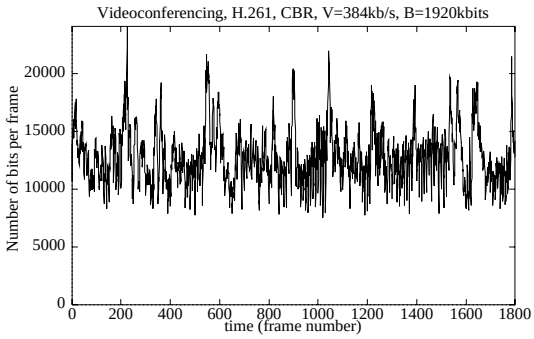
Figure 48: Frame size autocorrelation function for Star Trek, H.261, CBR



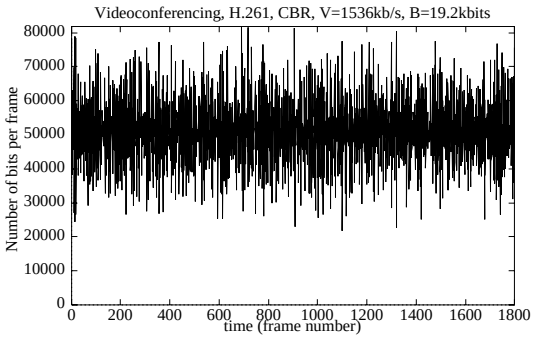
(a) V=384 kb/s, B=19.2 kbits



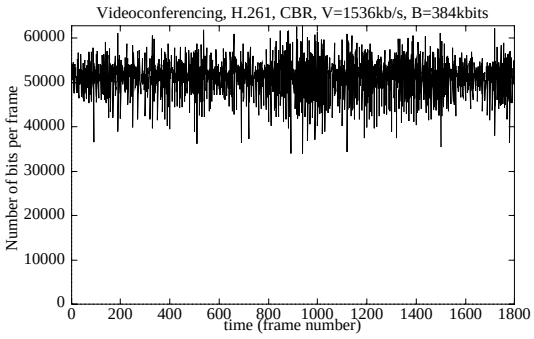
(b) V=384 kb/s, B=384 kbits



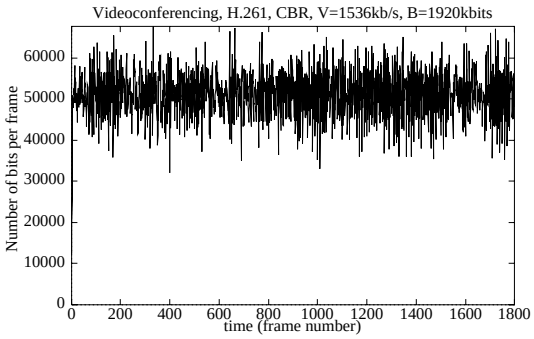
(c) V=384 kb/s, B=1920 kbits



(d) V=1536 kb/s, B=19.2 kbits



(e) V=1536 kb/s, B=384 kbits



(f) V=1536 kb/s, B=1920 kbits

Figure 49: Number of bits per frame for Videoconferencing, H.261, CBR

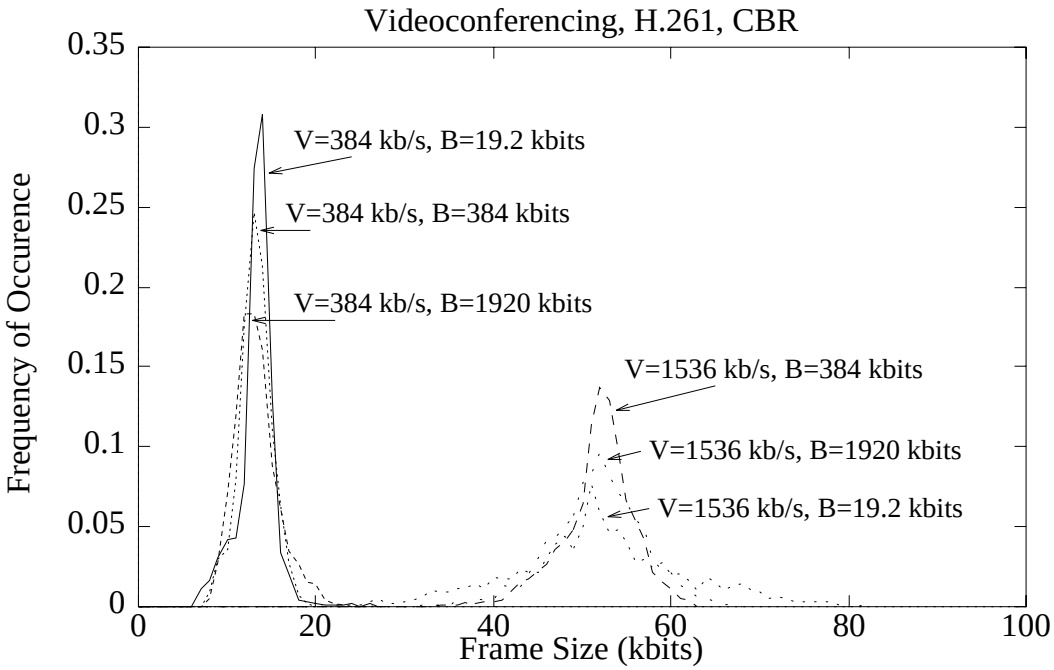
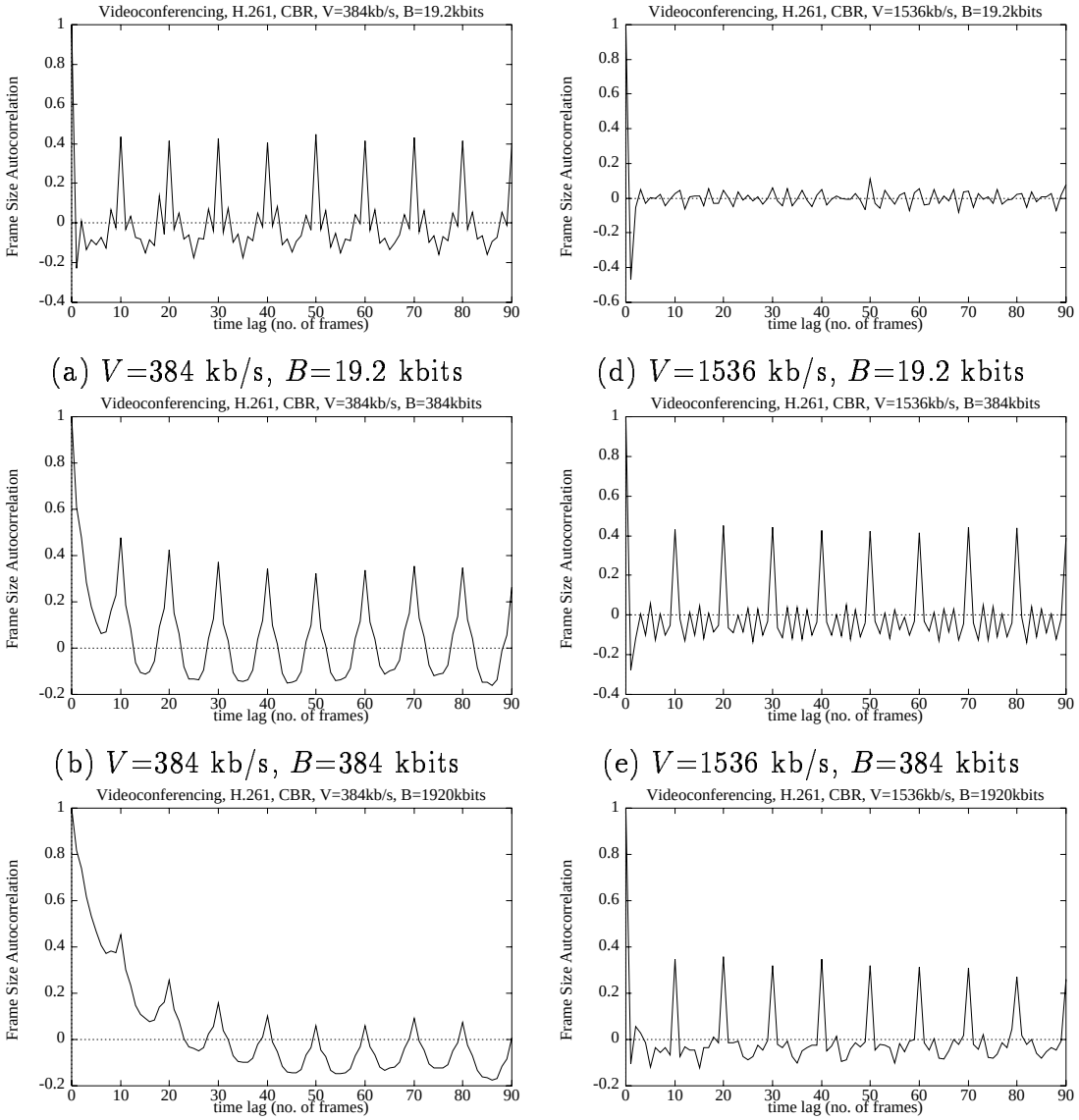


Figure 50: Frame size histogram for Videoconferencing, H.261, CBR.



(a) V=384 kb/s, B=19.2 kbits

(d) V=1536 kb/s, B=19.2 kbits

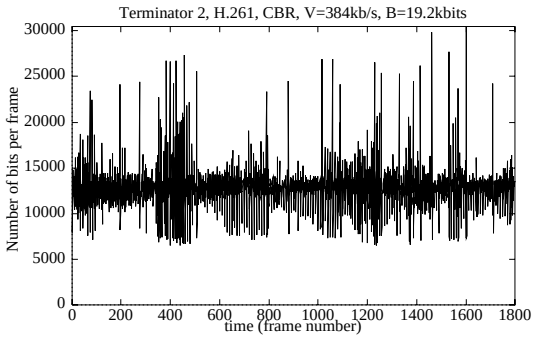
(b) V=384 kb/s, B=384 kbits

(e) V=1536 kb/s, B=384 kbits

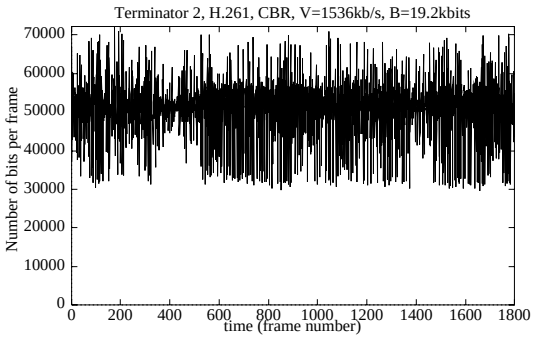
(c) V=384 kb/s, B=1920 kbits

(f) V=1536 kb/s, B=1920 kbits

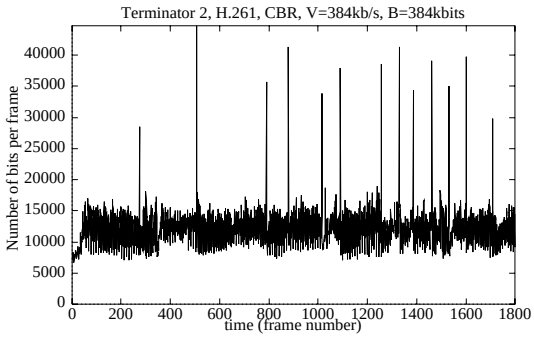
Figure 51: Frame size autocorrelation function for Videoconferencing, H.261, CBR



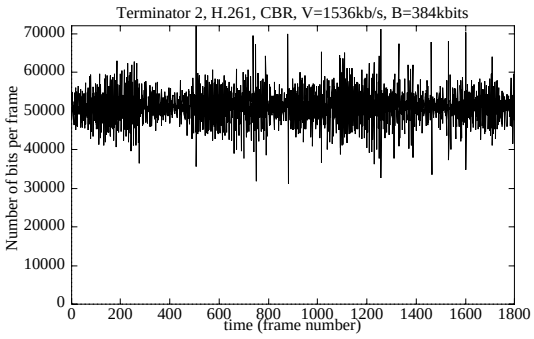
(a) V=384 kb/s, B=19.2 kbits



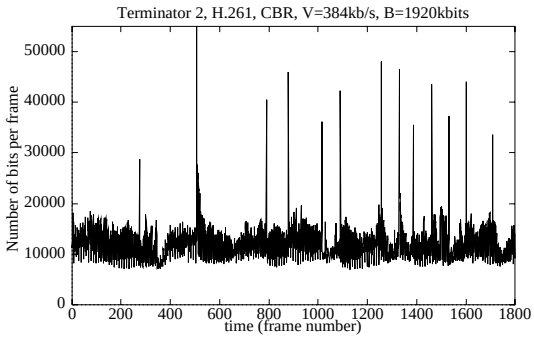
(d) V=1536 kb/s, B=19.2 kbits



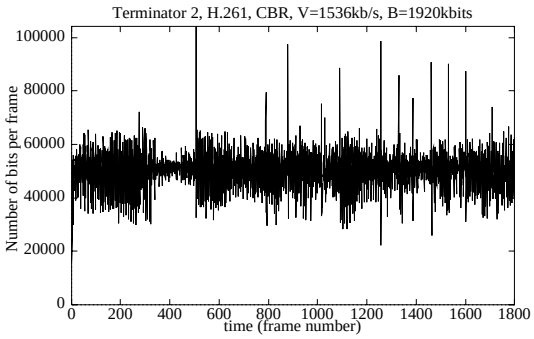
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits



(c) V=384 kb/s, B=1920 kbits



(f) V=1536 kb/s, B=1920 kbits

Figure 52: Number of bits per frame for Terminator-2, H.261, CBR

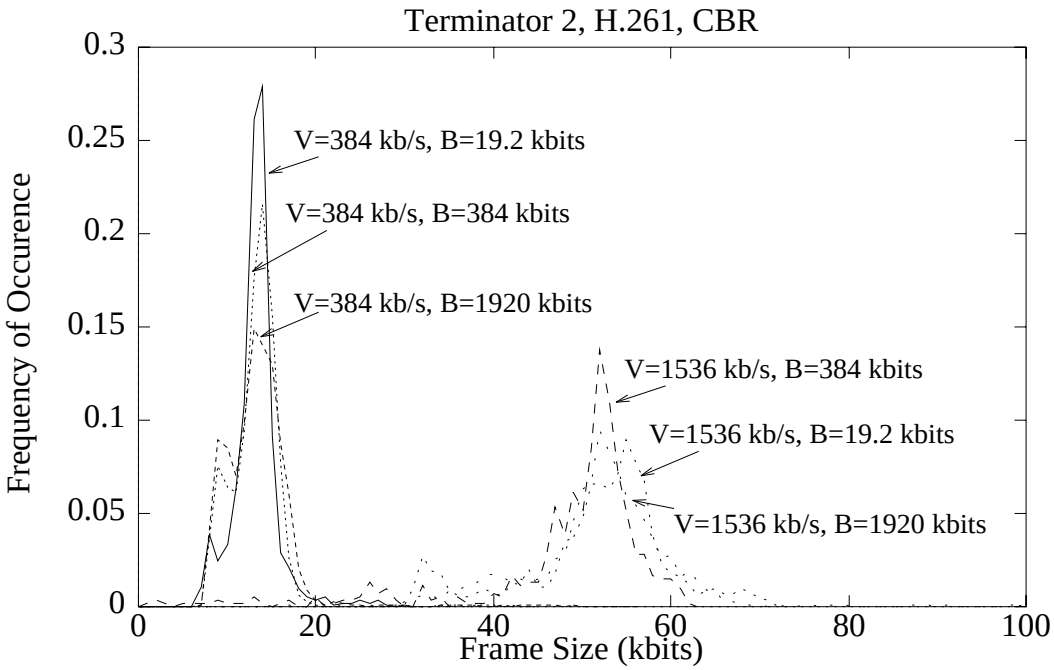
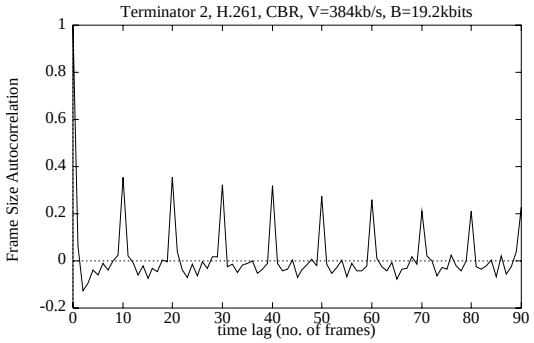
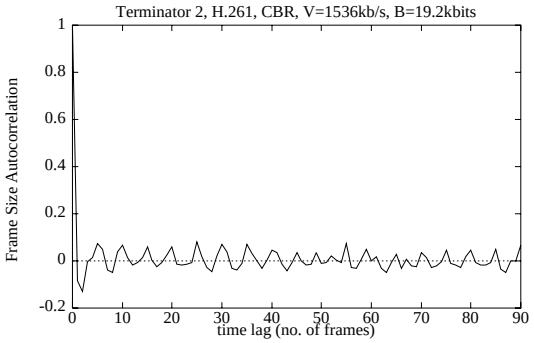


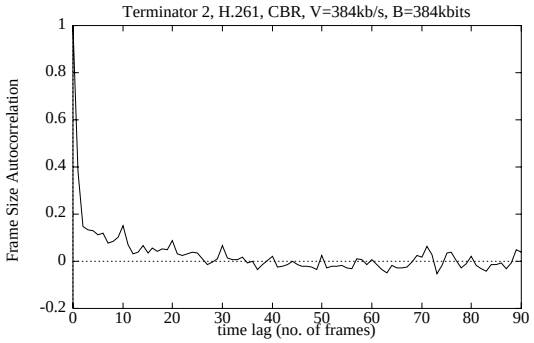
Figure 53: Frame size histogram for Terminator 2, H.261, CBR.



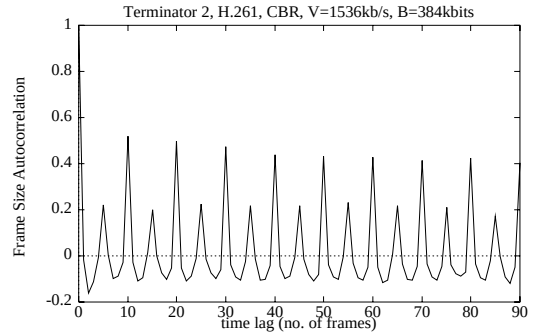
(a) V=384 kb/s, B=19.2 kbits



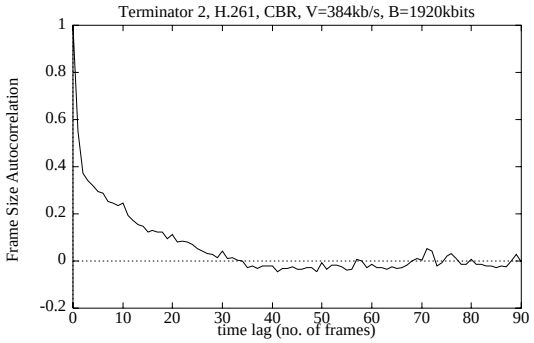
(d) V=1536 kb/s, B=19.2 kbits



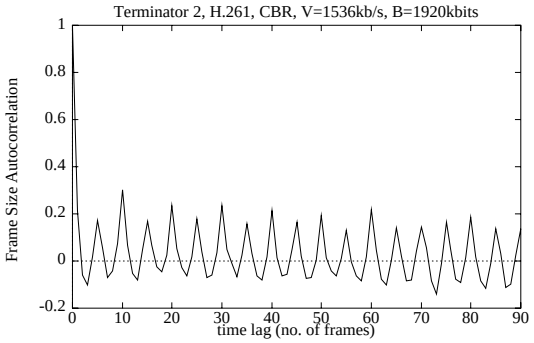
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits

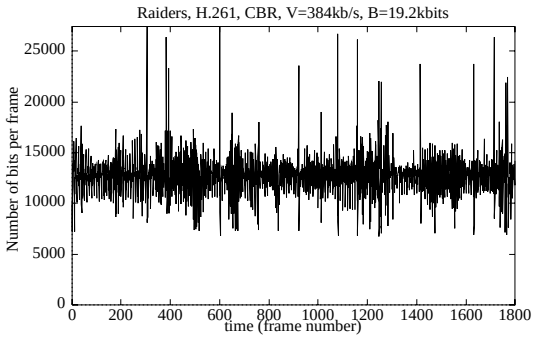


(c) V=384 kb/s, B=1920 kbits

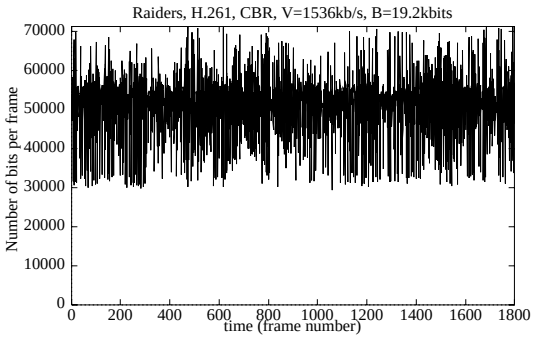


(f) V=1536 kb/s, B=1920 kbits

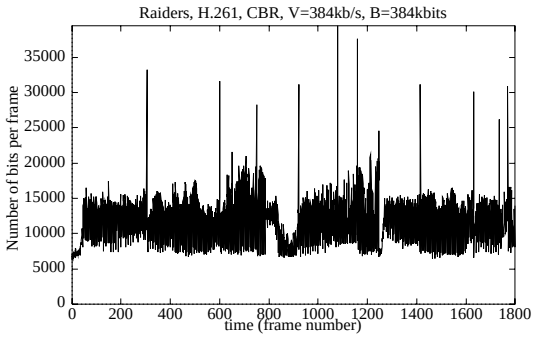
Figure 54: Frame size autocorrelation function for Terminator-2, H.261, CBR



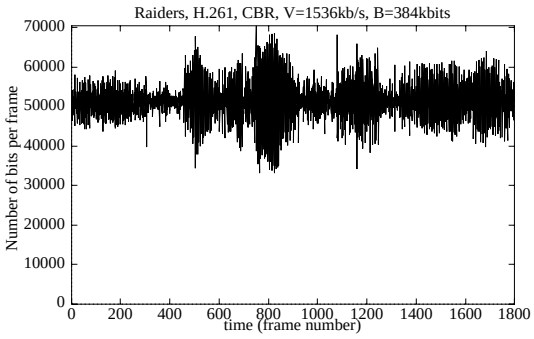
(a) V=384 kb/s, B=19.2 kbits



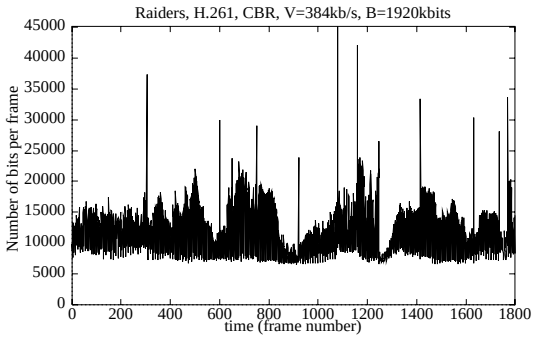
(d) V=1536 kb/s, B=19.2 kbits



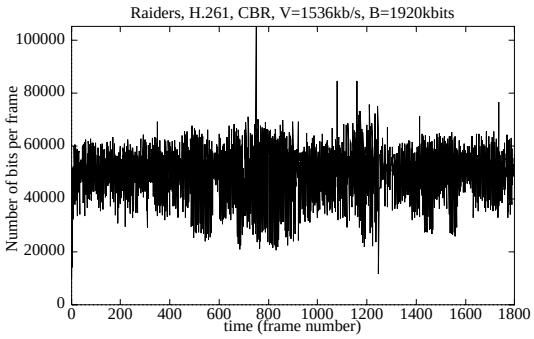
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits



(c) V=384 kb/s, B=1920 kbits



(f) V=1536 kb/s, B=1920 kbits

Figure 55: Number of bits per frame for Raiders, H.261, CBR

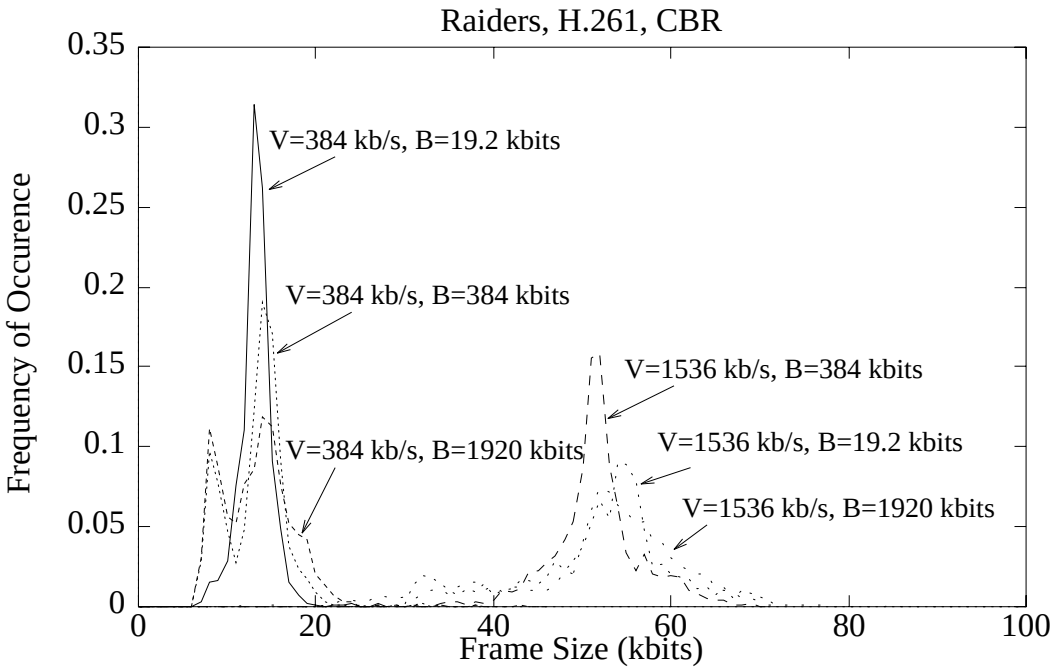
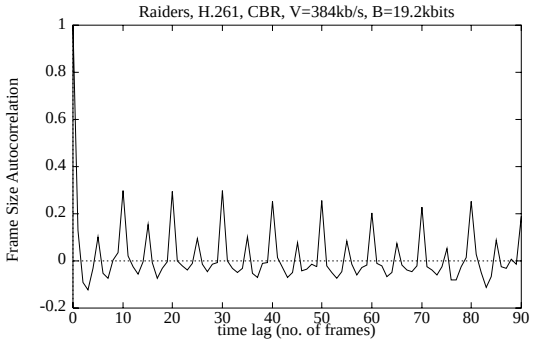
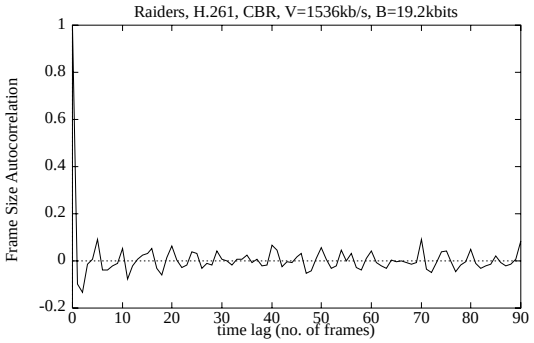


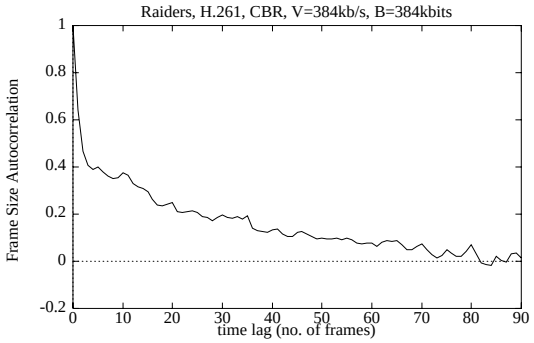
Figure 56: Frame size histogram for Raiders, H.261, CBR.



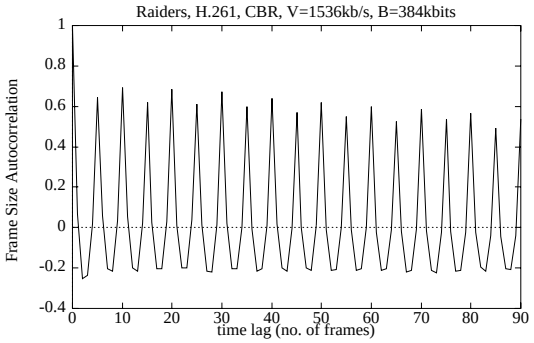
(a) V=384 kb/s, B=19.2 kbits



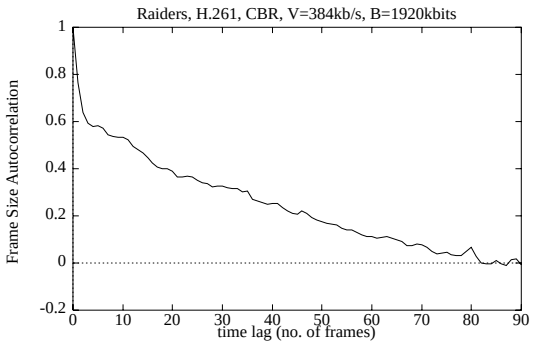
(d) V=1536 kb/s, B=19.2 kbits



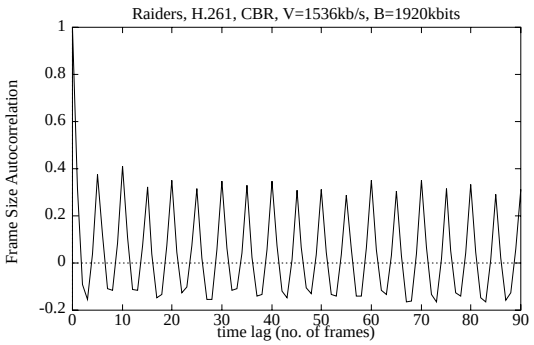
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits

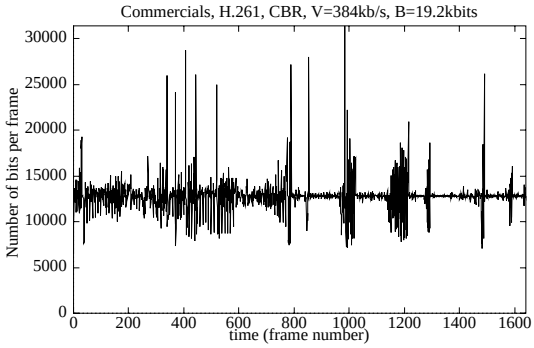


(c) V=384 kb/s, B=1920 kbits

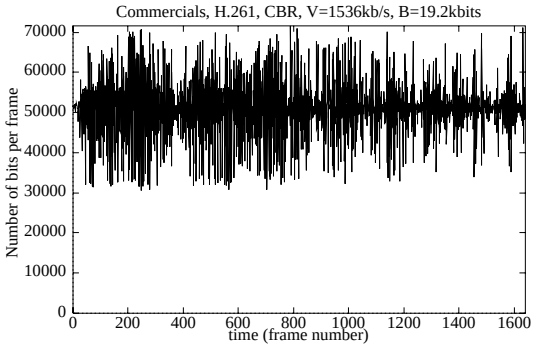


(f) V=1536 kb/s, B=1920 kbits

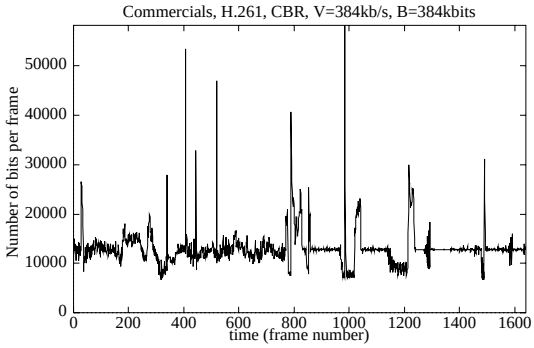
Figure 57: Frame size autocorrelation function for Raiders, H.261, CBR



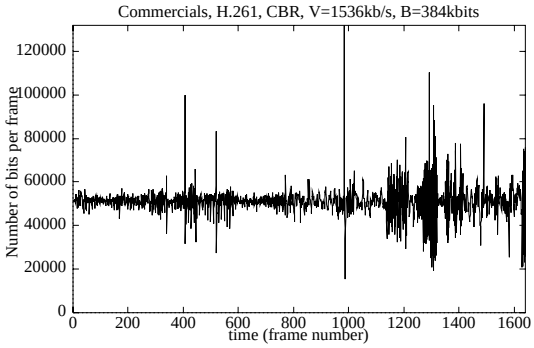
(a) V=384 kb/s, B=19.2 kbits



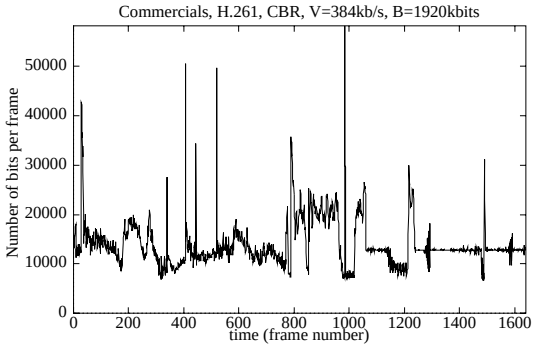
(d) V=1536 kb/s, B=19.2 kbits



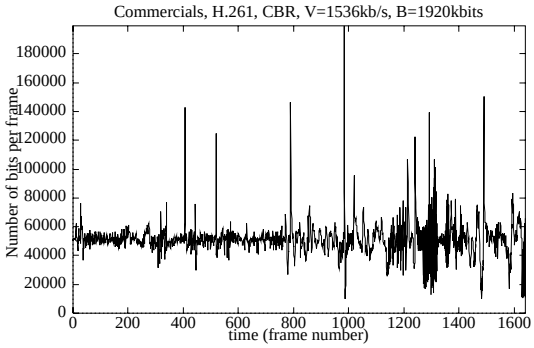
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits



(c) V=384 kb/s, B=1920 kbits



(f) V=1536 kb/s, B=1920 kbits

Figure 58: Number of bits per frame for Commercials, H.261, CBR

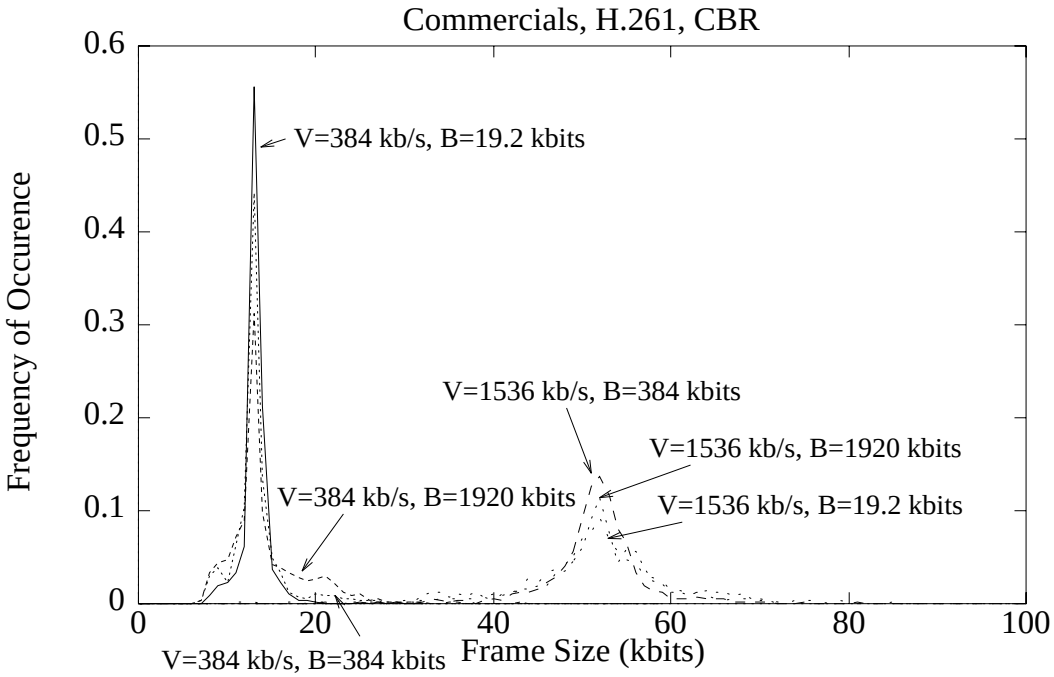
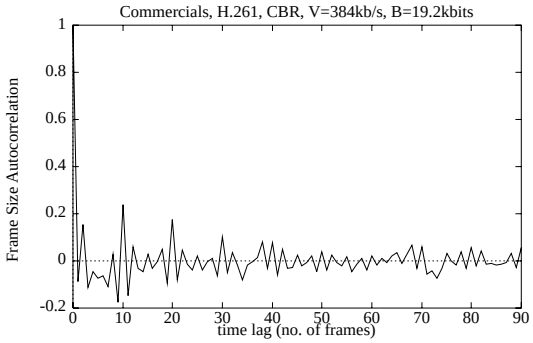
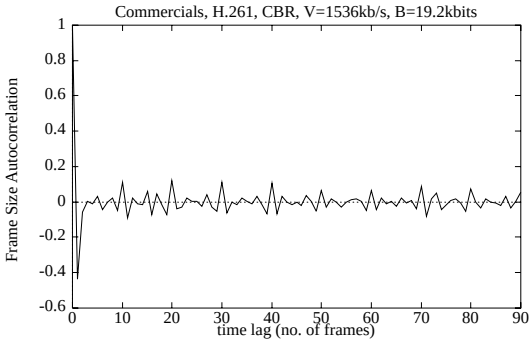


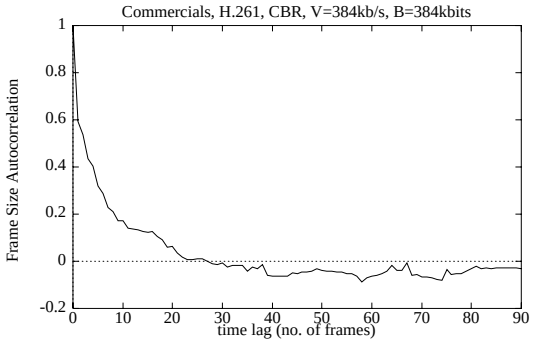
Figure 59: Frame size histogram for Commercials, H.261, CBR.



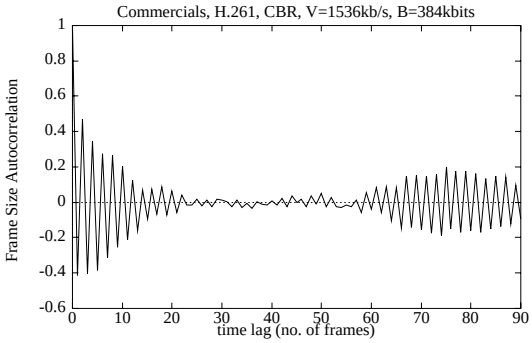
(a) V=384 kb/s, B=19.2 kbits



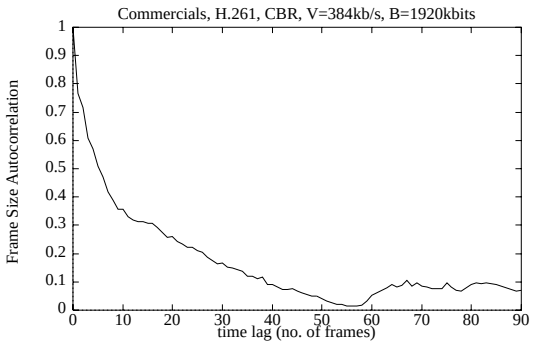
(d) V=1536 kb/s, B=19.2 kbits



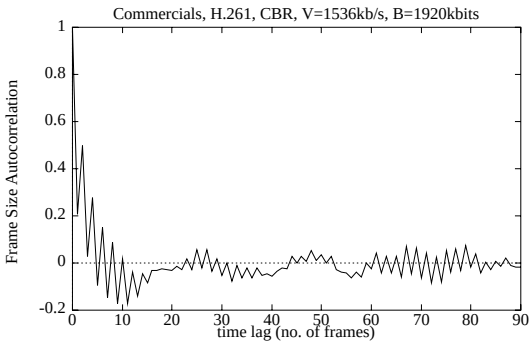
(b) V=384 kb/s, B=384 kbits



(e) V=1536 kb/s, B=384 kbits



(c) V=384 kb/s, B=1920 kbits



(f) V=1536 kb/s, B=1920 kbits

Figure 60: Frame size autocorrelation function for Commercials, H.261, CBR

	Maximum	Minimum	Std. Dev.
Commercials	163084	9981	10157
Raiders	83119	6512	7903
Star Trek	103854	17921	8076
Terminator 2	83684	8114	6565
Videoconferencing	63165	7784	4493

Table 1: Maximum, minimum, and standard deviation of bits per frame for the five sequences, V=1536 kb/s, B=384 kbits.

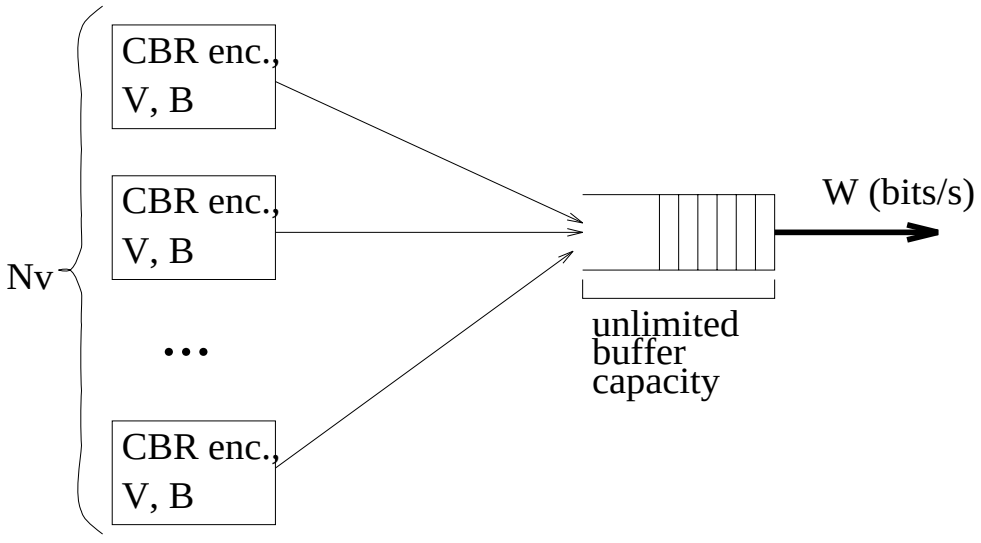
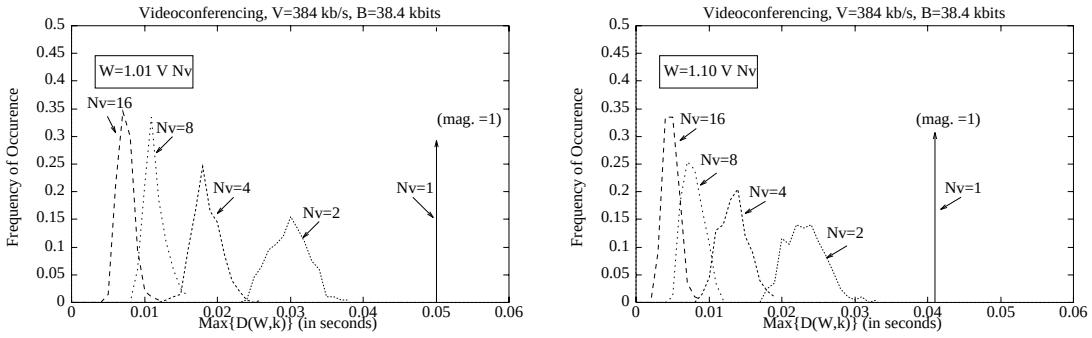
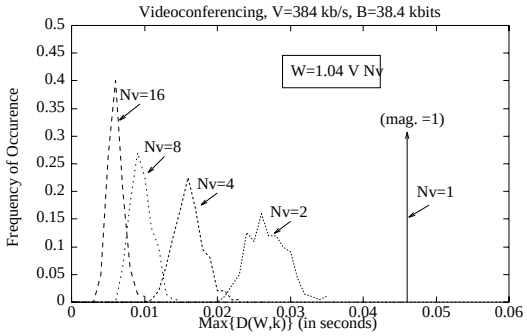


Figure 61: Multiplexing a number of video streams over a circuit

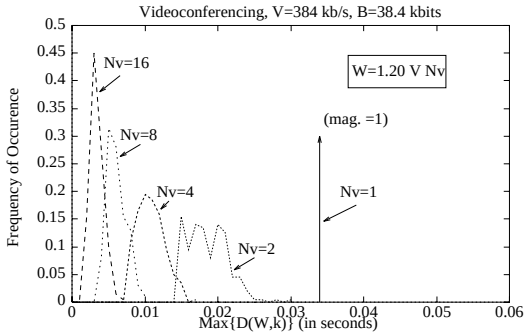


(a) $W = 1.01VN_v$

(c) $W = 1.10VN_v$

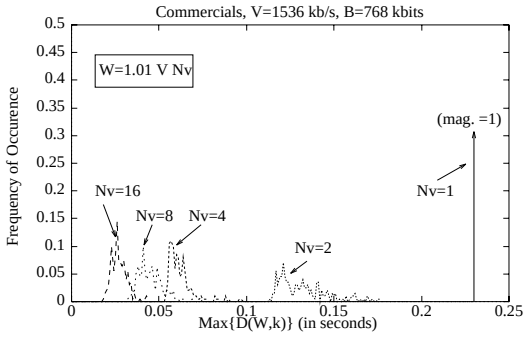


(b) $W = 1.04VN_v$

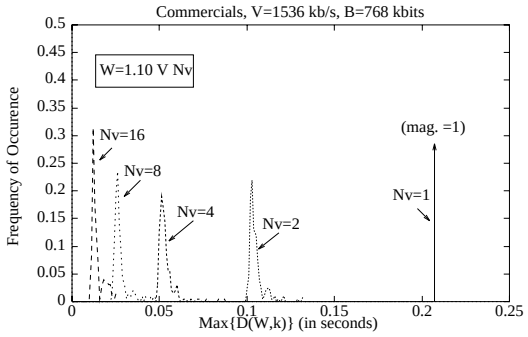


(d) $W = 1.20VN_v$

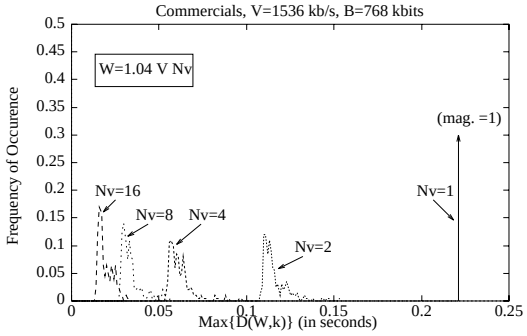
Figure 62: Histogram of $\max_k\{D(W,k)\}$ for Videoconferencing, V=384 kb/s, B=38.4 kbits, various values of W and N_v.



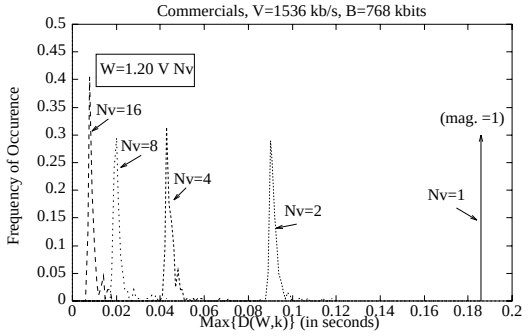
(a) $W = 1.01VN_v$



(c) $W = 1.10VN_v$



(b) $W = 1.04VN_v$



(d) $W = 1.20VN_v$

Figure 63: Histogram of $\max_k\{D(W,k)\}$ for Videoconferencing, V=1536 kb/s, B=768 kbits, various values of W and N_v.

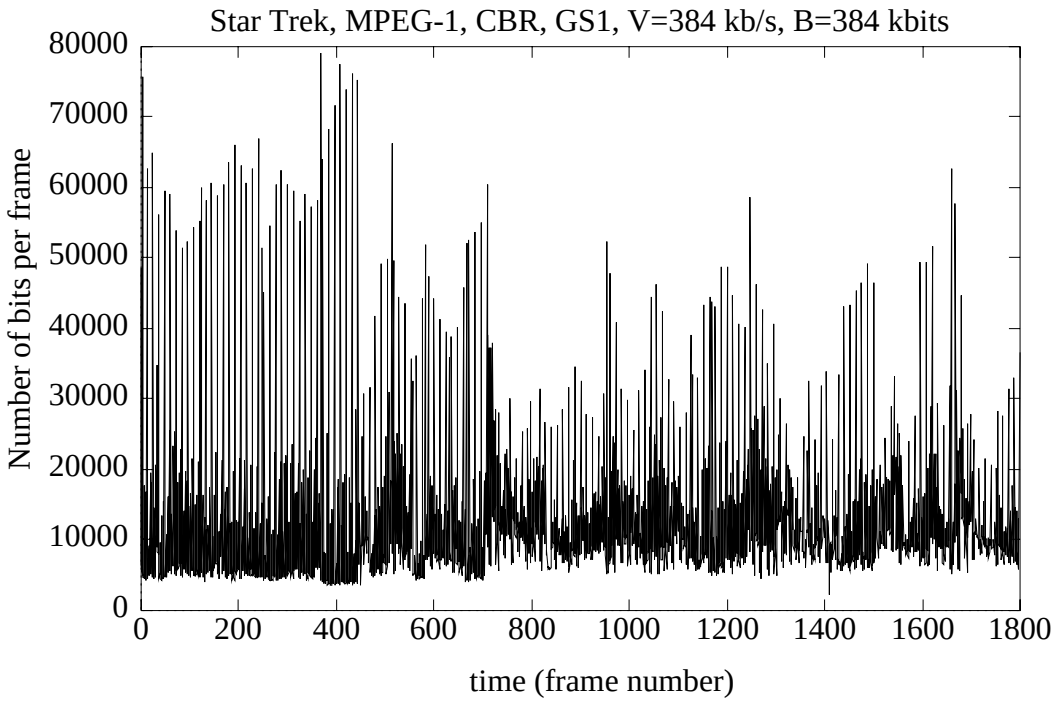


Figure 64: Number of bits per frame for the Star Trek sequence, MPEG, GS1, CBR, V=384 kb/s, B=384 kbits.

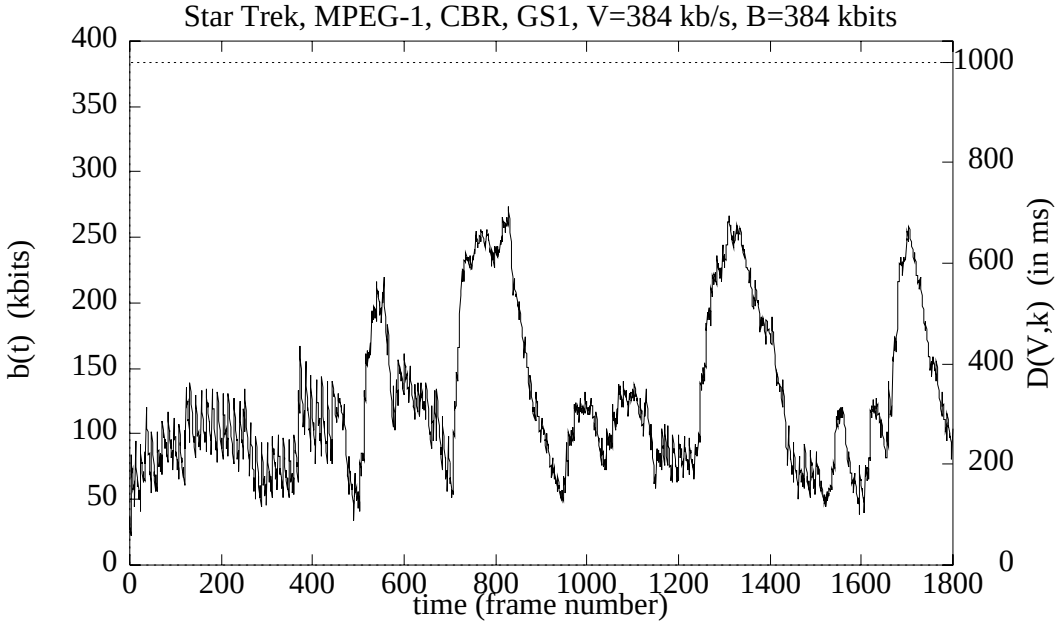
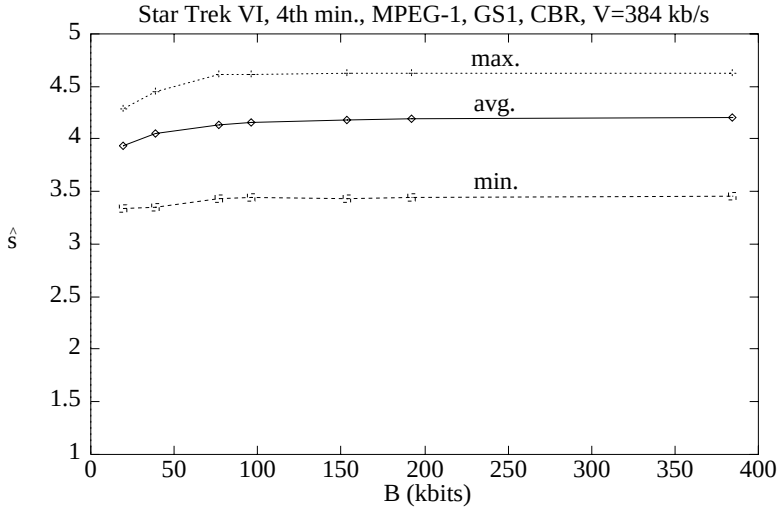
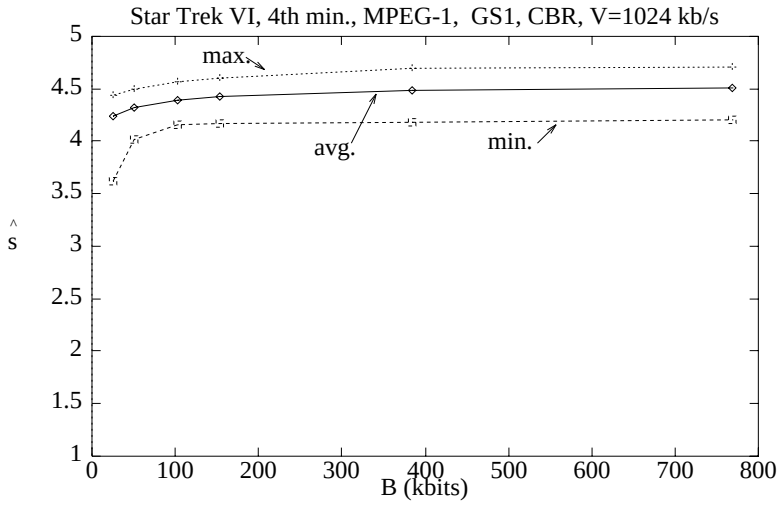


Figure 65: D_r versus time for the Star Trek sequence, MPEG, GS1, CBR, V=384 kb/s, B=384 kbits.



(a) V=384 kb/s



(b) V=1024 kb/s

Figure 66: Maximum, average, and minimum $\hat{s}$ versus B for the Star Trek sequence, MPEG, GS1, CBR.

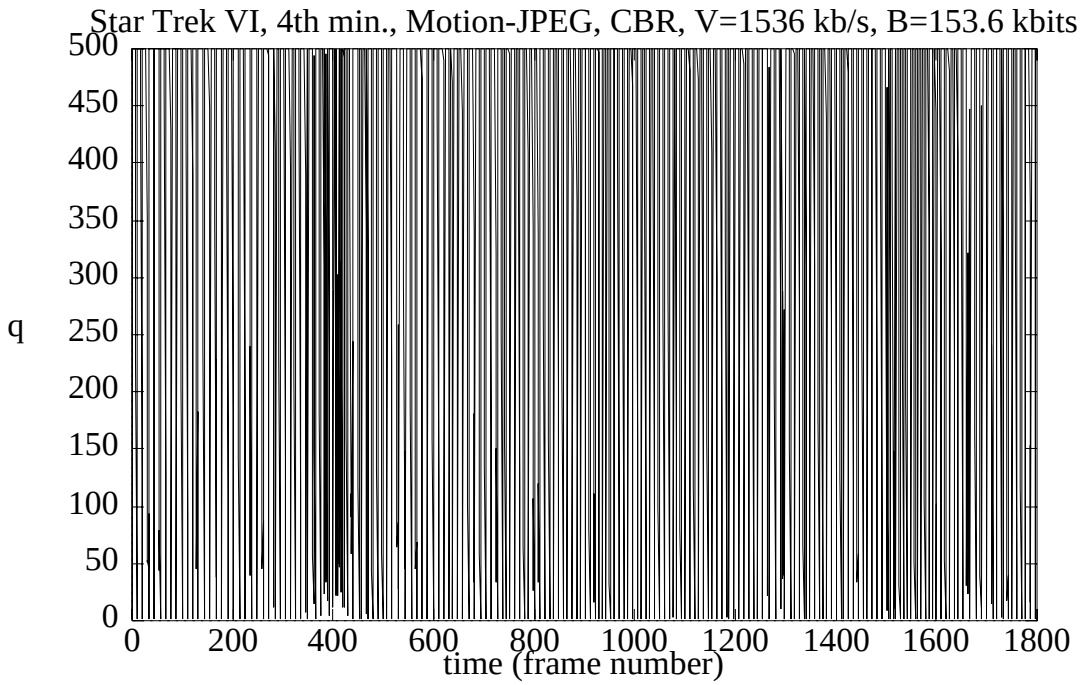


Figure 67: q versus time for the Star Trek sequence, Motion-JPEG, CBR, V=1536 kb/s, B=153.6 kbits.

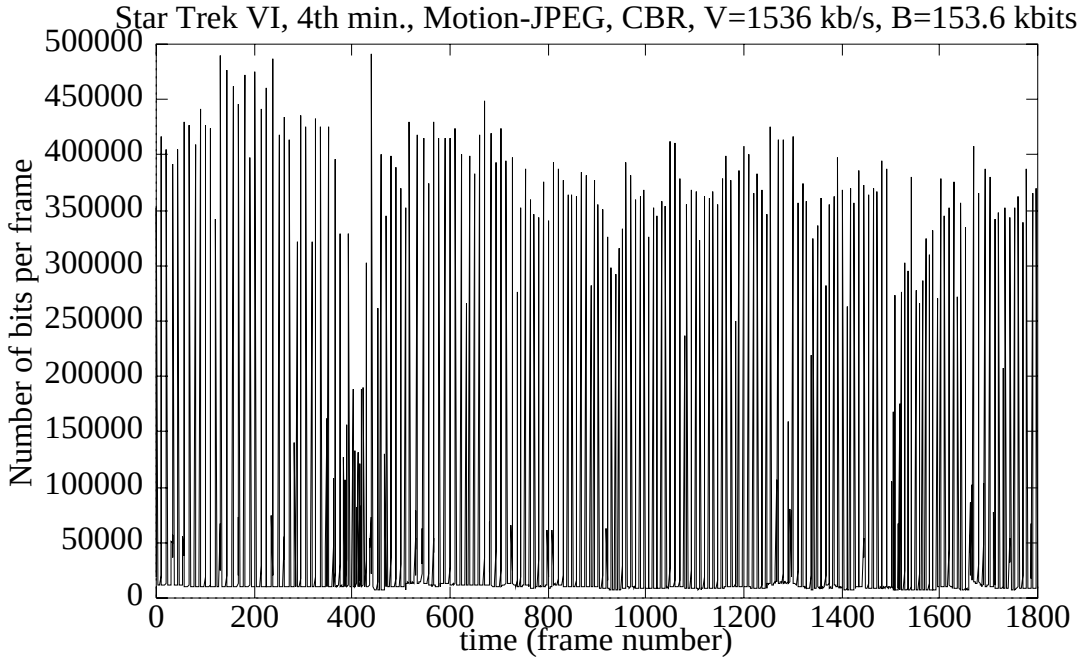


Figure 68: Number of bits per frame for the Star Trek sequence, Motion-JPEG, CBR, V=1536 kb/s, B=153.6 kbits.

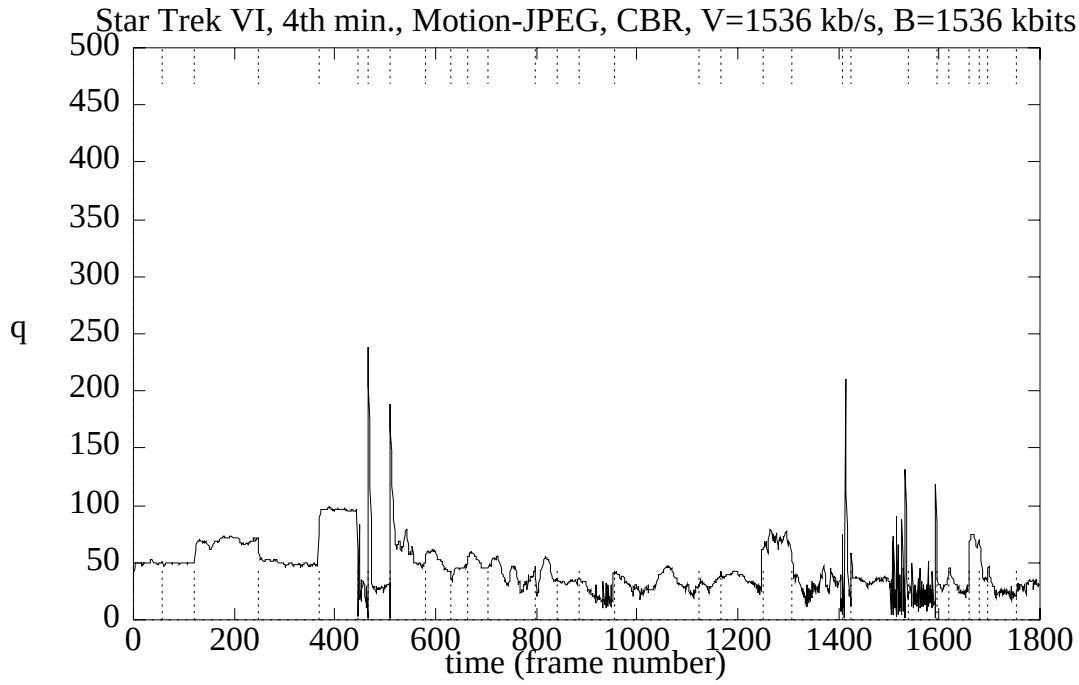


Figure 69: q versus time for the Star Trek sequence, Motion-JPEG, CBR, V=1536 kb/s, B=1536 kbits.

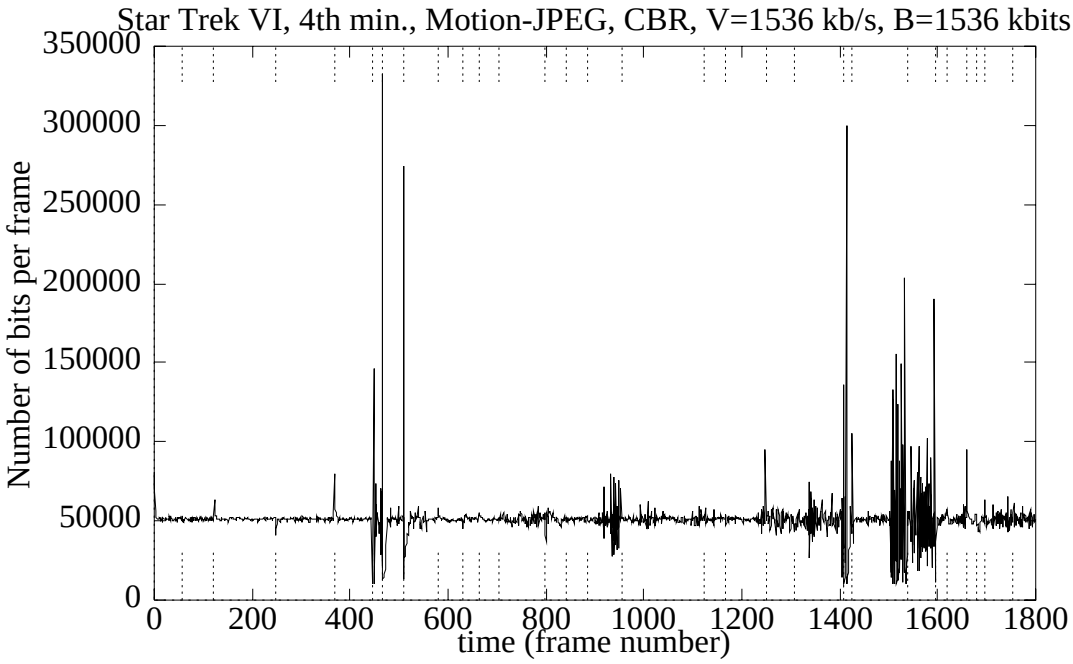


Figure 70: Number of bits per frame for the Star Trek sequence, Motion-JPEG, CBR, V=1536 kb/s, B=1536 kbits.

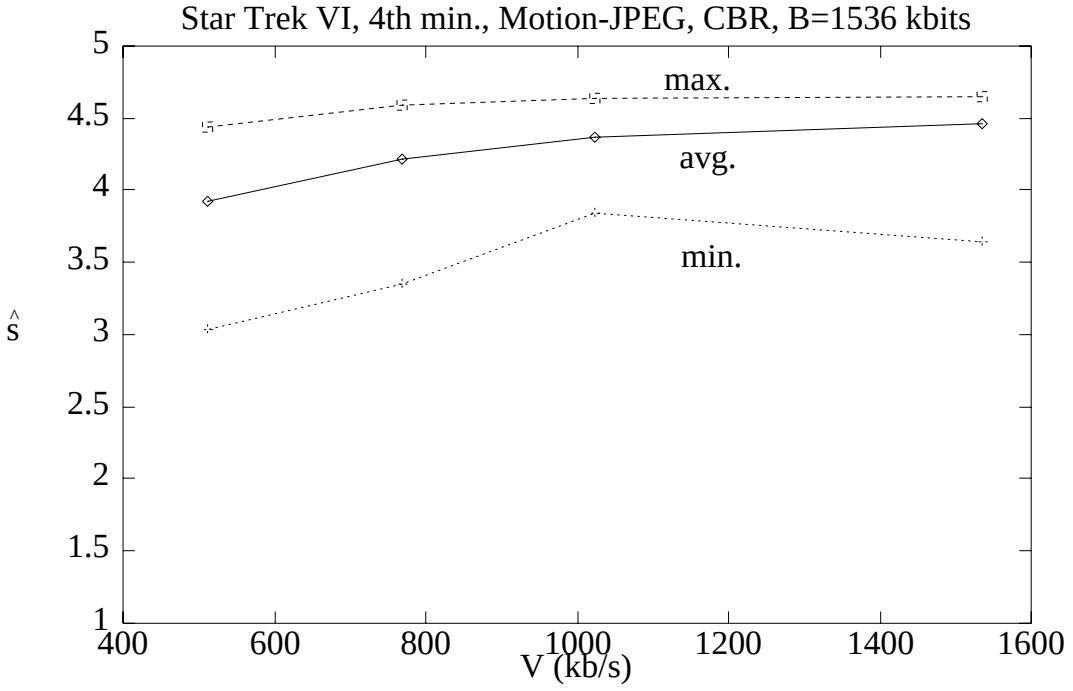


Figure 71: Minimum, average, and maximum $\hat{s}$ versus V for the Star Trek sequence, Motion-JPEG, CBR, B=1536 kbits.

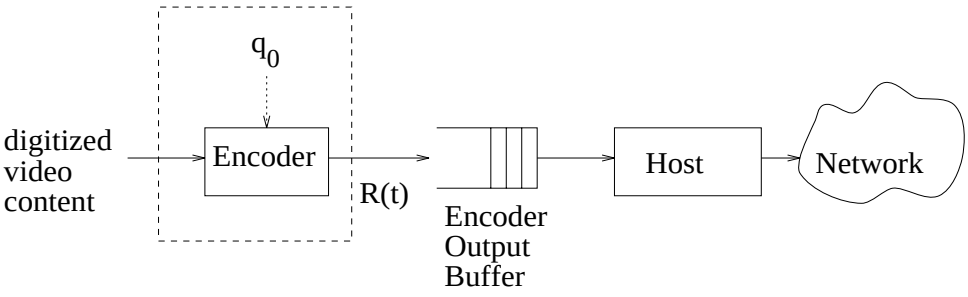
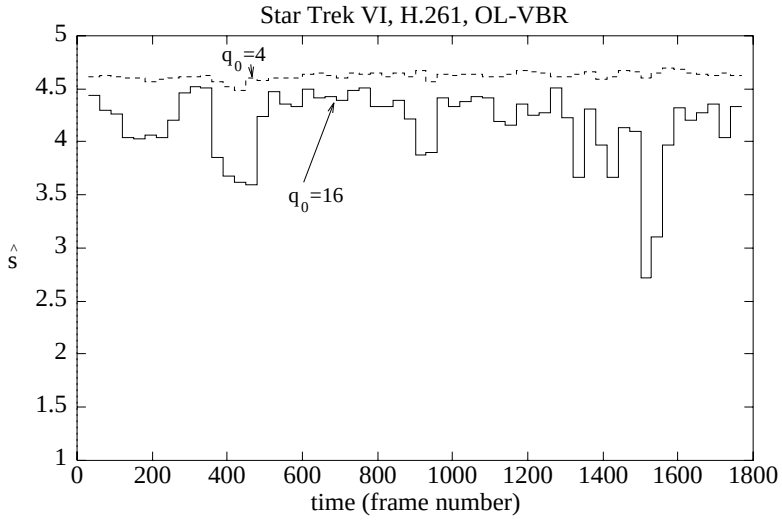
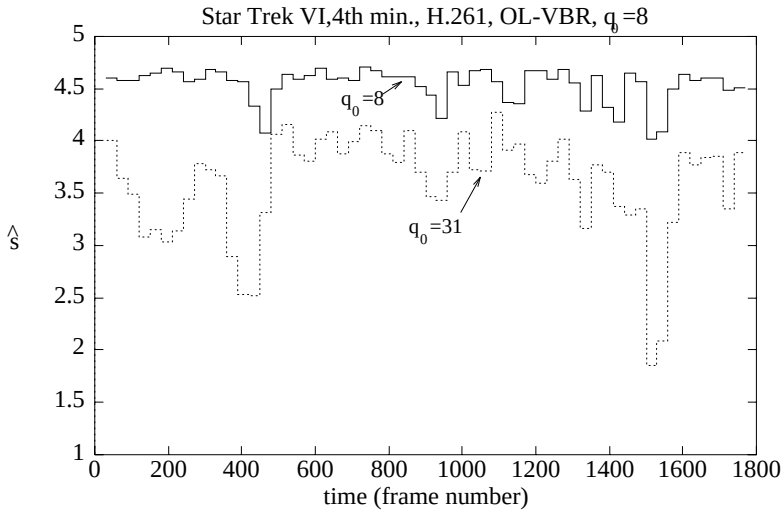


Figure 72: Block diagram of the encoder for Open-Loop VBR encoding



(a) q_0={4,16}



(b) q_0={8,31}

Figure 73: $\hat{s}$ versus time for the Star Trek sequence, H.261, OL-VBR.

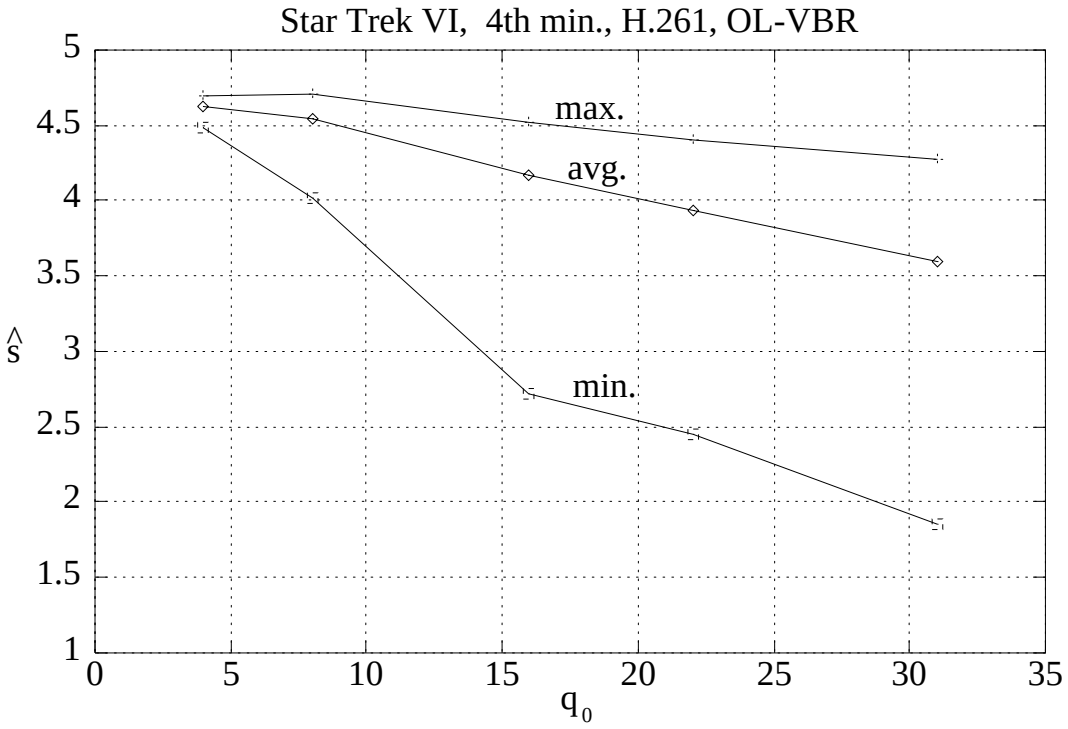
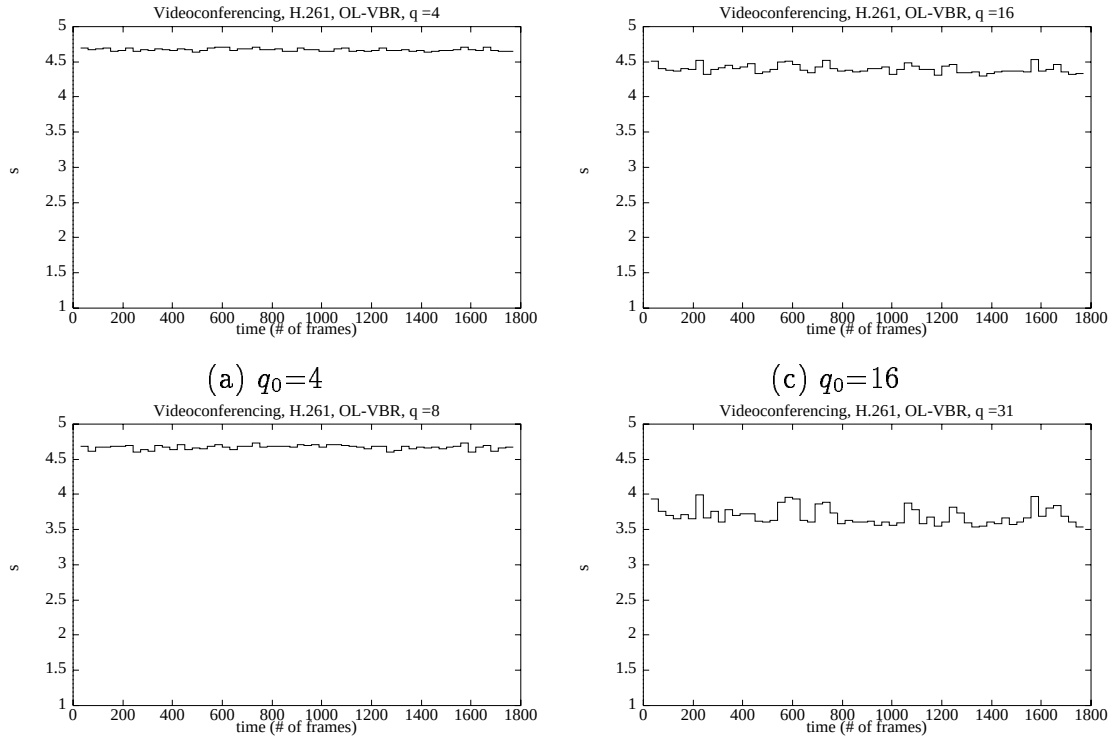


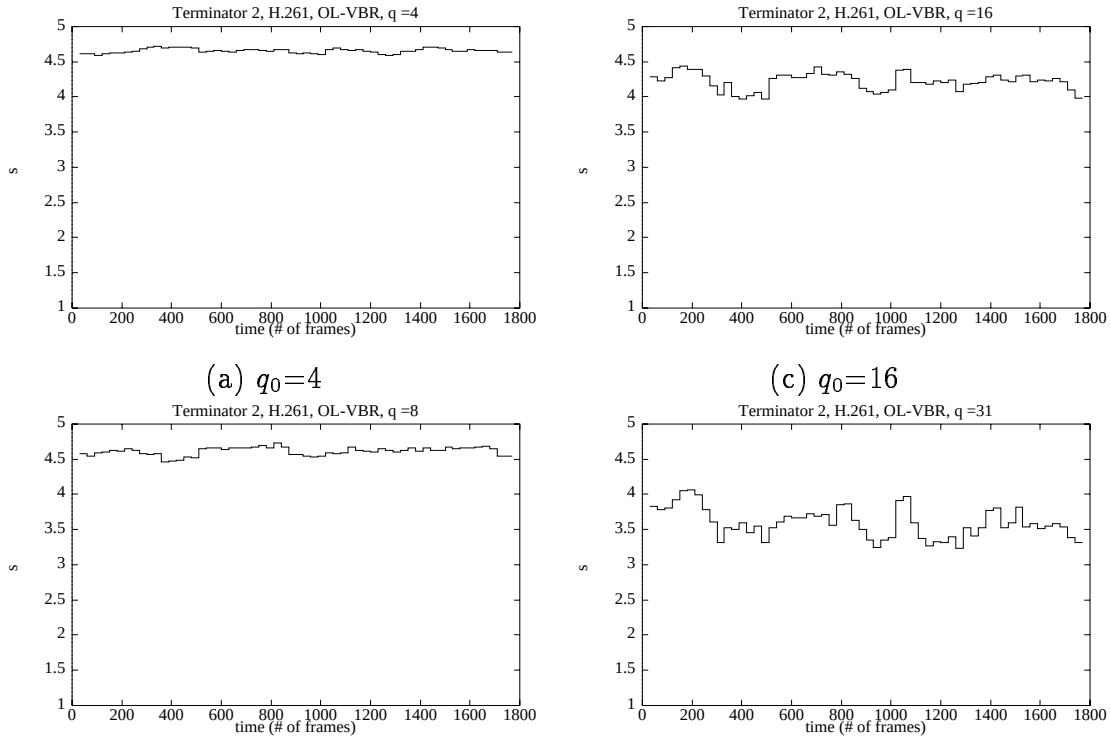
Figure 74: Maximum, average, and minimum $\hat{s}$ versus q_0 for the Star Trek sequence, H.261, OL-VBR.



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

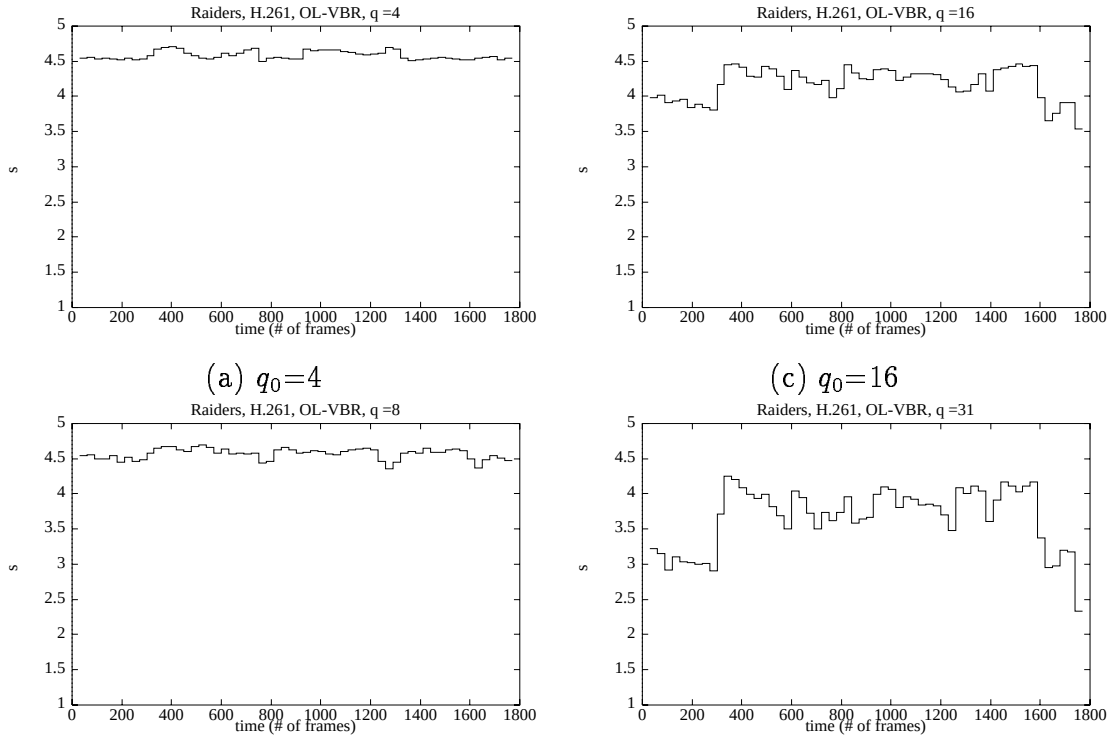
Figure 75: $\hat{s}$ versus time for Videoconferencing, H.261, OL-VBR



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

Figure 76: $\hat{s}$ versus time for Terminator 2, H.261, OL-VBR



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

Figure 77: $\hat{s}$ versus time for Raiders, H.261, OL-VBR

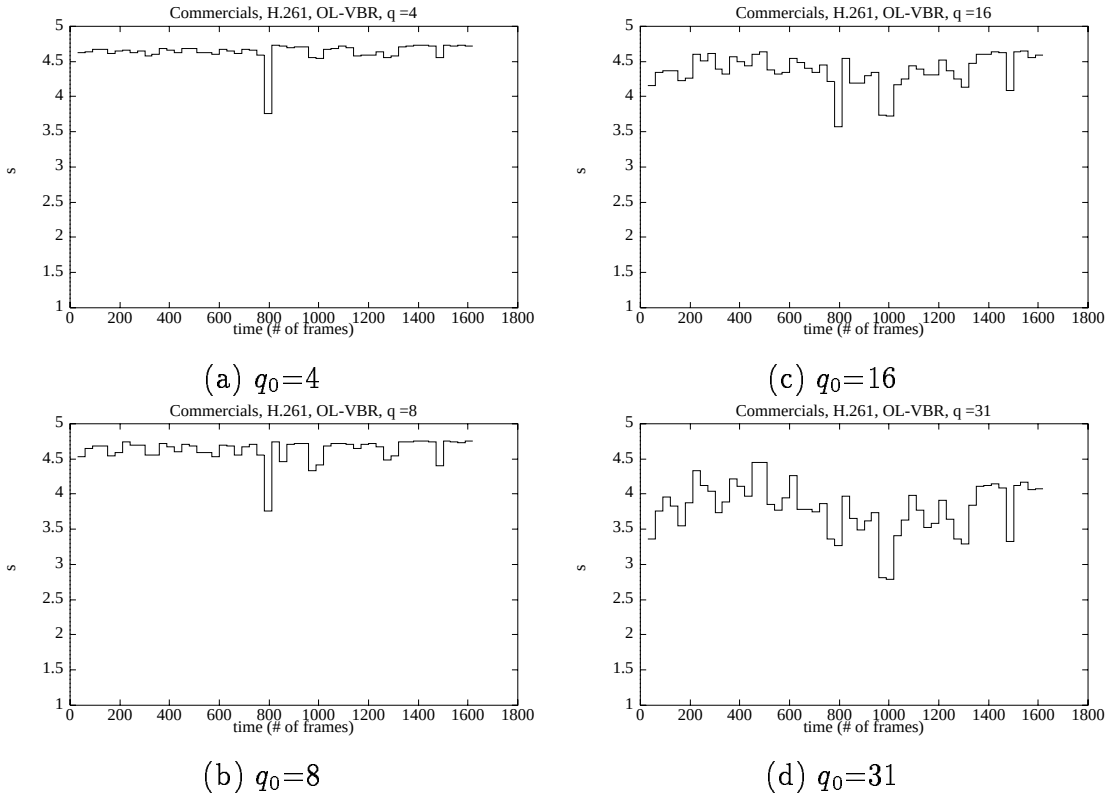


Figure 78: $\hat{s}$ versus time for Commercials, H.261, OL-VBR

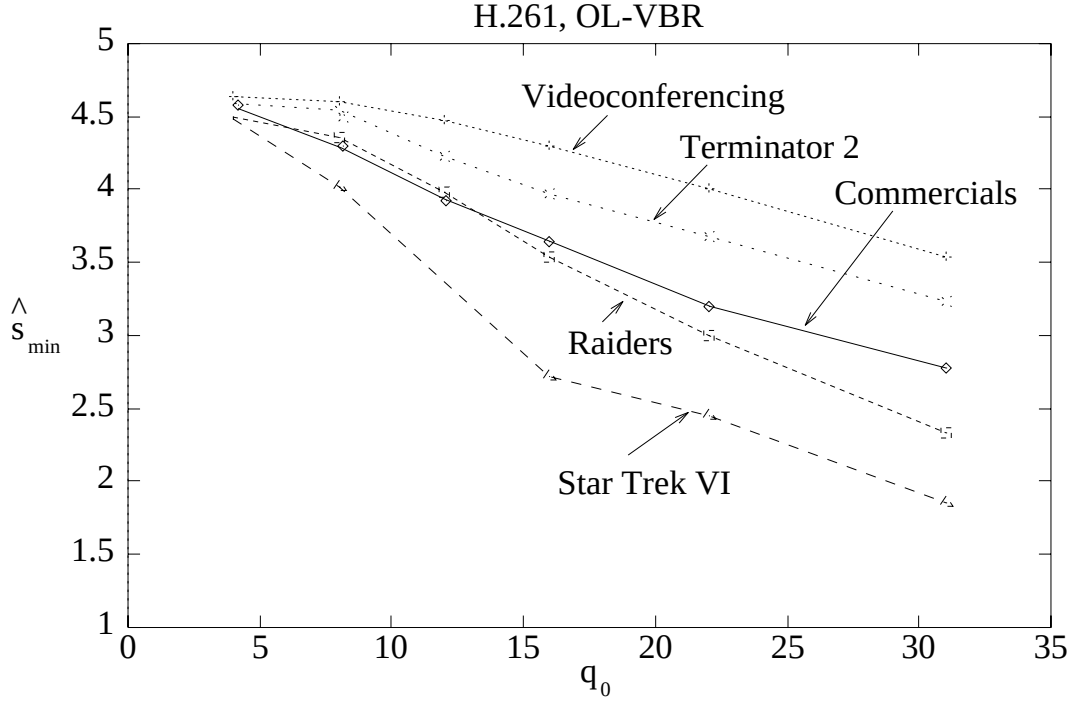
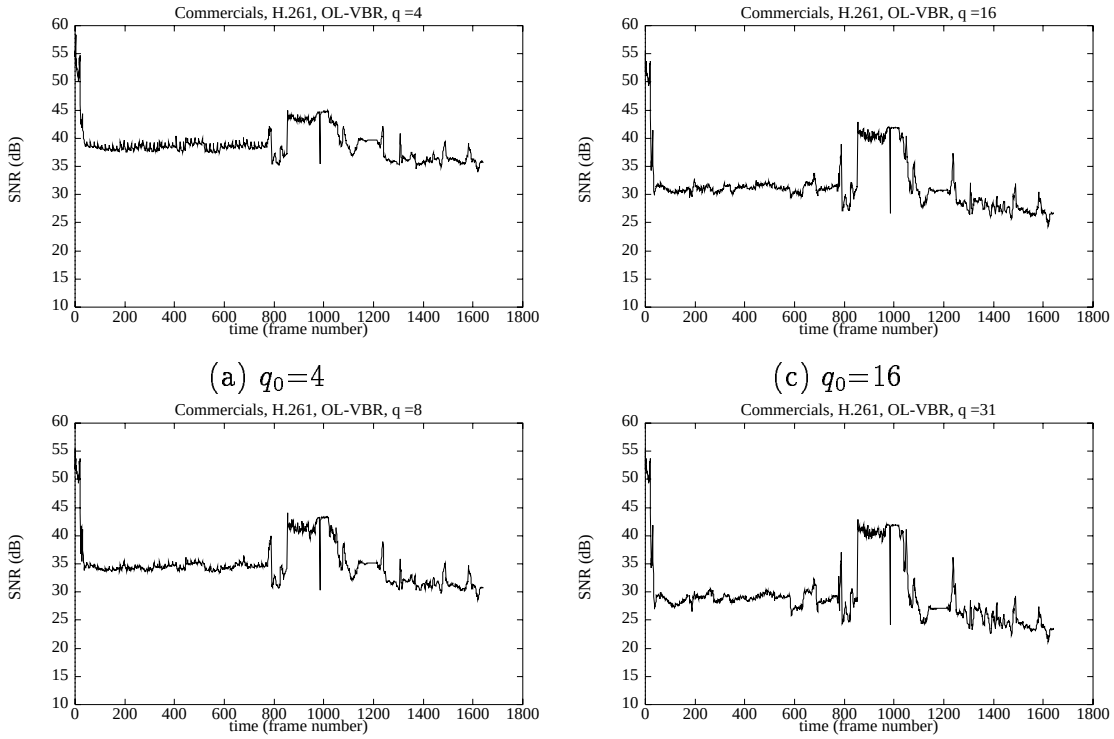


Figure 79: Minimum quality versus q_0 for various sequences, H.261, OL-VBR.



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

Figure 80: SNR versus time for Commercials, H.261, OL-VBR

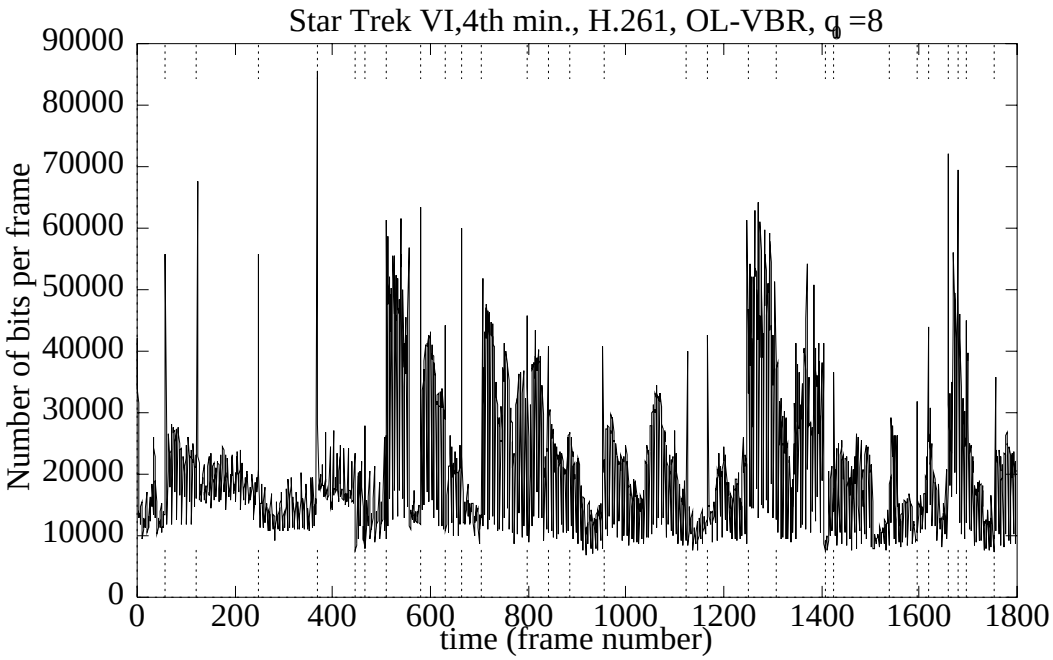


Figure 81: Number of bits per frame for the Star Trek sequence, H.261, OL-VBR, q_0=8, every frame is shown. (Resulting average frame size = 20.9 kbits.)

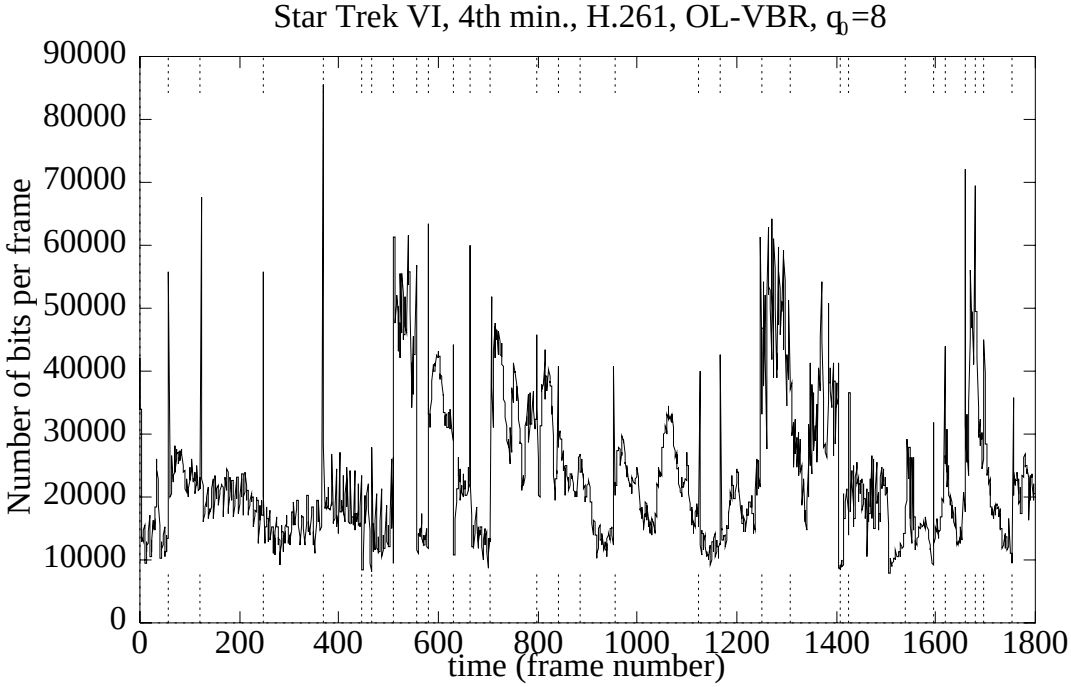
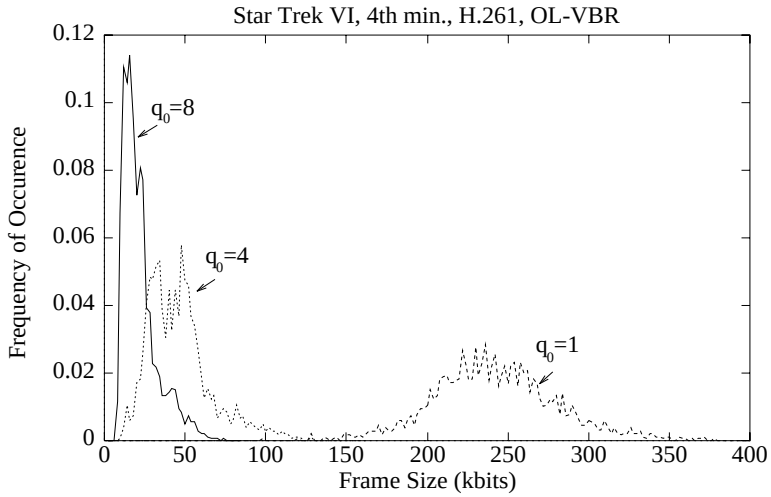
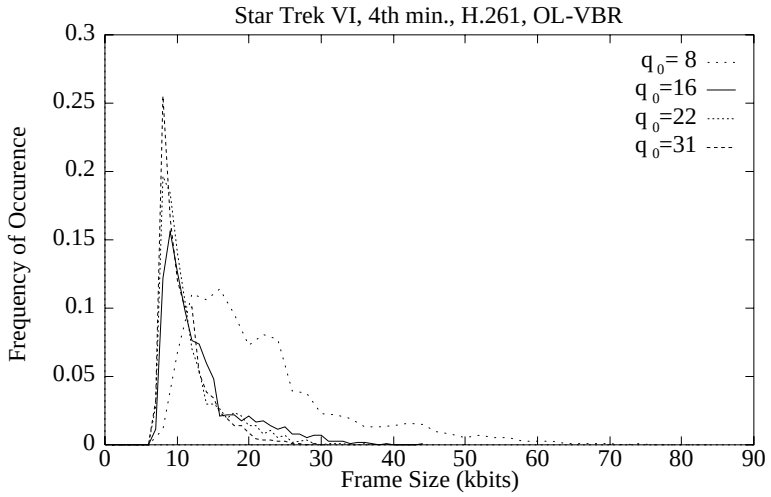


Figure 82: Number of bits per frame for the Star Trek sequence, H.261, OL-VBR, q_0=8, repeated frames not shown.

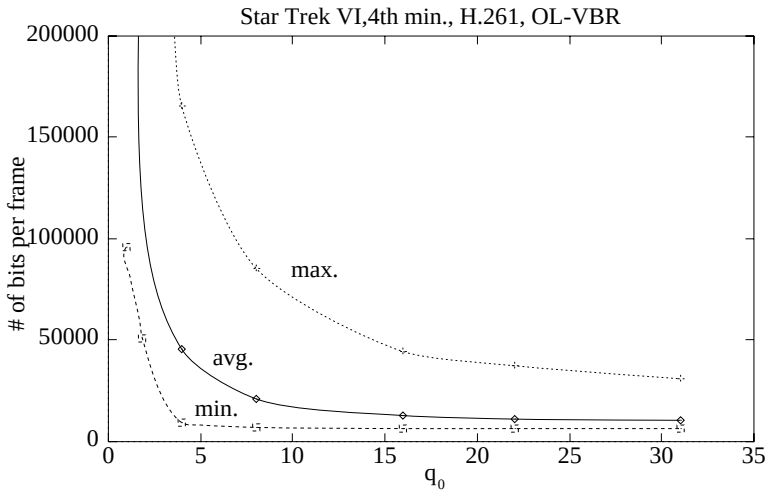


(a)

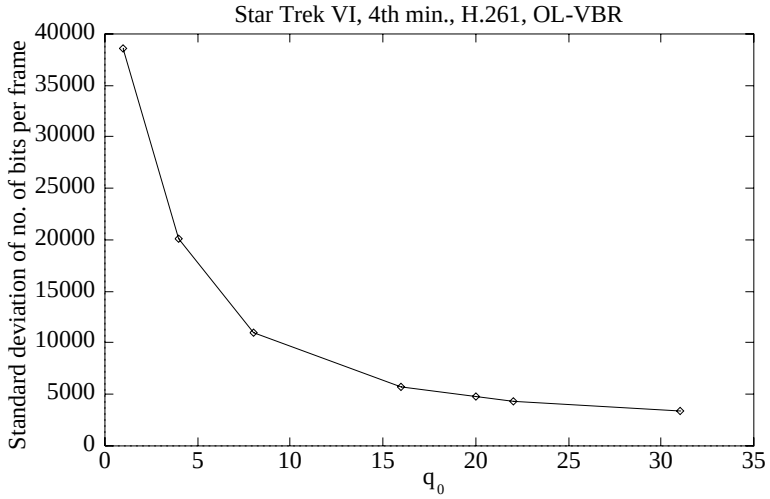


(b)

Figure 83: Frame size histogram for various values of q_0 for the Star Trek sequence, H.261, OL-VBR.

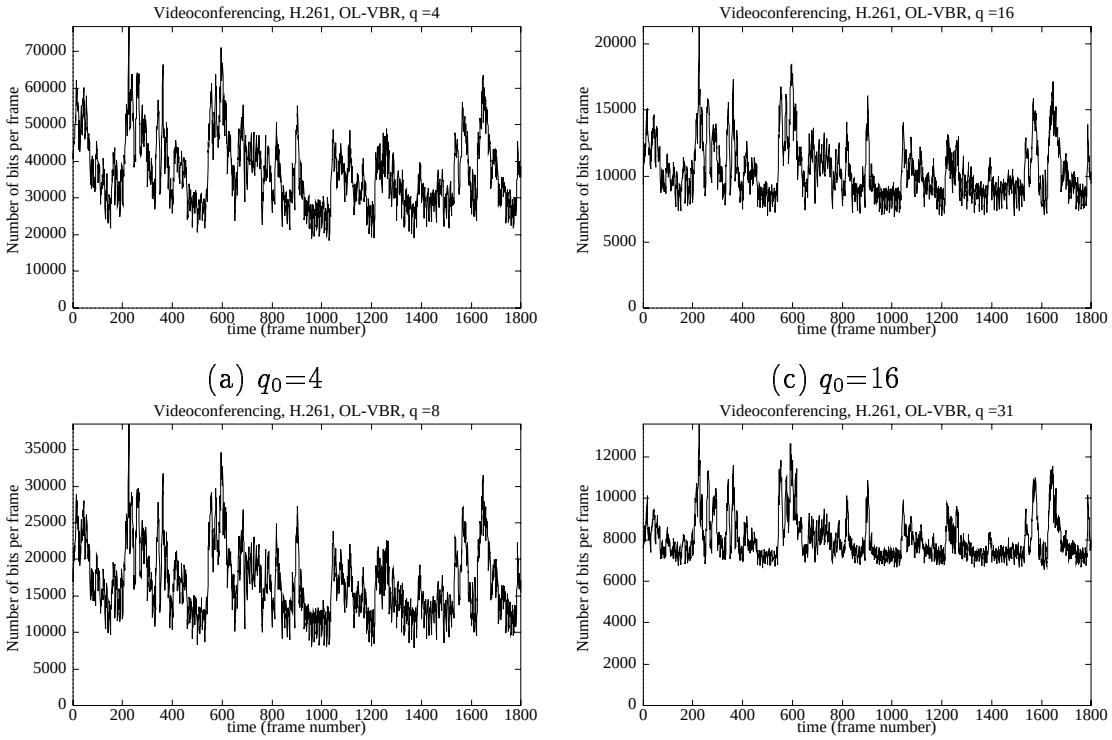


(a) Maximum, average, and minimum number of bits per frame



(b) Standard deviation of number of bits per frame

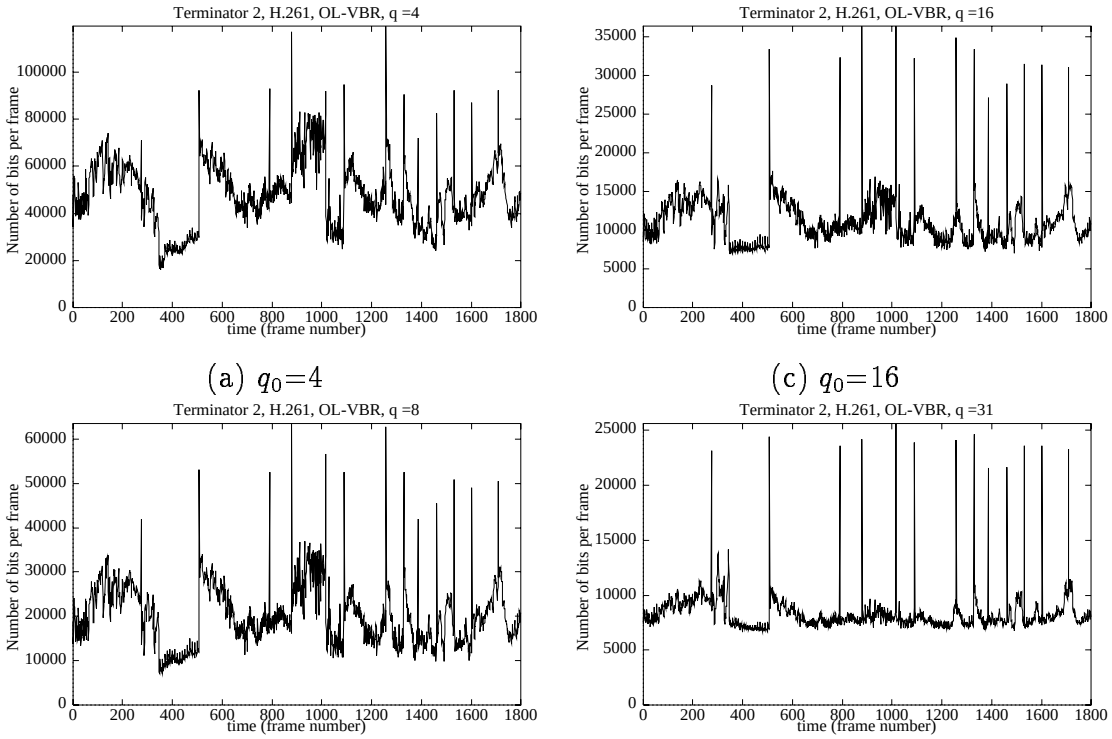
Figure 84: Maximum, average, minimum, and standard deviation of number of bits per frame versus q_0 for the Star Trek sequence, H.261, OL-VBR.



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

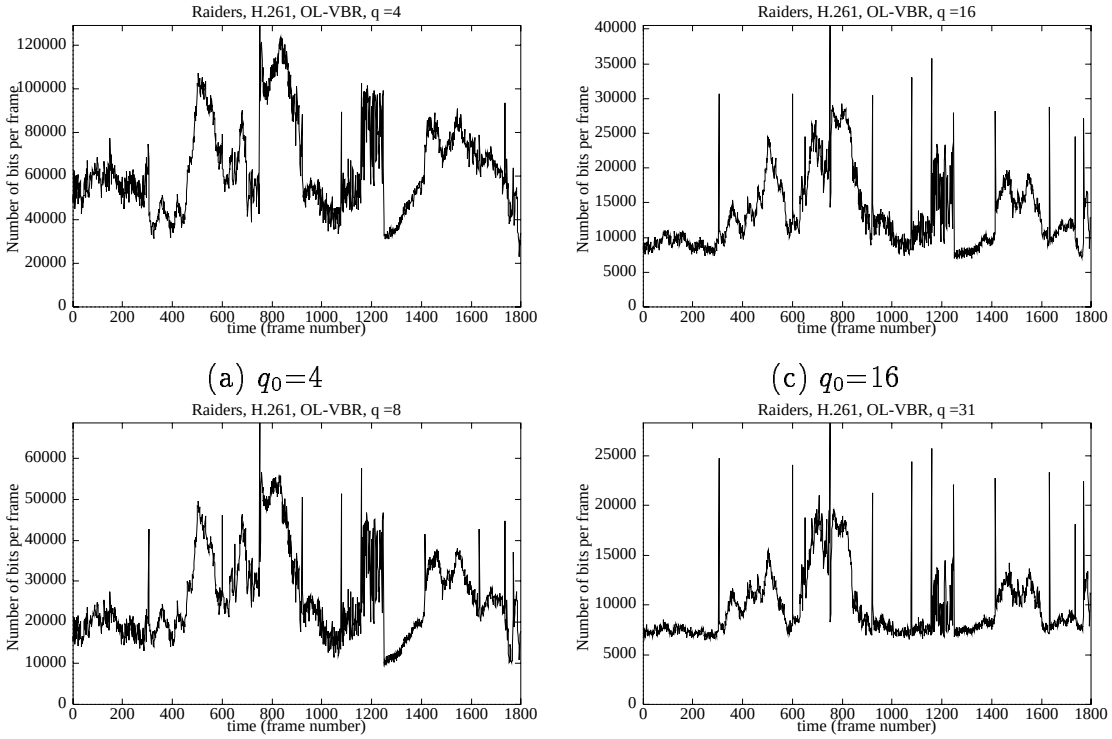
Figure 85: Number of bits per frame for Videoconferencing, H.261, OL-VBR



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

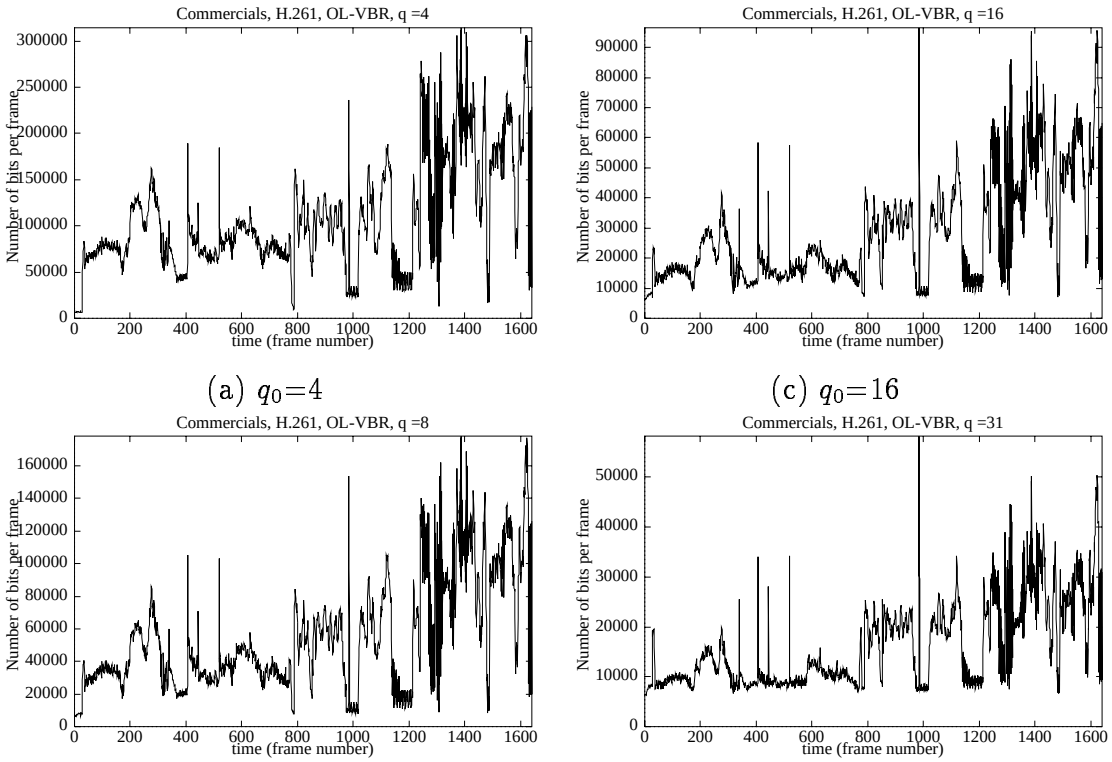
Figure 86: Number of bits per frame for Terminator 2, H.261, OL-VBR



(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

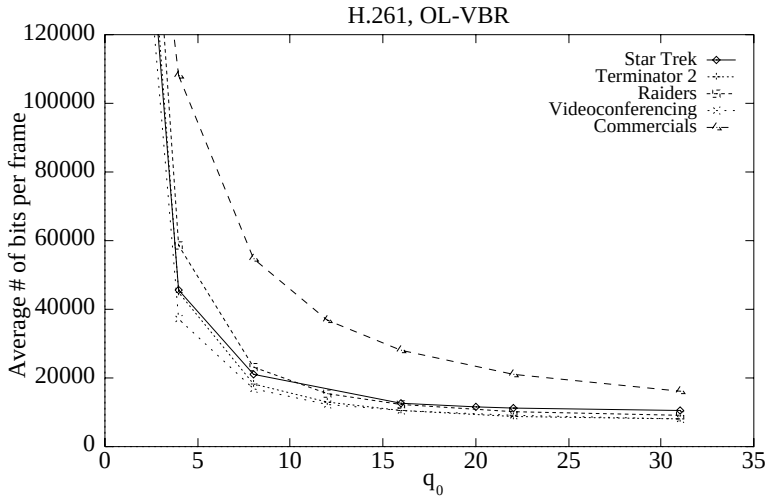
Figure 87: Number of bits per frame for Raiders, H.261, OL-VBR



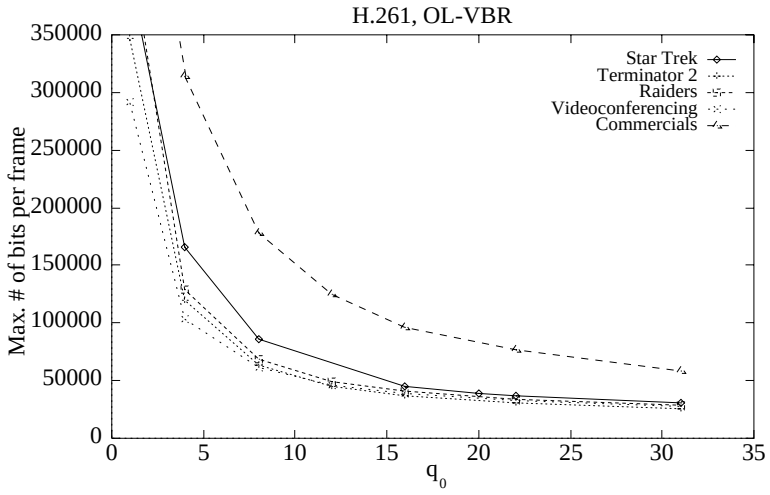
(a) q_0=4 (c) q_0=16

(b) q_0=8 (d) q_0=31

Figure 88: Number of bits per frame for Commercials, H.261, OL-VBR



(a) Average number of bits per frame



(b) Maximum number of bits per frame

Figure 89: Average and maximum number of bits per frame versus q_0 for various sequences, H.261, OL-VBR.

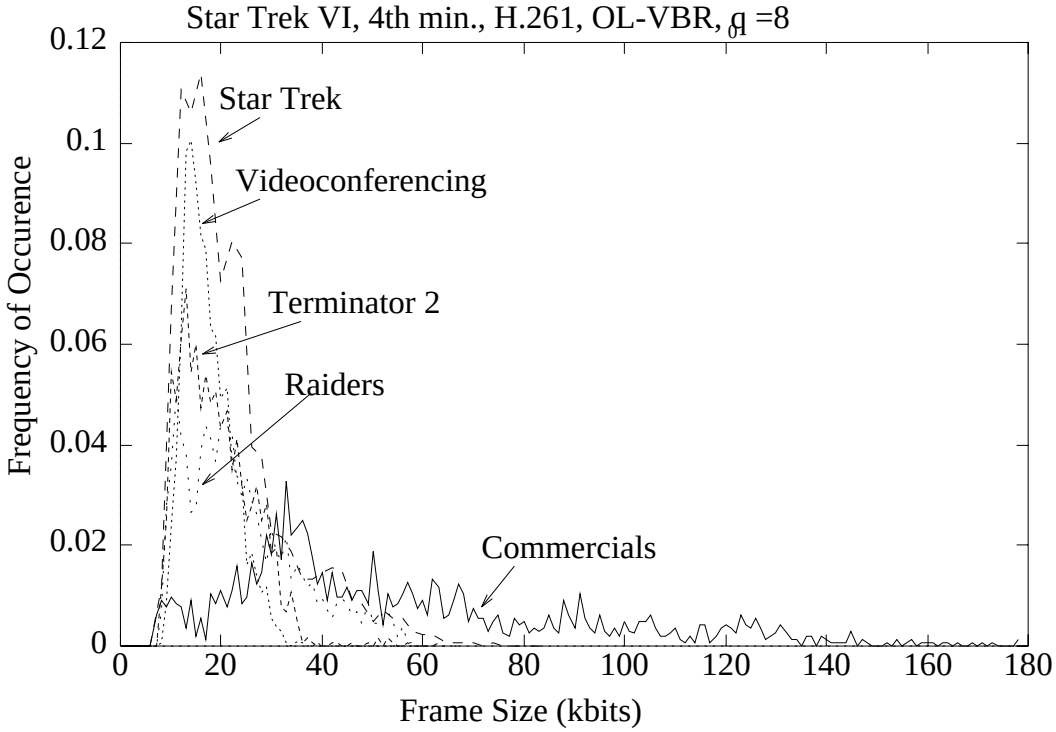


Figure 90: Frame size histogram for various sequences, H.261, OL-VBR, q_0=8.

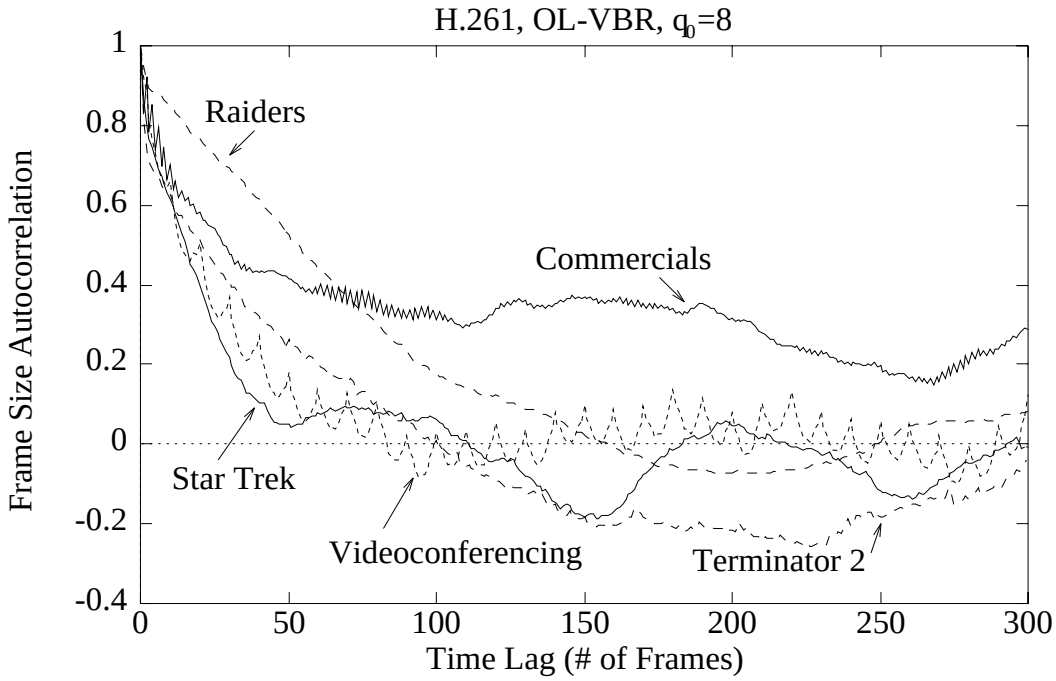


Figure 91: Frame size autocorrelation for various sequences, H.261, OL-VBR, q_0=8.

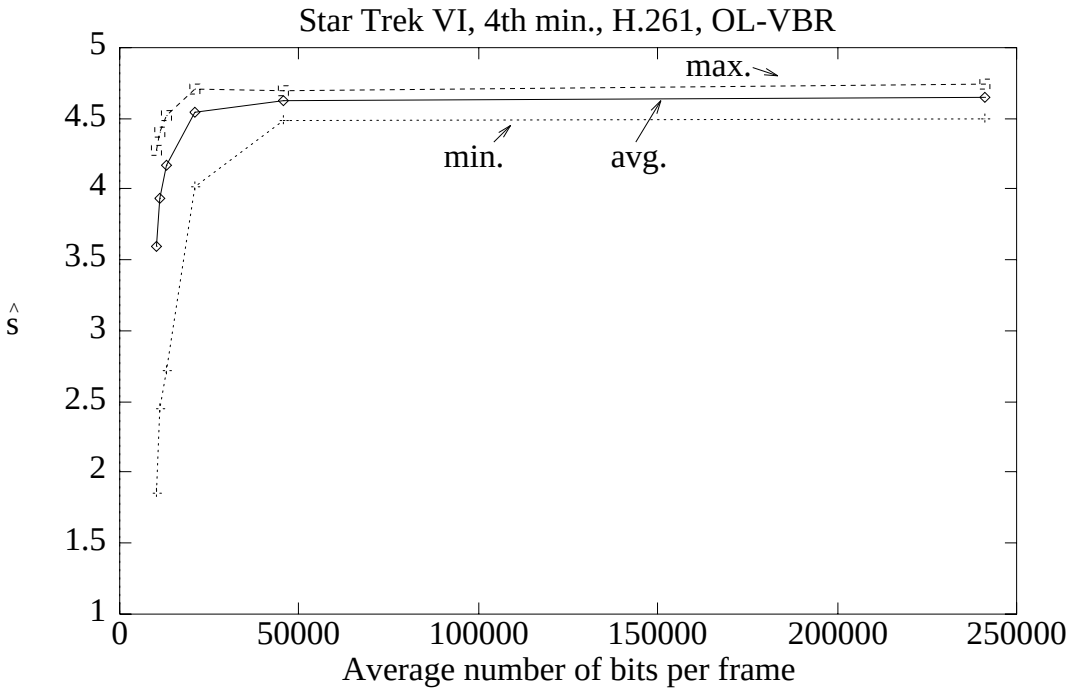


Figure 92: Maximum, average, and minimum $\hat{s}$ versus average frame size for the Star Trek sequence, H.261, OL-VBR.

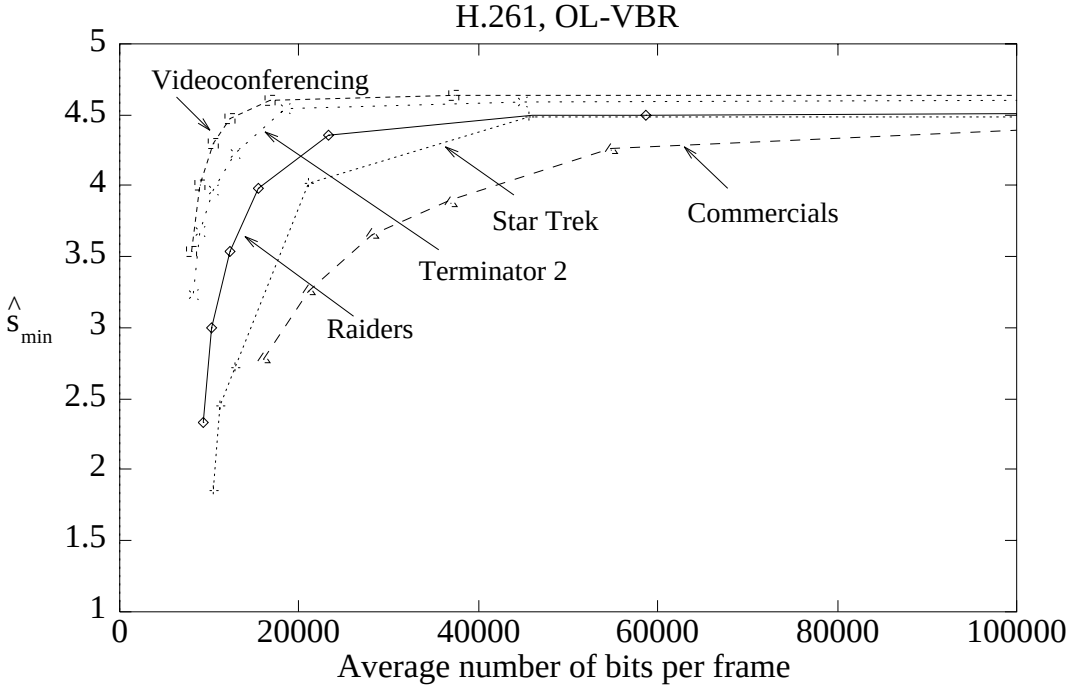


Figure 93: Minimum quality versus average frame size for various sequences, H.261, OL-VBR.

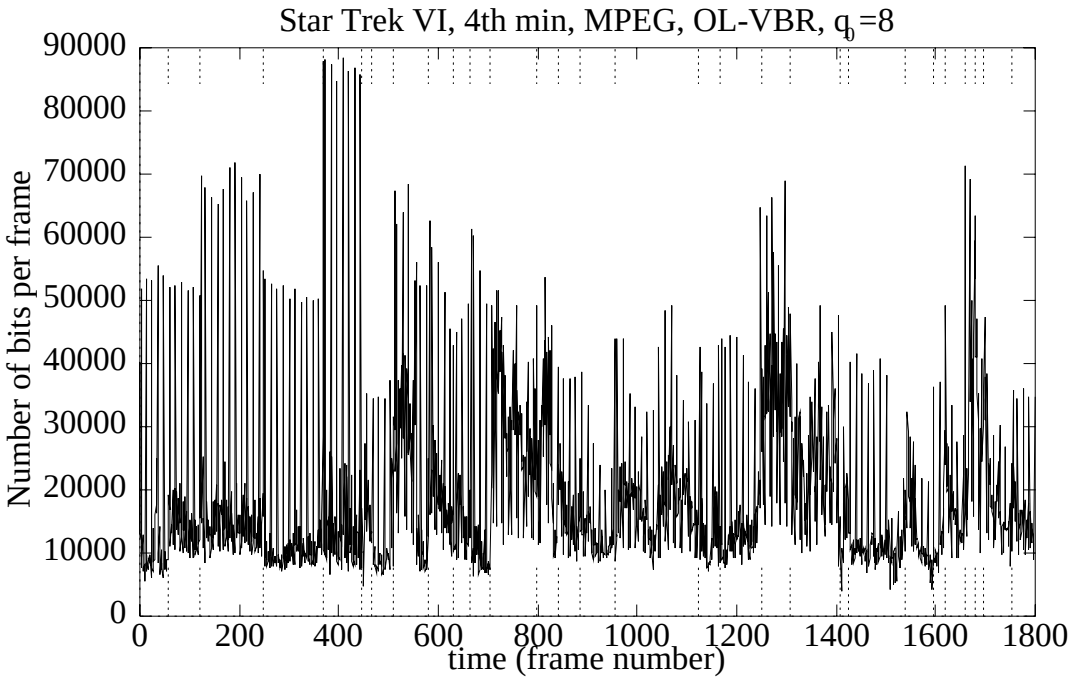


Figure 94: Number of bits per frame versus time for the Star Trek sequence, MPEG, GS1, OL-VBR, q_0=8. (Resulting average frame size = 18.8 kbits.)

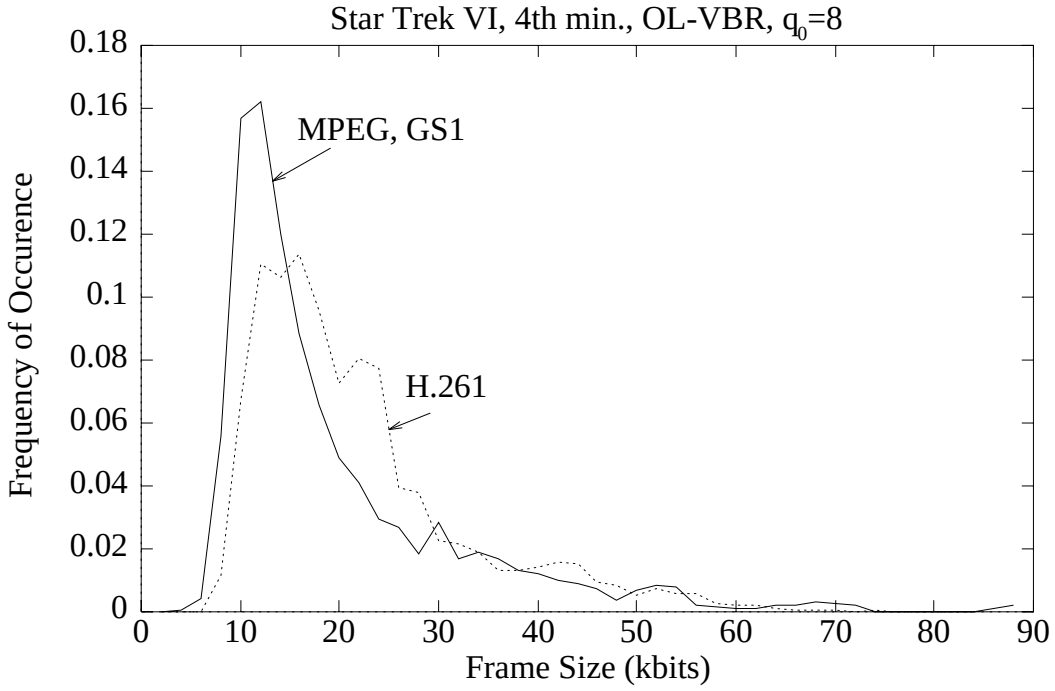


Figure 95: Frame size histogram for the Star Trek sequence, MPEG, GS1, OL-VBR, q_0=8. (The histogram for the corresponding H.261 sequence also shown for comparison.)

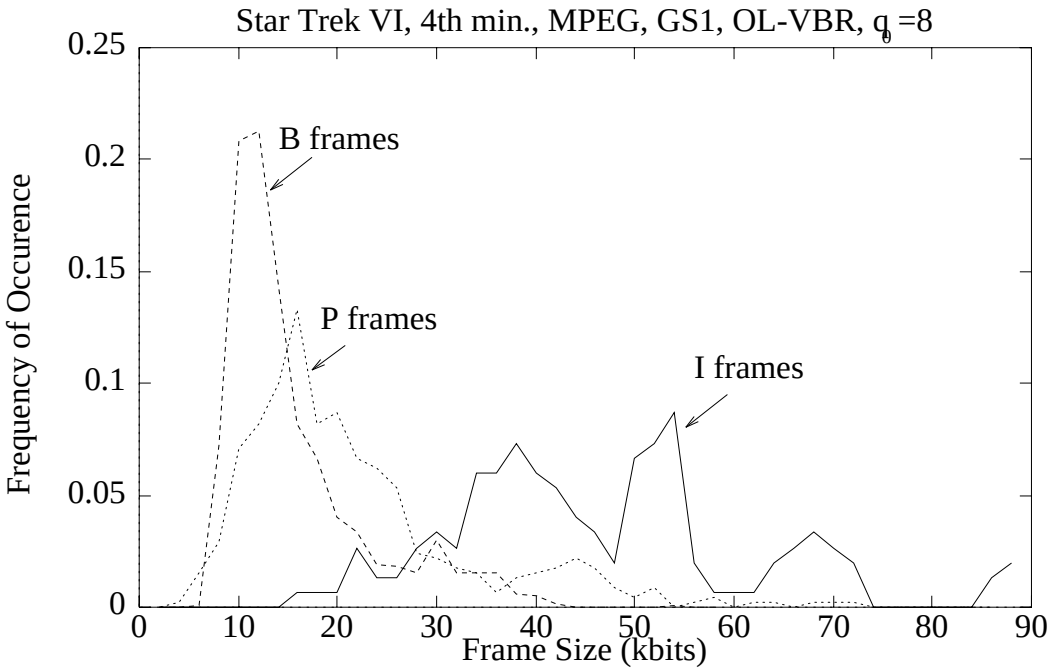


Figure 96: Frame size histogram for I, P, and B frames for the Star Trek sequence, MPEG, GS1, OL-VBR, q_0=8.

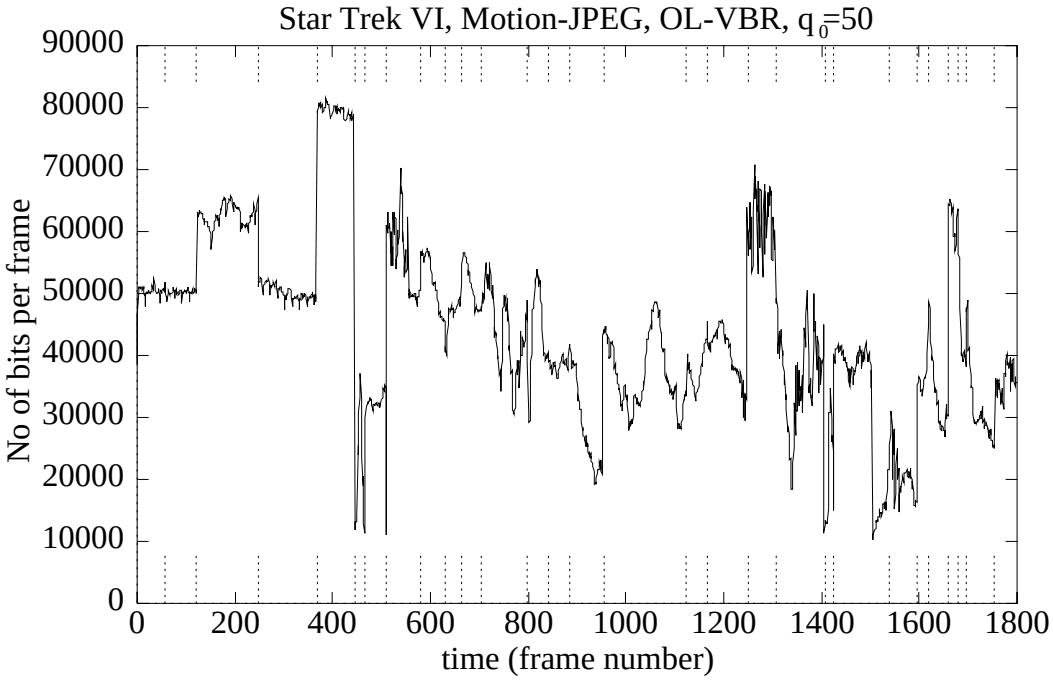


Figure 97: Number of bits per frame versus time for the Star Trek sequence, Motion-JPEG, OL-VBR, q_0=50. (Resulting average frame size = 43.9 kbits.)

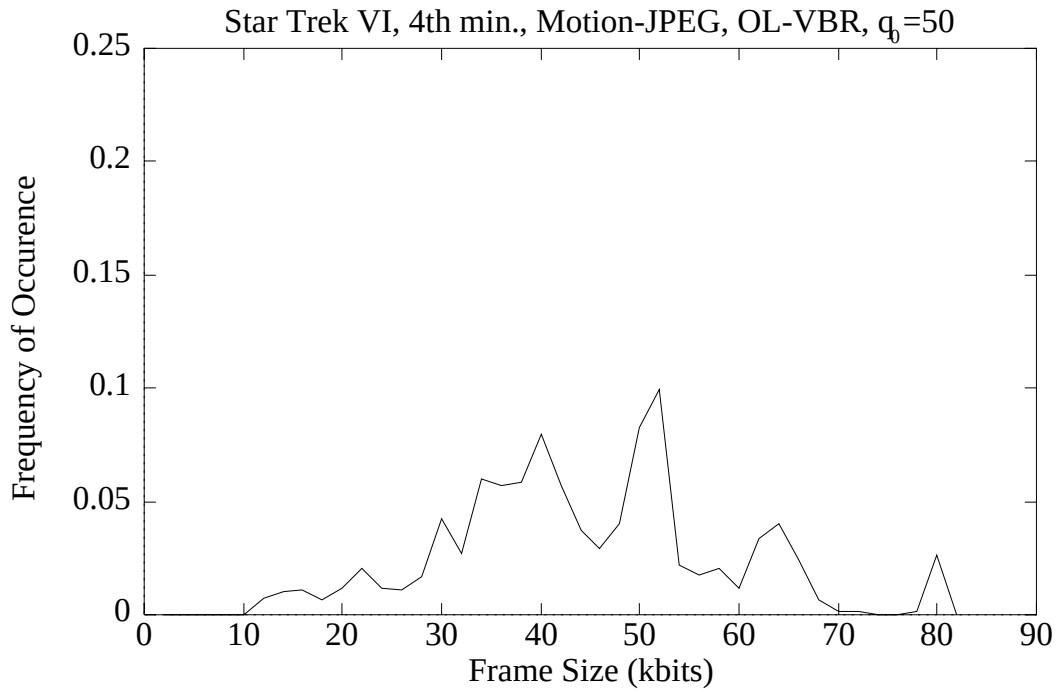


Figure 98: Frame size histogram for the Star Trek sequence, Motion-JPEG, OL-VBR, q_0=50.

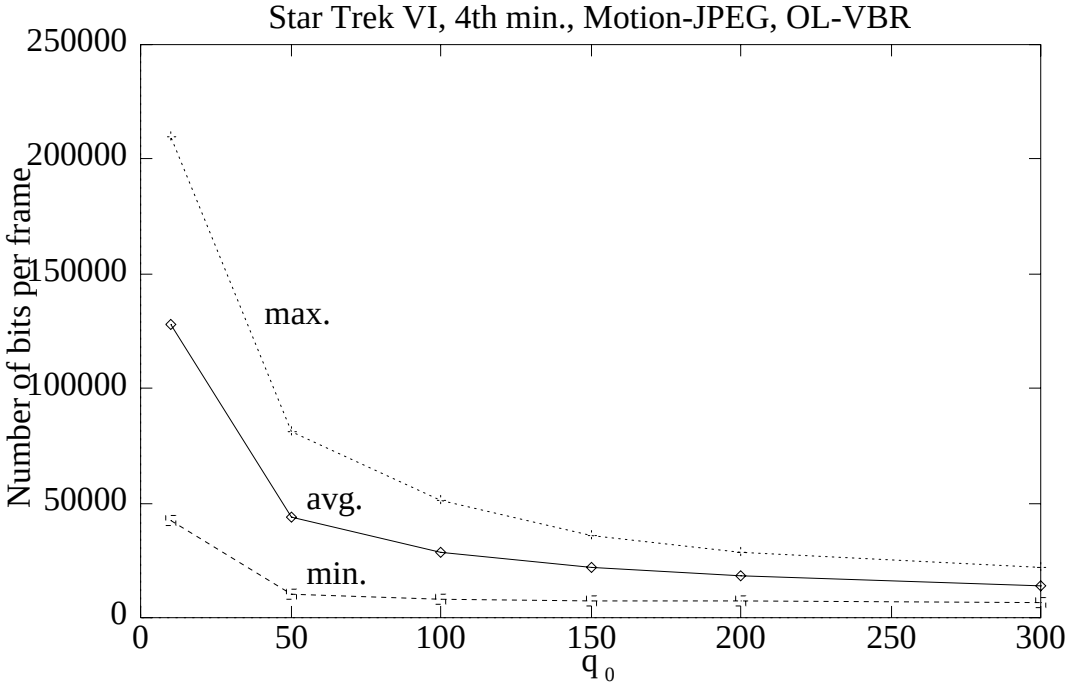


Figure 99: Maximum, average, and minimum frame size versus q_0 for the Star Trek sequence, Motion-JPEG, OL-VBR.

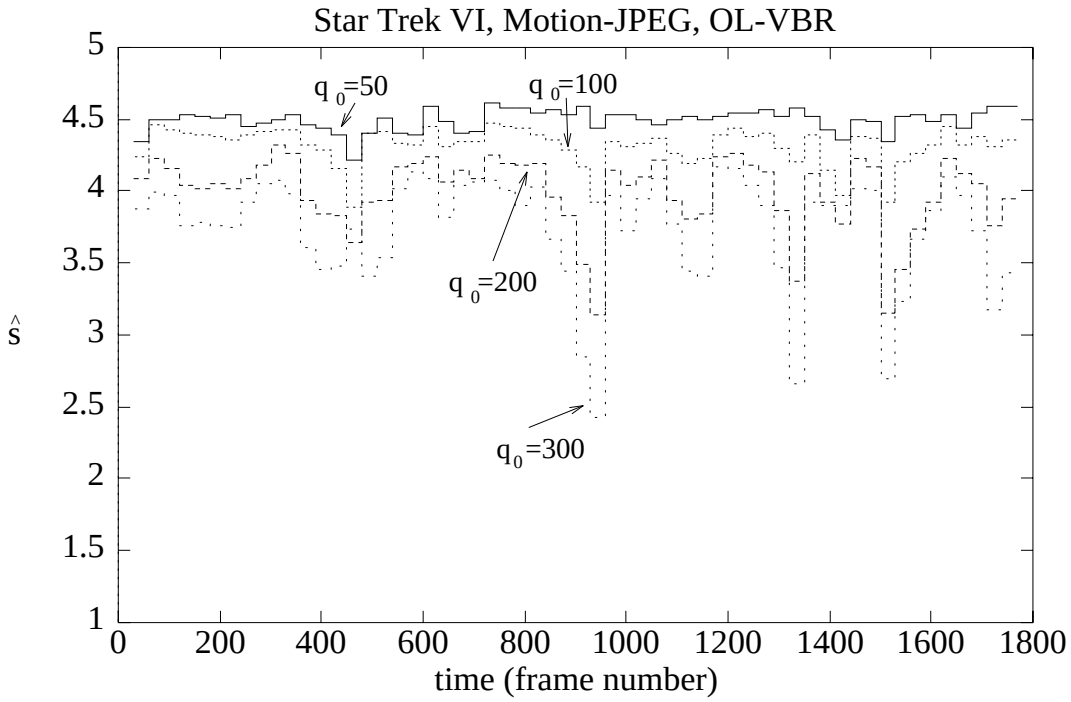


Figure 100: $\hat{s}$ versus time for the Star Trek sequence, Motion-JPEG, OL-VBR, various values of q_0.

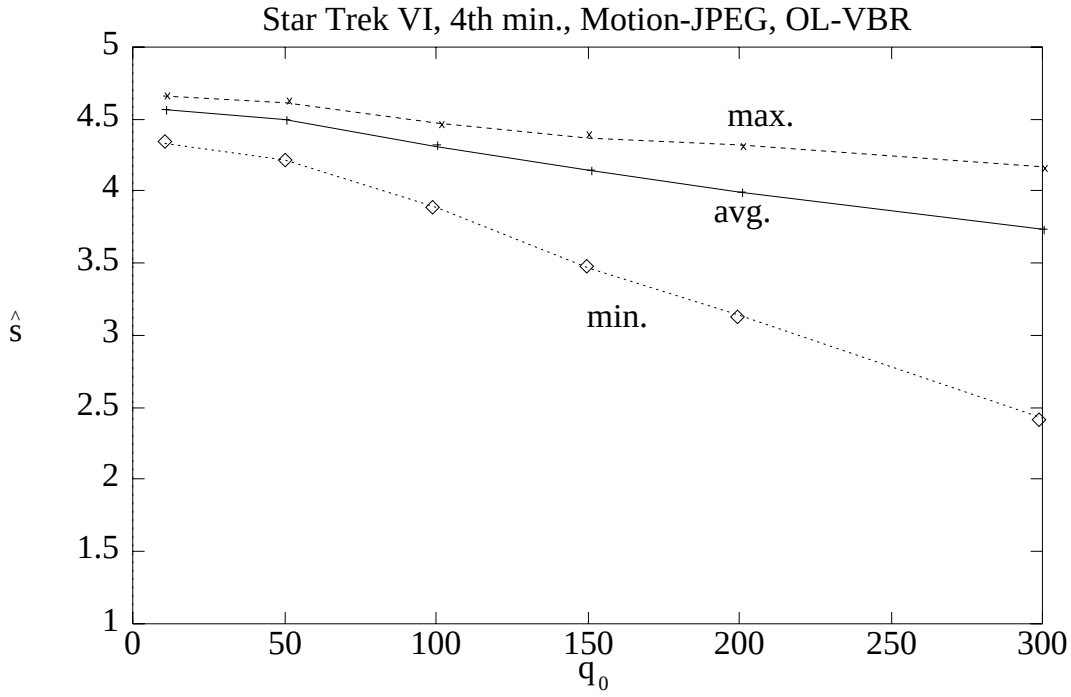


Figure 101: Maximum, average, and minimum $\hat{s}$ versus q_0 for the Star Trek sequence, Motion-JPEG, OL-VBR.

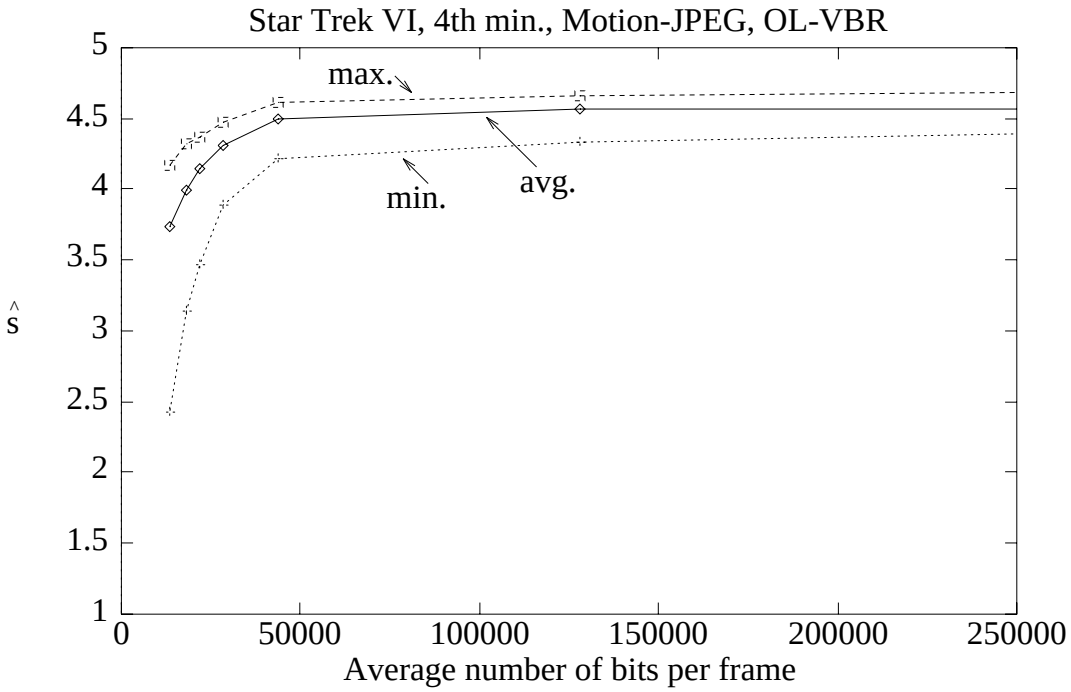


Figure 102: Maximum, average, and minimum $\hat{s}$ versus average frame size for the Star Trek sequence, Motion-JPEG, OL-VBR.

	Frame Size (kbits)			
	Average	Std. Dev.	Maximum	Minimum
Commercials	79.2	22.3	144.3	7.1
Raiders	52.7	12.8	88.0	22.9
Star Trek VI	43.9	14.4	81.5	10.4
Terminator 2	53.7	9.8	77.5	26.3
Videoconferencing	58.6	2.7	69.3	50.4

Table 2: Frame size statistics for all five sequences, Motion-JPEG, OL-VBR, q_0=50.

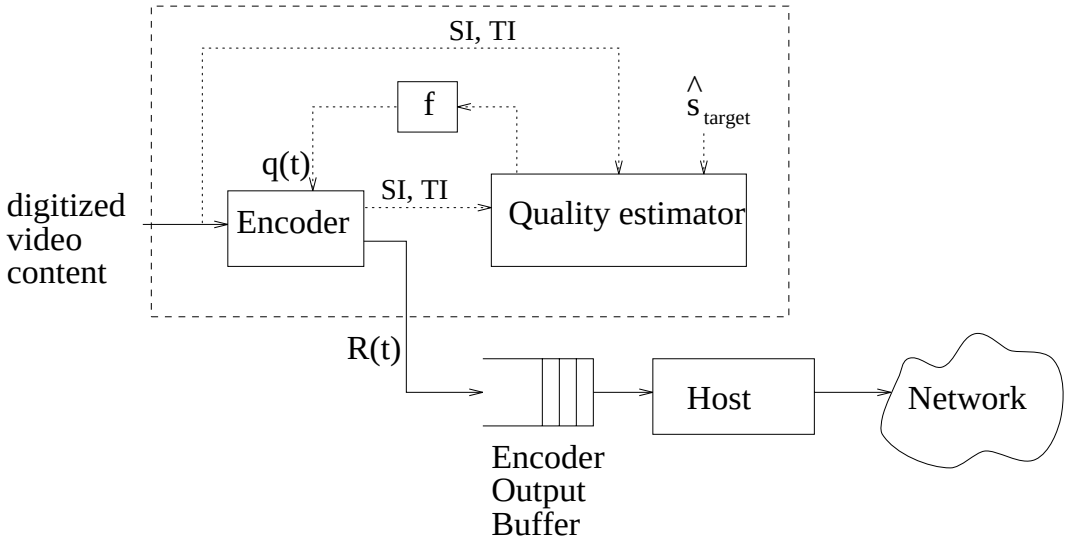


Figure 103: Block diagram of the encoder for Constant Quality VBR encoding

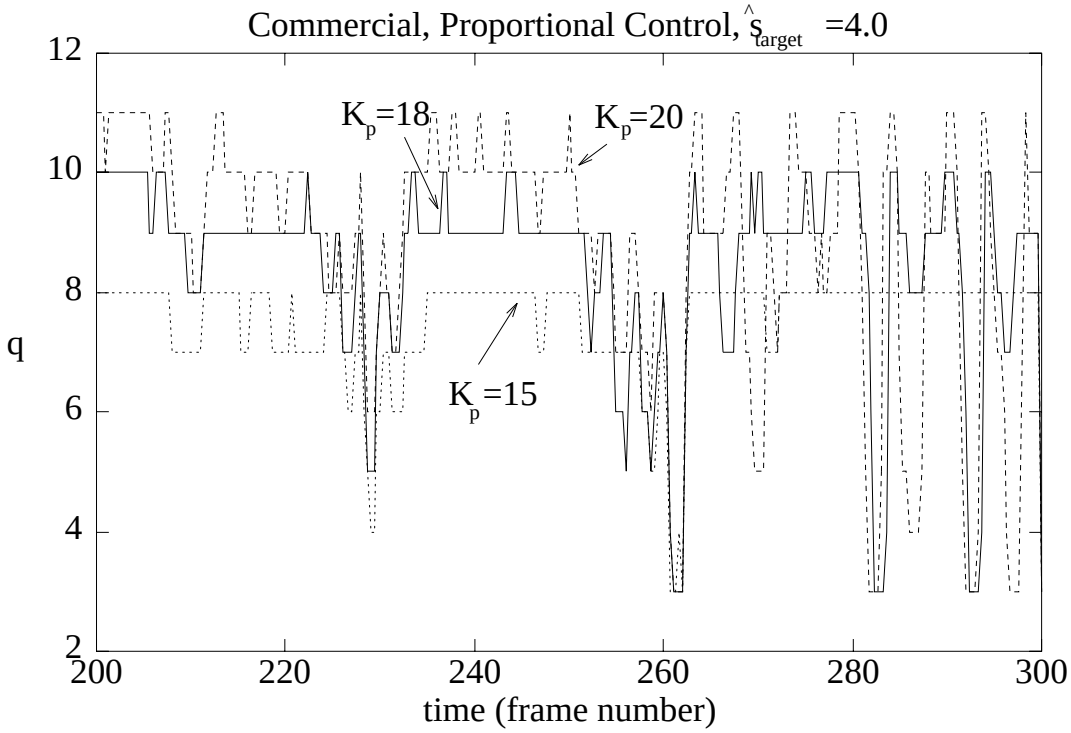


Figure 104: q versus time for the Commercials sequence, proportionally controlled, K_p={15,18,20}.

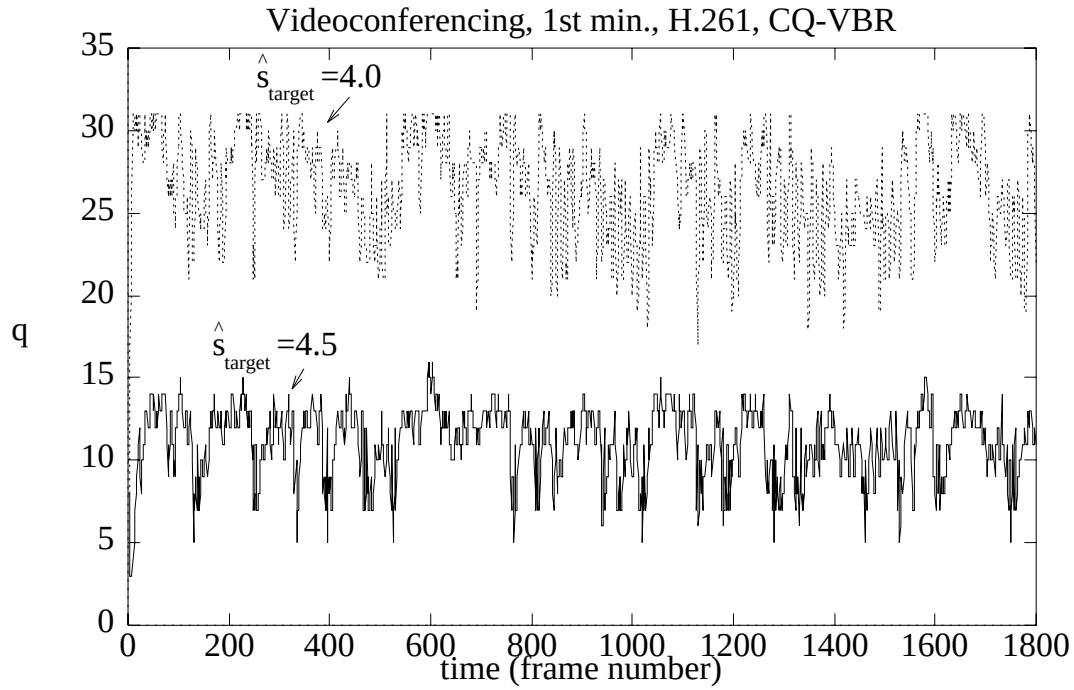


Figure 105: q versus time for the Videoconferencing sequence, CQ-VBR, $\hat{s}_{target}$={4.0,4.5}.

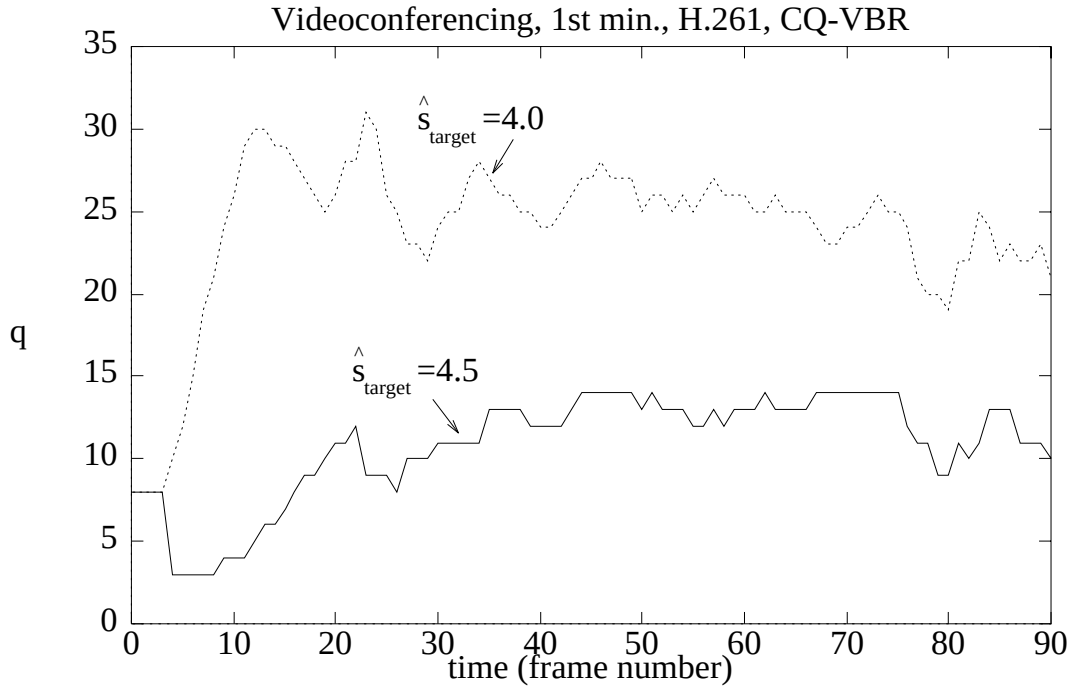


Figure 106: q versus time for the first three seconds of the Videoconferencing sequence, CQ-VBR, $\hat{s}_{target}$={4.0,4.5}.

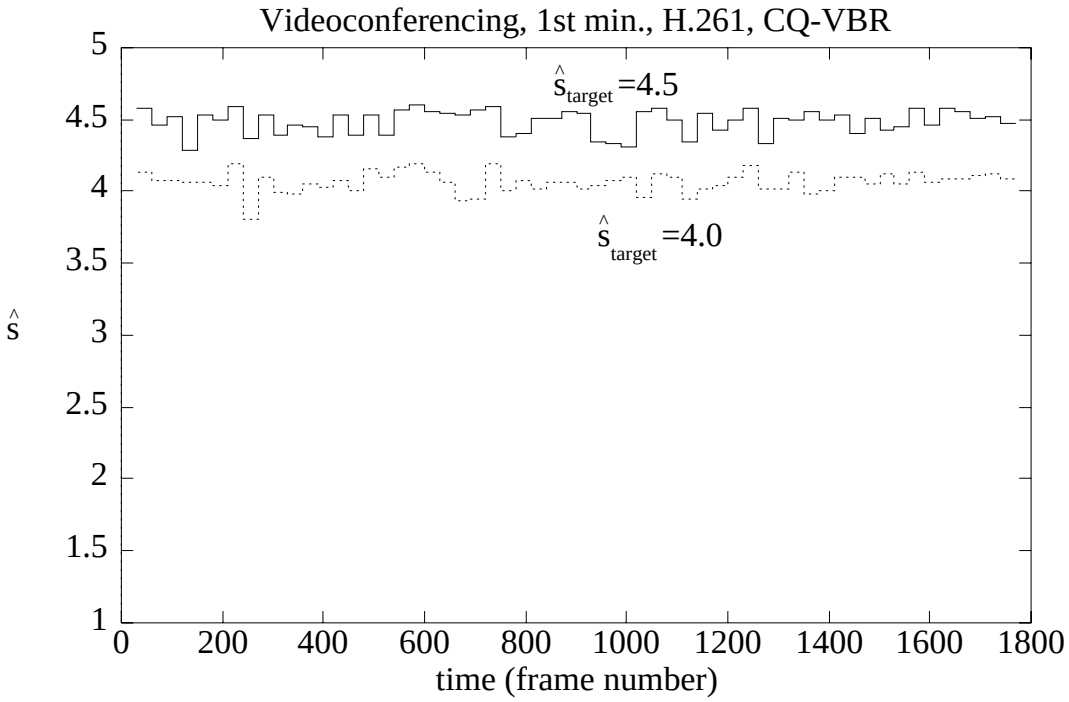
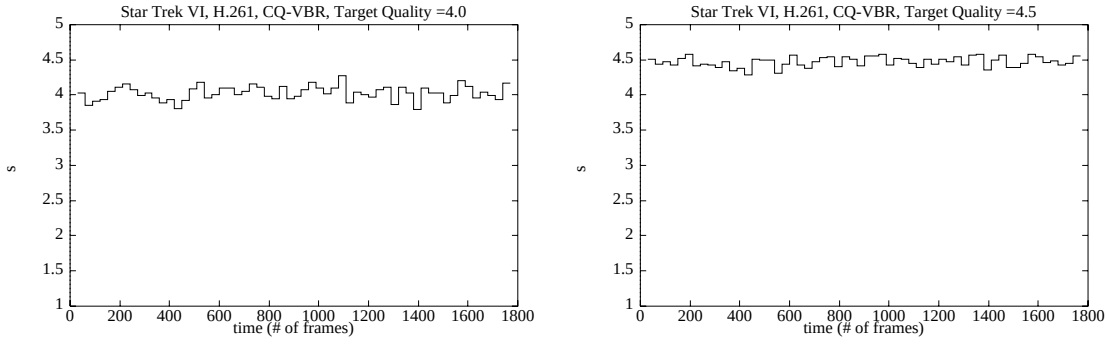
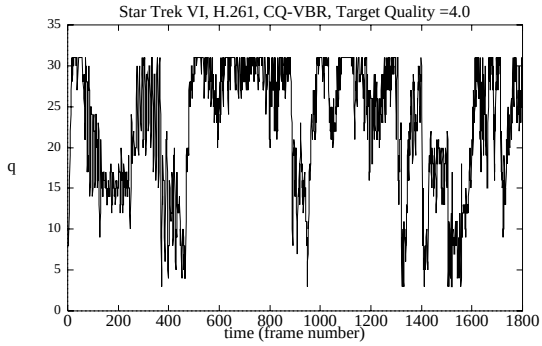


Figure 107: $\hat{s}$ versus time for the Videoconferencing sequence, CQ-VBR, $\hat{s}_{target}$={4.0,4.5}.

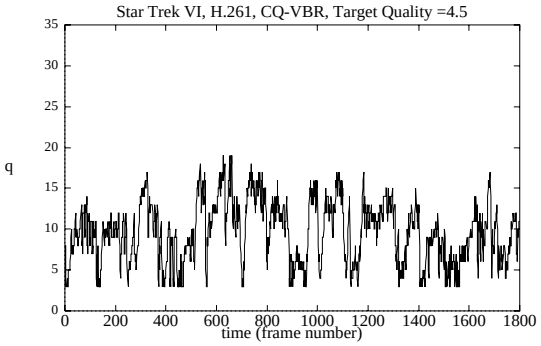


(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 108: $\hat{s}$ versus time for Star Trek VI, H.261, CQ-VBR

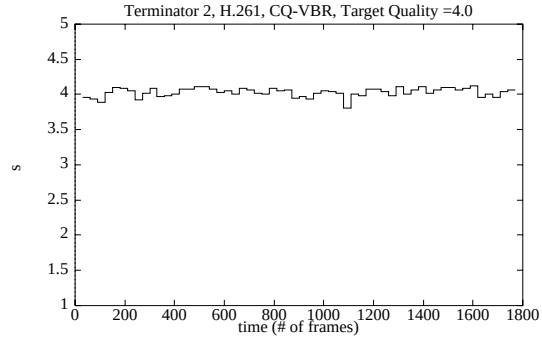


(a) $\hat{s}_{target}$=4.0

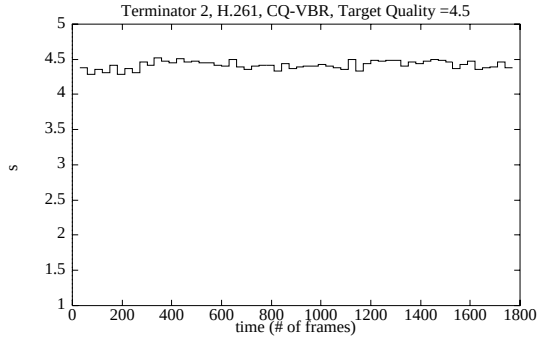


(b) $\hat{s}_{target}$=4.5

Figure 109: q versus time for Star Trek VI, H.261, CQ-VBR

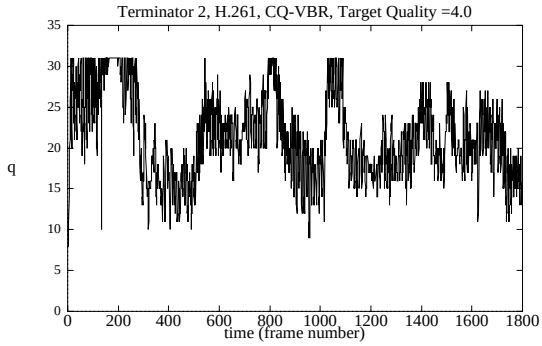


(a) $\hat{s}_{target}$=4.0

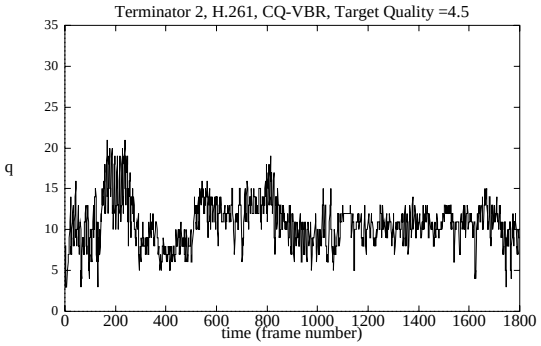


(b) $\hat{s}_{target}$=4.5

Figure 110: $\hat{s}$ versus time for Terminator 2, H.261, CQ-VBR

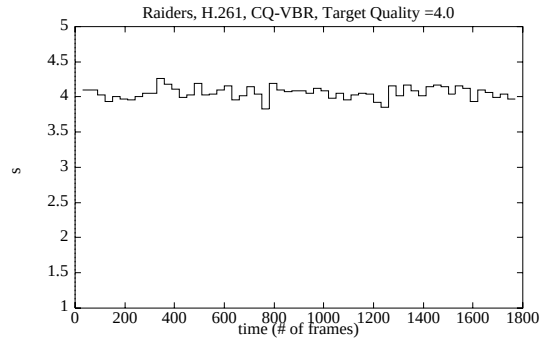


(a) $\hat{s}_{target}$=4.0

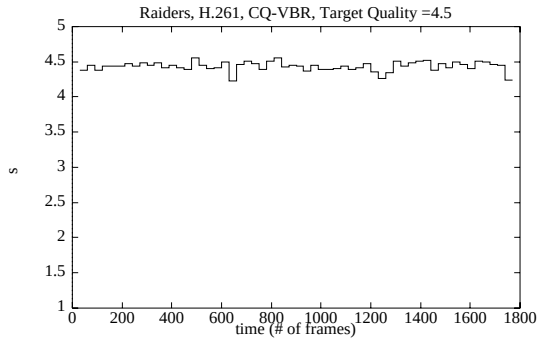


(b) $\hat{s}_{target}$=4.5

Figure 111: q versus time for Terminator 2, H.261, CQ-VBR

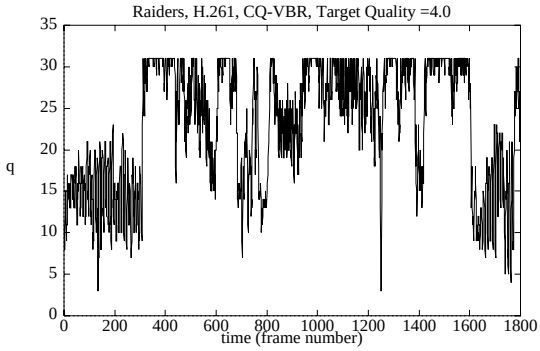


(a) $\hat{s}_{target}$=4.0

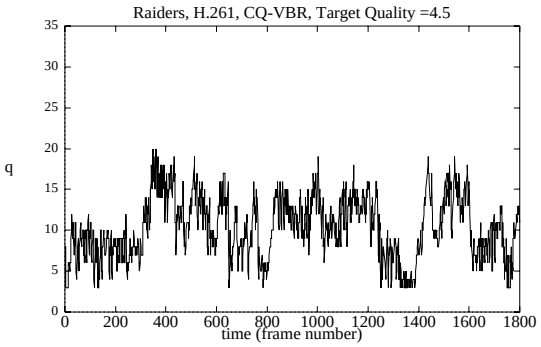


(b) $\hat{s}_{target}$=4.5

Figure 112: $\hat{s}$ versus time for Raiders, H.261, CQ-VBR

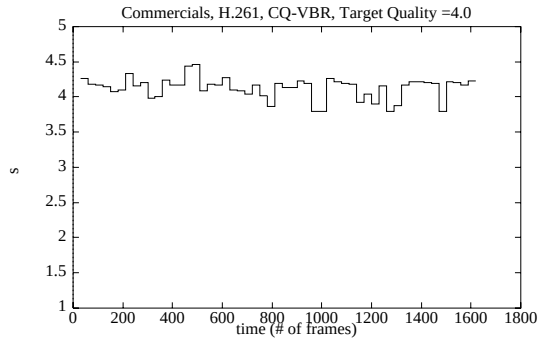


(a) $\hat{s}_{target}$=4.0

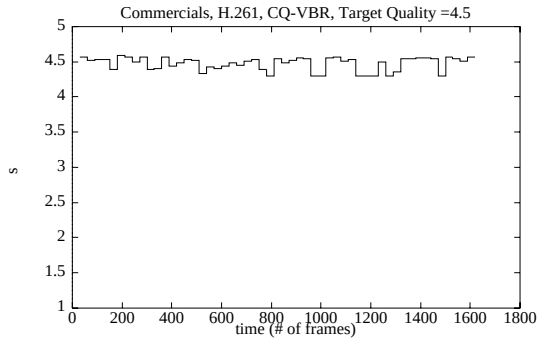


(b) $\hat{s}_{target}$=4.5

Figure 113: q versus time for Raiders, H.261, CQ-VBR

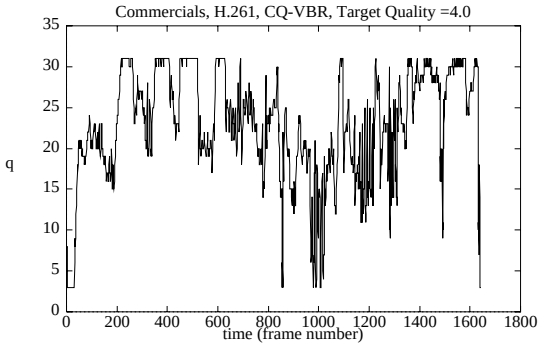


(a) $\hat{s}_{target}$=4.0

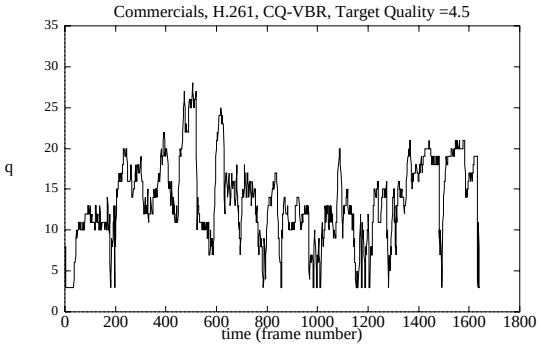


(b) $\hat{s}_{target}$=4.5

Figure 114: $\hat{s}$ versus time for Commercials, H.261, CQ-VBR



(a) $\hat{s}_{target}$=4.0



(b) $\hat{s}_{target}$=4.5

Figure 115: q versus time for Commercials, H.261, CQ-VBR

	$\hat{s}$			
	Average	Std. Dev.	Minimum	Maximum
Commercials	4.02	0.11	3.77	4.29
Raiders	4.05	0.08	3.84	4.27
Star Trek VI	4.03	0.10	3.80	4.28
Terminator 2	4.03	0.06	3.81	4.12
Videoconferencing	4.06	0.07	3.80	4.19

(a) $\hat{s}_{target}$=4.0

	$\hat{s}$			
	Average	Std. Dev.	Minimum	Maximum
Commercials	4.52	0.08	4.33	4.73
Raiders	4.51	0.05	4.37	4.62
Star Trek VI	4.51	0.08	4.28	4.64
Terminator 2	4.48	0.06	4.35	4.58
Videoconferencing	4.49	0.08	4.28	4.60

(b) $\hat{s}_{target}$=4.5

Table 3: Quality statistics for all five sequences, H.261, CQ-VBR, $\hat{s}_{target}$={4.0,4.5}.

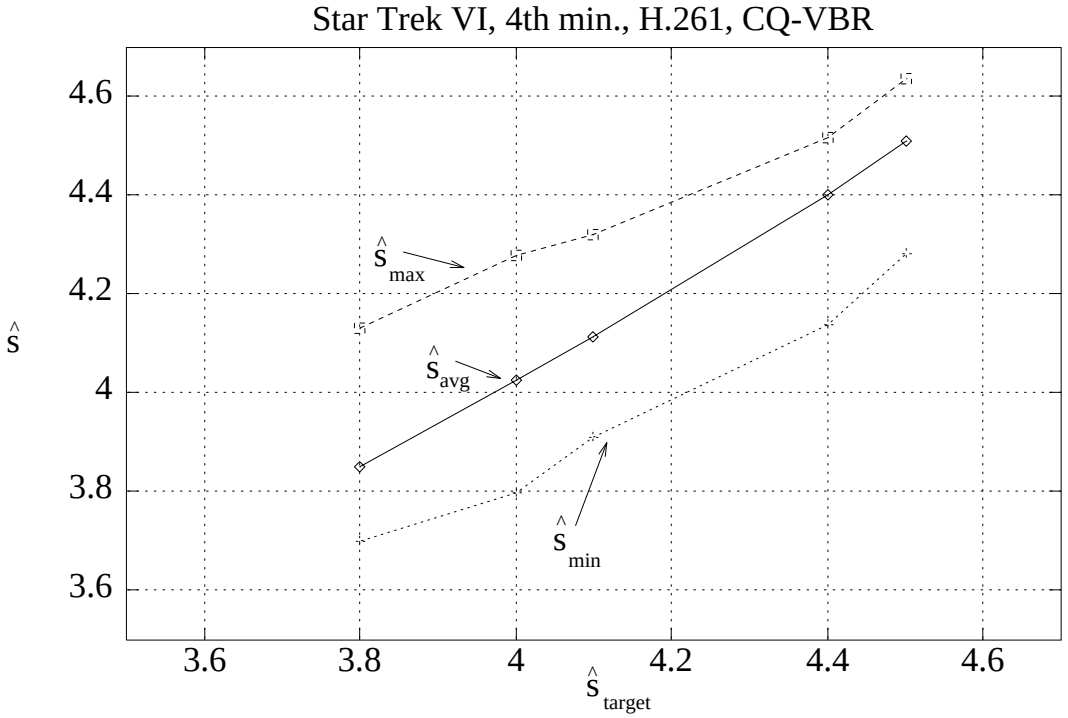
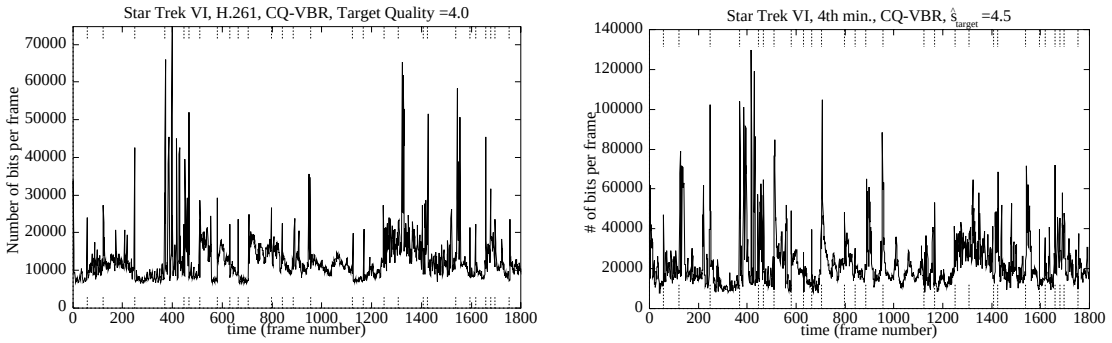
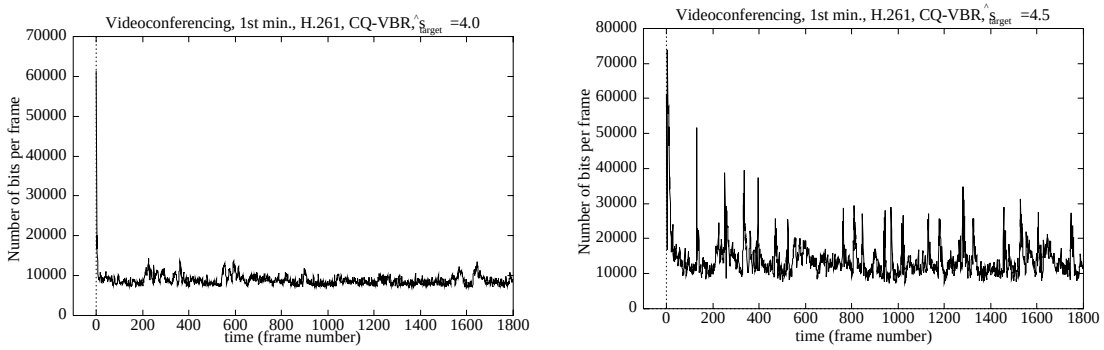


Figure 116: Average, minimum, and maximum $\hat{s}$ versus $\hat{s}_{target}$ for the Star Trek sequence, CQ-VBR.



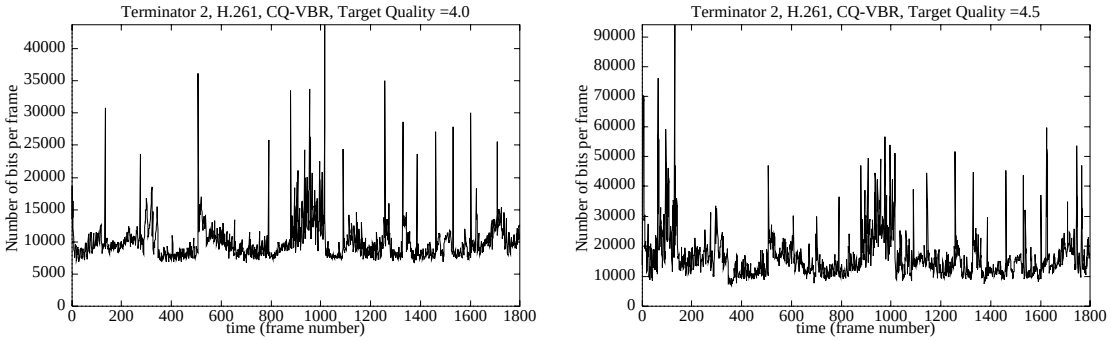
(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 117: Number of bits per frame for Star Trek VI, H.261, CQ-VBR



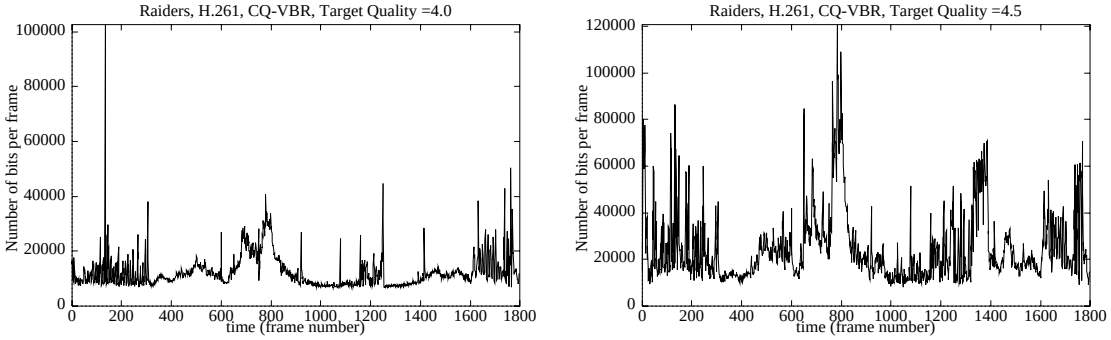
(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 118: Number of bits per frame for Videoconferencing, H.261, CQ-VBR.



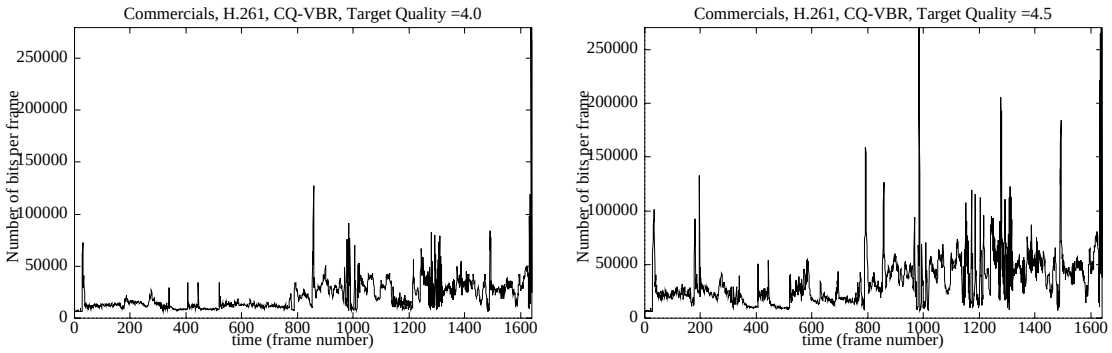
(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 119: Number of bits per frame for Terminator 2, H.261, CQ-VBR



(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 120: Number of bits per frame for Raiders, H.261, CQ-VBR



(a) $\hat{s}_{target}$=4.0 (b) $\hat{s}_{target}$=4.5

Figure 121: Number of bits per frame for Commercials, H.261, CQ-VBR

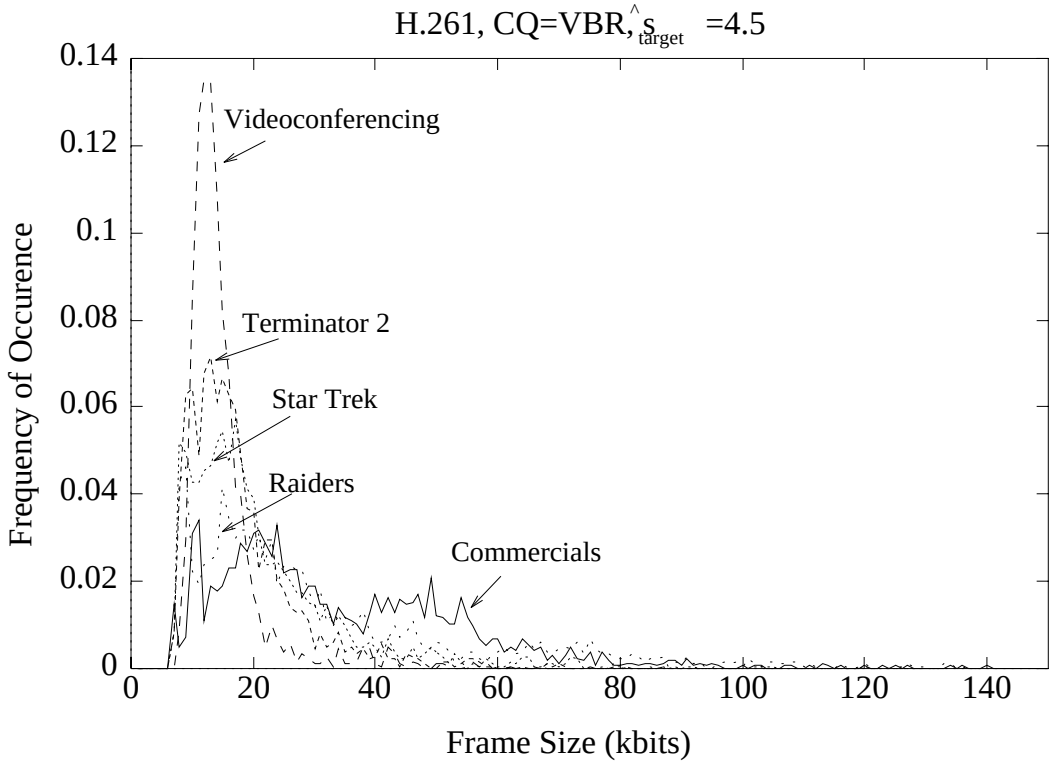


Figure 122: Frame size histograms for all five sequences, H.261, CQ-VBR, $\hat{s}_{target}$=4.5.

	Frame Size (kbits)			
	Average	Std. Dev.	Maximum	Minimum
Commercials	20.3	11.6	186.5	6.1
Raiders	11.2	5.8	102.5	6.2
Star Trek VI	12.0	6.7	98.2	6.4
Terminator 2	9.5	3.2	43.7	6.3
Videoconferencing	8.9	1.1	14.3	6.9

(a) $\hat{s}_{target}$= 4.0

	Frame Size (kbits)			
	Average	Std. Dev.	Maximum	Minimum
Commercials	35.0	25.7	270.6	6.1
Raiders	31.8	24.6	138.9	6.5
Star Trek VI	20.9	13.8	130.0	6.7
Terminator 2	17.6	10.0	93.8	6.6
Videoconferencing	14.7	7.2	52.7	7.2

(b) $\hat{s}_{target}$= 4.5

Table 4: Frame size statistics for all five sequences, H.261, CQ-VBR, $\hat{s}_{target}$={4.0,4.5}.

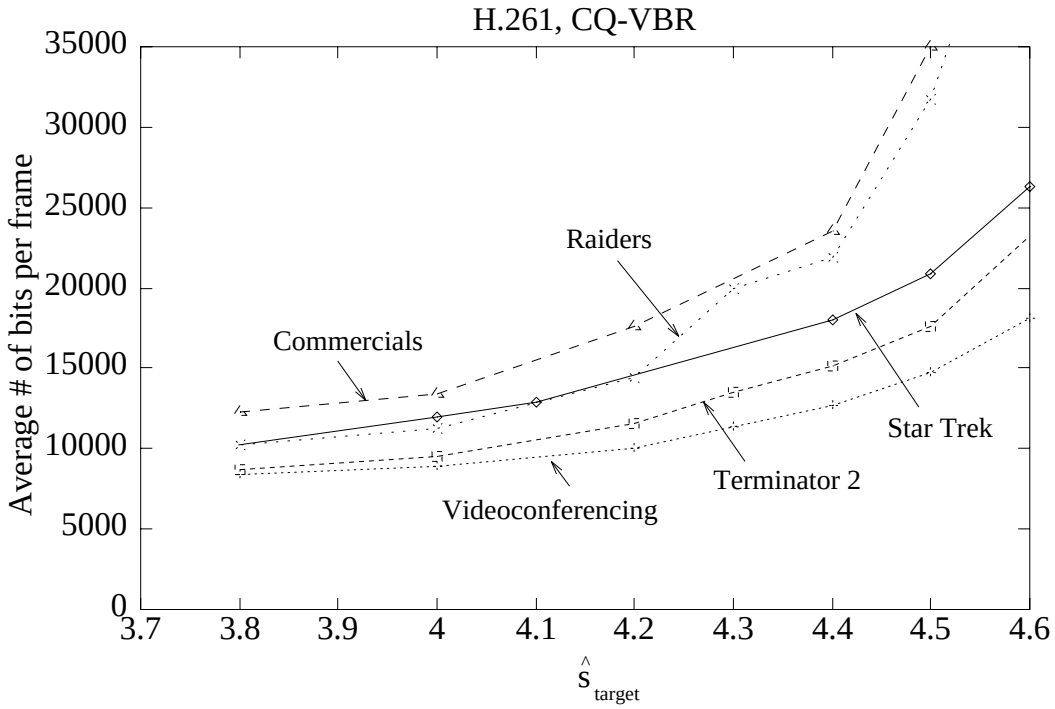


Figure 123: Frame size versus $\hat{s}_{target}$ for various sequences, H.261, CQ-VBR.

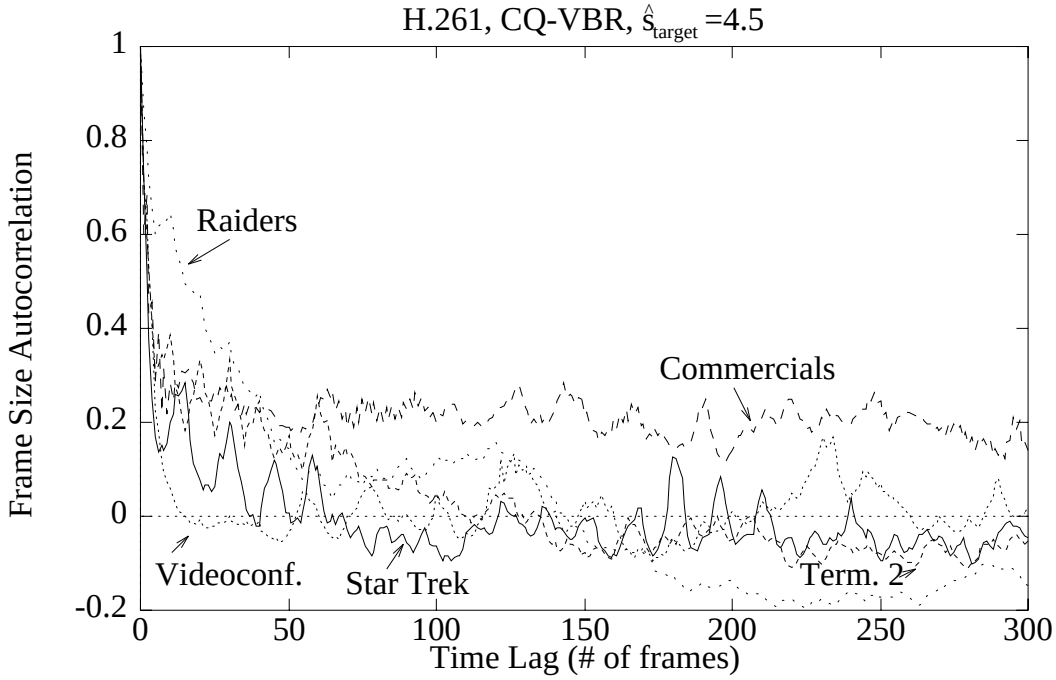


Figure 124: Frame size autocorrelation for various sequences, H.261, CQ-VBR, $\hat{s}_{target}$=4.5.

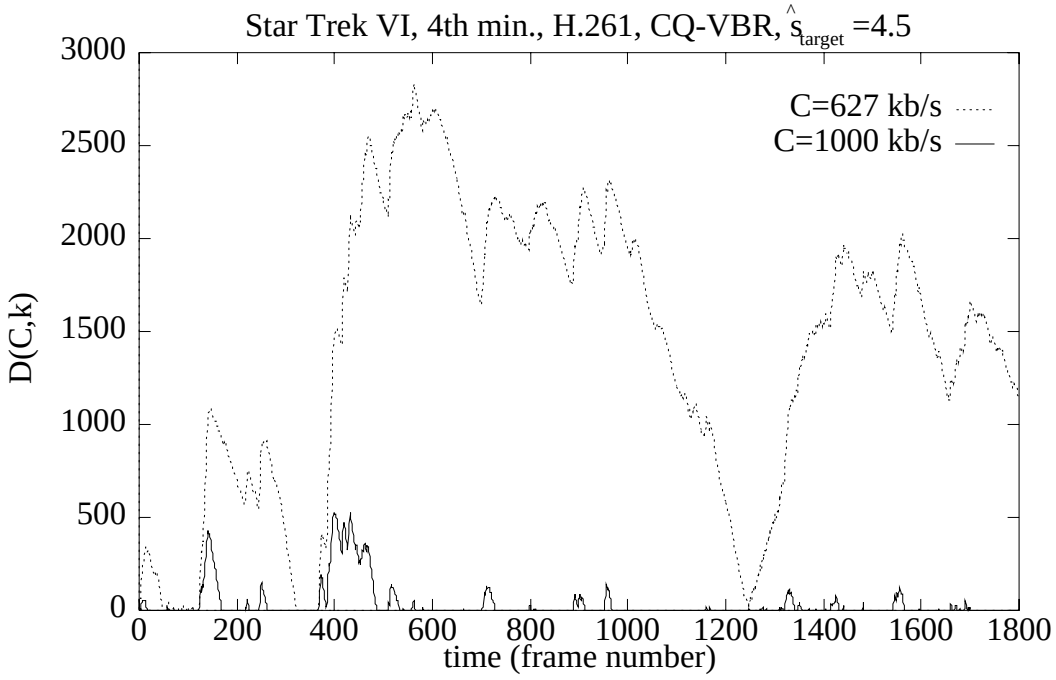


Figure 125: $D(C,k)$ versus time for the Star Trek sequence, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, C={627,1000} kb/s. (Average rate of the sequence is 627 kb/s.)

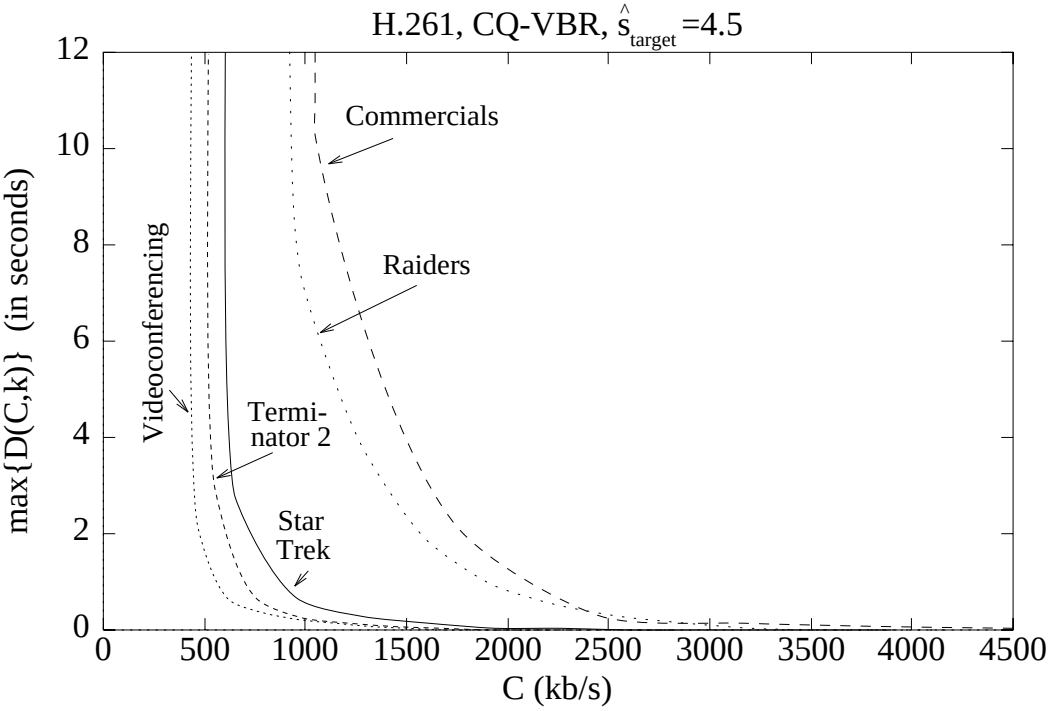
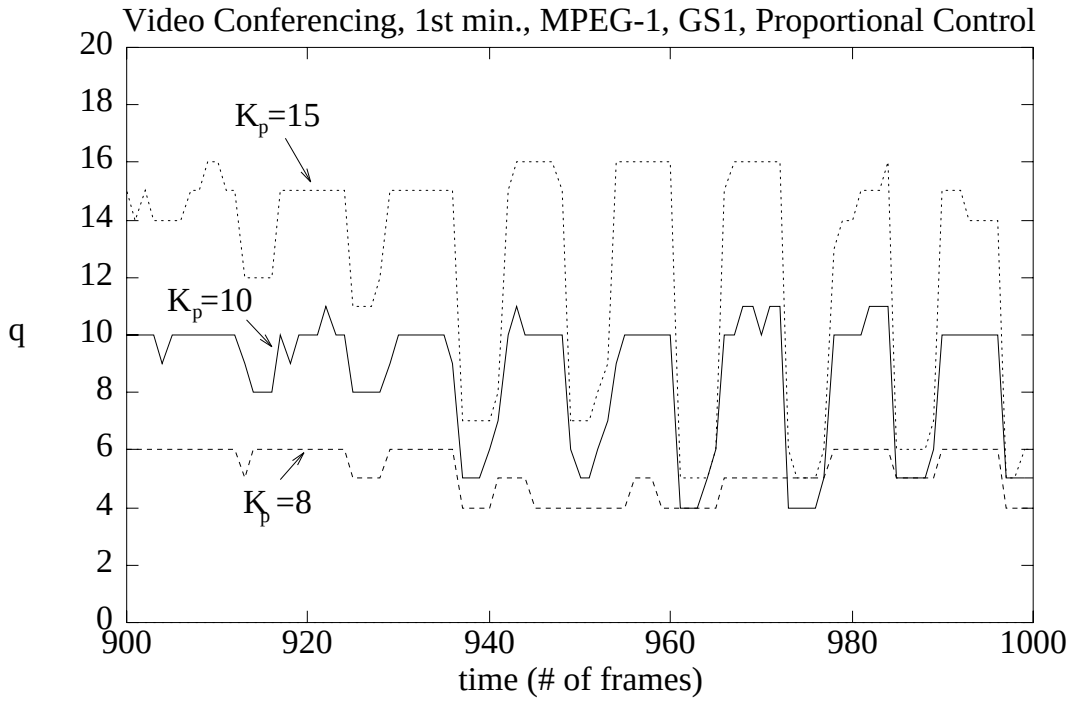
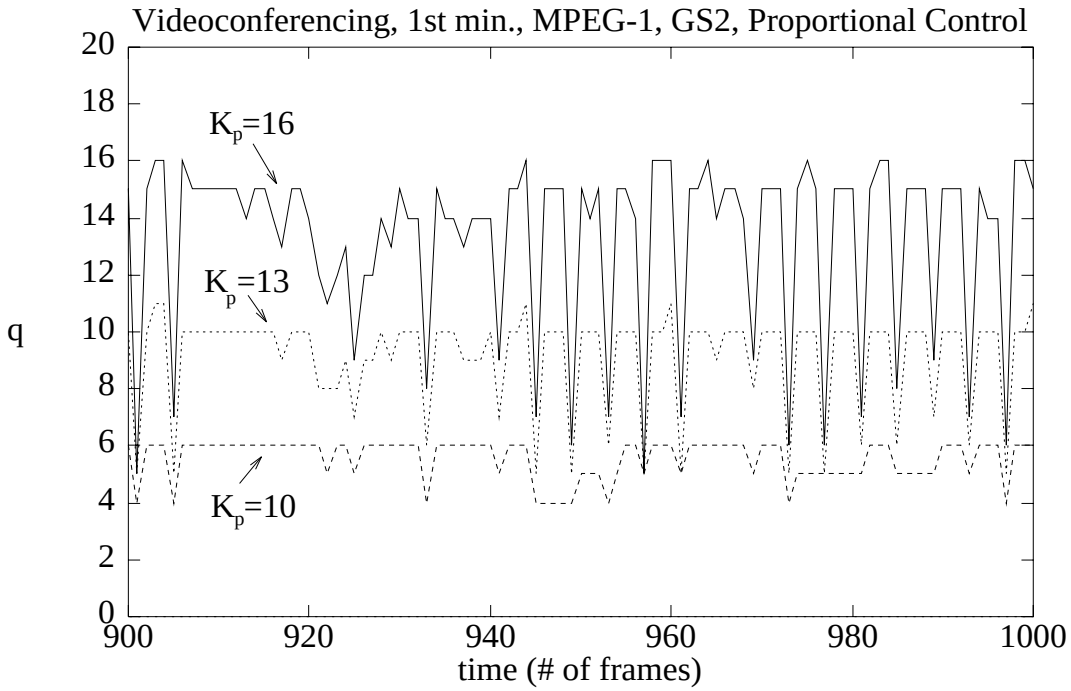


Figure 126: $\max_k\{D(C,k)\}$ versus C for various sequences, H.261, CQ-VBR, $\hat{s}_{target}$=4.5.



(a) GS1



(b) GS2

Figure 127: q versus time for proportional control, Videoconferencing sequence, $\hat{s}_{target}$=4.0, GS1 and GS2.

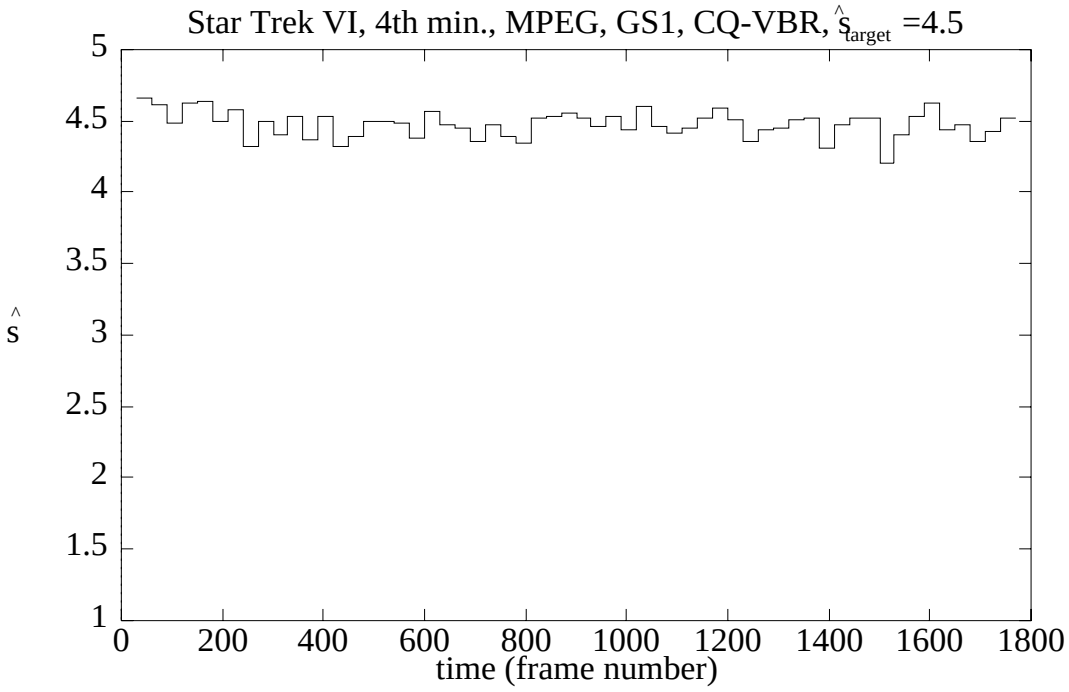


Figure 128: $\hat{s}$ versus time for Star Trek, MPEG, GS1, CQ-VBR, $\hat{s}_{target}$=4.5.

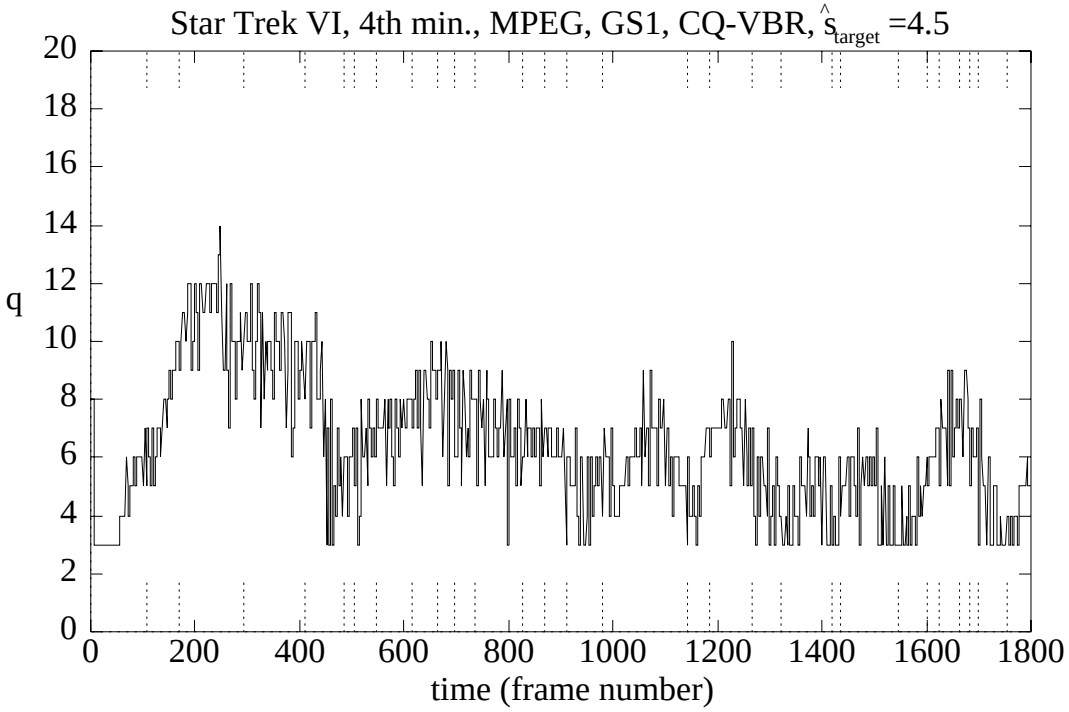


Figure 129: q versus time for Star Trek, MPEG, GS1, CQ-VBR, $\hat{s}_{target}$=4.5.

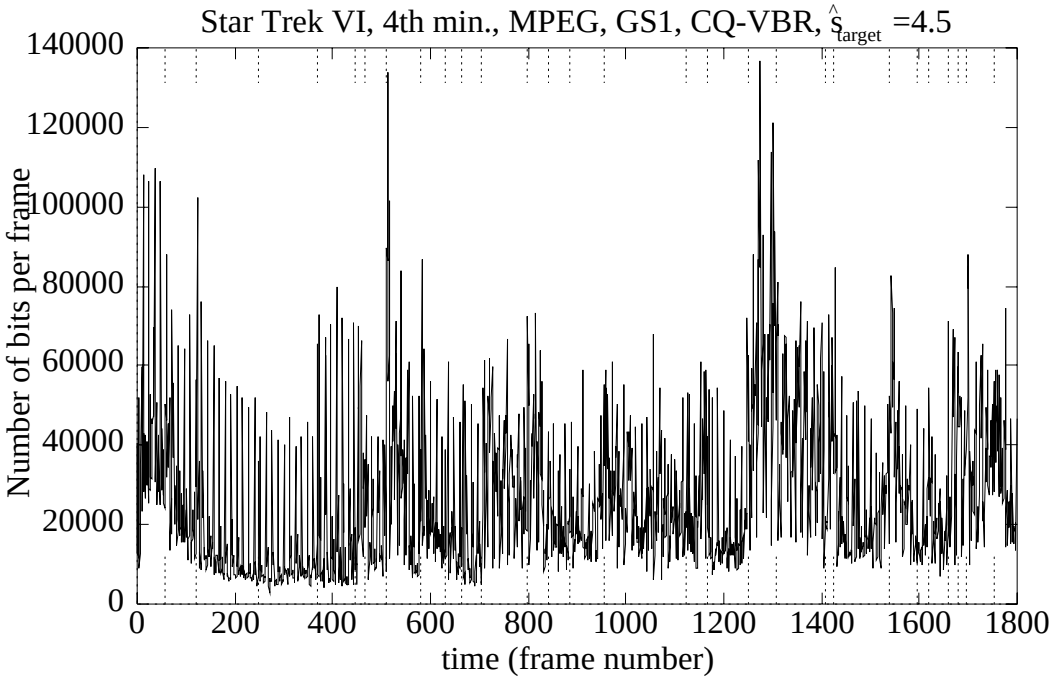


Figure 130: Number of bits per frame versus time for Star Trek, MPEG, GS1, CQ-VBR, $\hat{s}_{target}$=4.5. (Average number of bits per frame = 24.5 kbits.)

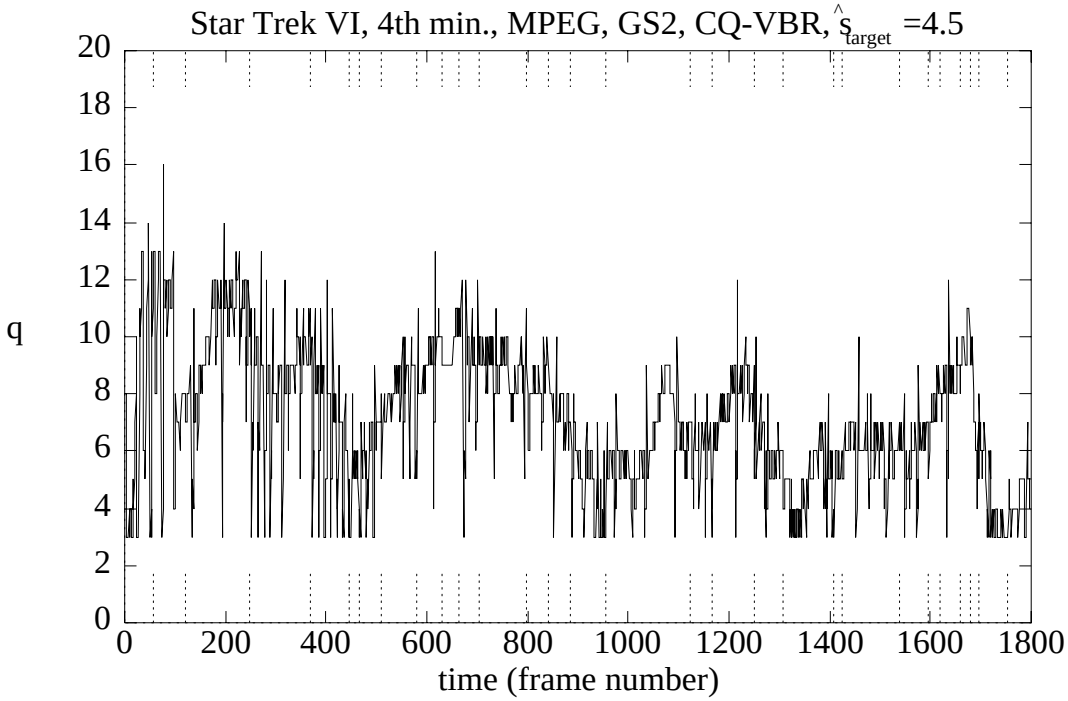


Figure 131: q versus time for Star Trek, MPEG, GS2, CQ-VBR, $\hat{s}_{target}$=4.5.

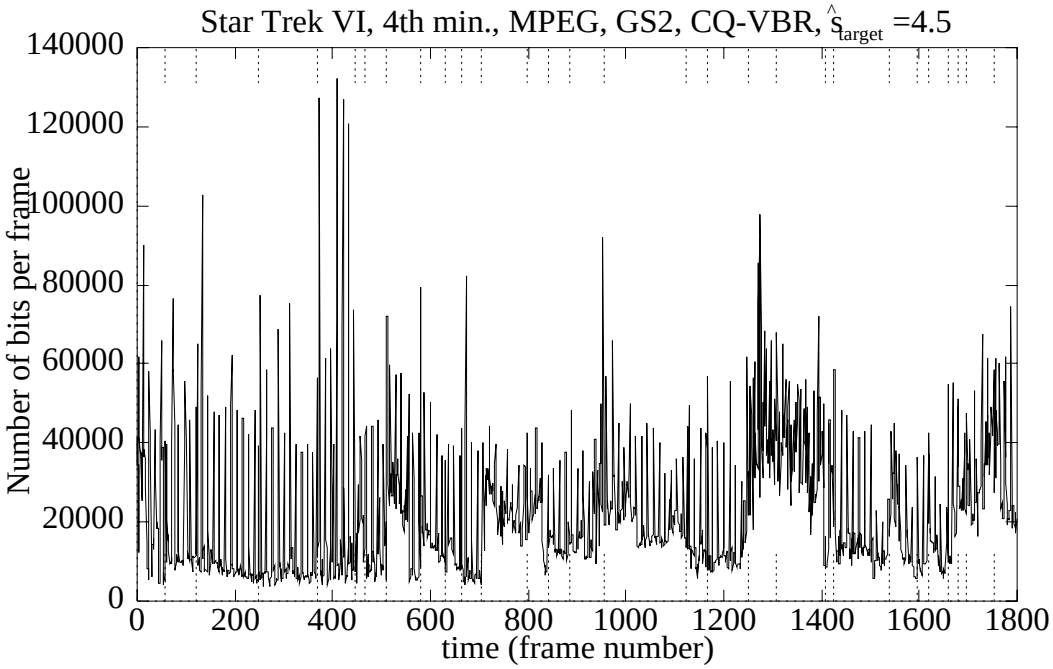


Figure 132: Number of bits per frame versus time for Star Trek, MPEG, GS2, CQ-VBR, $\hat{s}_{target}$=4.5. (Average number of bits per frame = 19.8 kbits.)

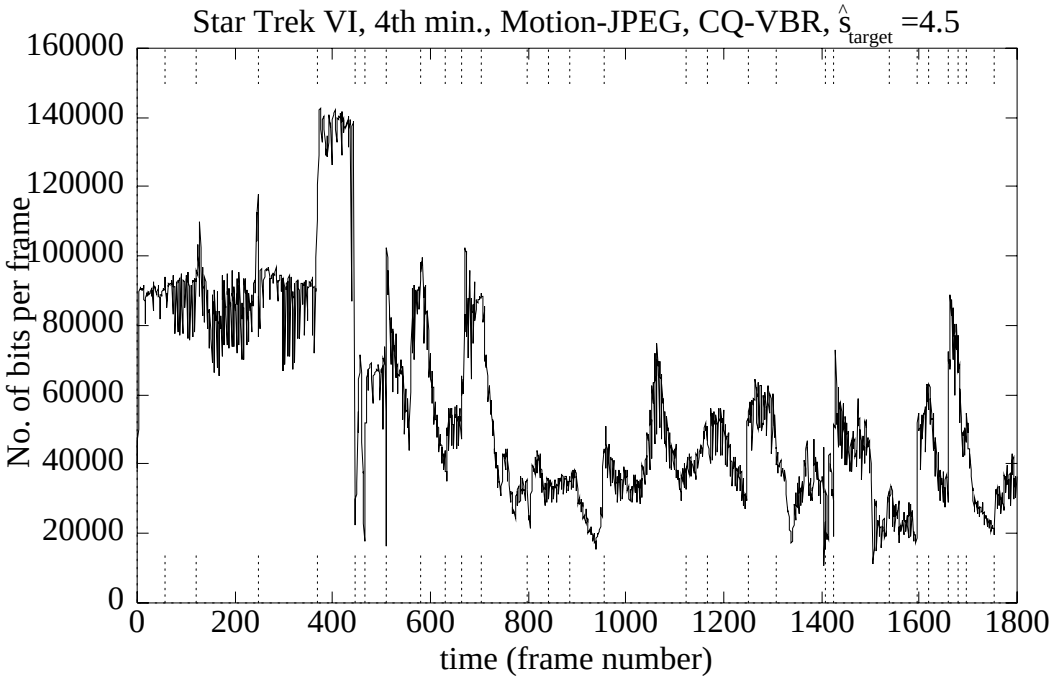


Figure 133: Number of bits per frame versus time for Star Trek, Motion-JPEG, CQ-VBR, $\hat{s}_{target}$=4.5. (Average number of bits per frame = 57.3 kbits.)

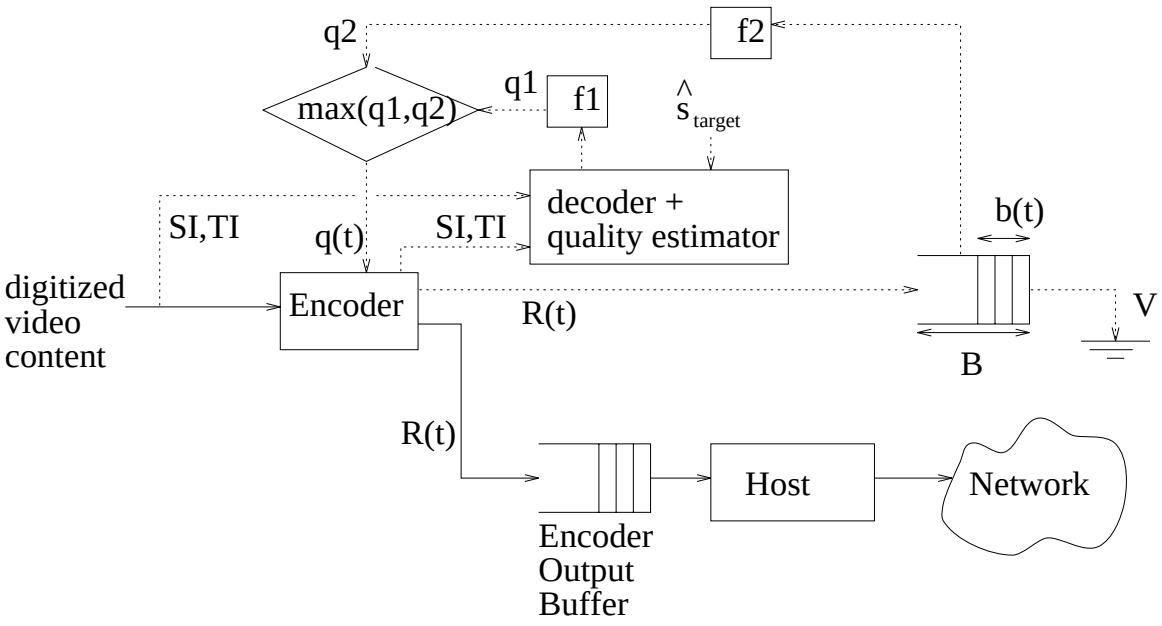


Figure 134: Block diagram of the encoder for JPQC-VBR encoding

	$\hat{s}_{avg}$	$\hat{s}_{std}$	$\hat{s}_{min}$	$\hat{s}_{max}$
JPQC-VBR, V=1536 kb/s, B=153.6 kbits	4.47	0.08	4.20	4.58
JPQC-VBR, V=1024 kb/s, B=1024 kbits	4.46	0.09	4.11	4.56
JPQC-VBR, V=768 kb/s, B=1536 kbits	4.41	0.12	4.02	4.53
CQ-VBR	4.51	0.08	4.28	4.64

Table 5: Quality statistics for Star Trek, H.261, JPQC-VBR, $\hat{s}_{target}$=4.5.

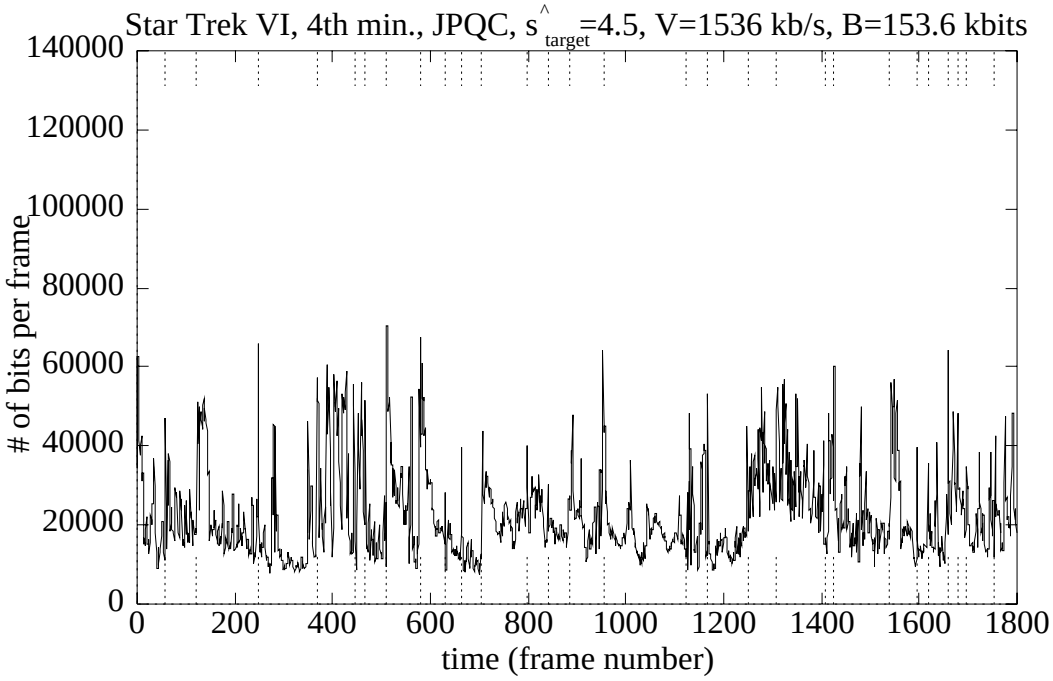


Figure 135: Number of bits per frame versus time for the Star Trek sequence, JPQC-VBR, $\hat{s}_{target}$=4.5, V=1536 kb/s, B=153.6 kbits.

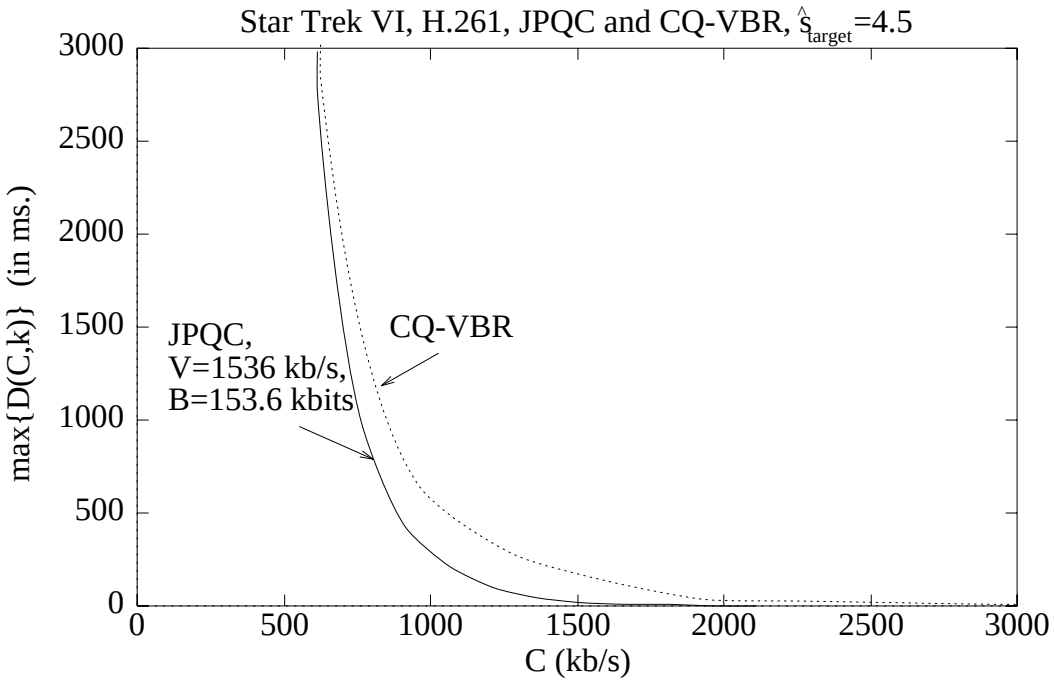


Figure 136: $\max_k\{D(C,k)\}$ versus C for the Star Trek sequence, JPQC-VBR, $\hat{s}_{target}$=4.5, V=1536 kb/s, B=153.6 kbits, and CQ-VBR, $\hat{s}_{target}$=4.5.

	$\hat{s}$			
	Average	Std. Dev.	Minimum	Maximum
Commercials	4.51	0.13	3.98	4.70
Raiders	4.47	0.10	4.24	4.70
Star Trek VI	4.47	0.08	4.20	4.58
Terminator 2	4.48	0.06	4.32	4.58
Videoconferencing	4.49	0.08	4.28	4.60

Table 6: Quality statistics for all five sequences, JPQC-VBR, $\hat{s}_{target}$=4.5, V=1536 kb/s, B=153.6 kbits.

	Frame Size (kbits)			
	Average	Std. Dev.	Peak	Minimum
Commercials	34.1	17.2	112.3	6.1
Raiders	33.6	17.1	65.7	6.5
Star Trek VI	20.6	11.5	70.4	6.6
Terminator 2	17.7	9.7	62.6	6.5
Videoconferencing	14.5	6.8	51.3	6.8

Table 7: Frame size statistics for all five sequences, JPQC-VBR, $\hat{s}_{target}$=4.5, V=1536 kb/s, B=153.6 kbits.

	$\hat{s}$							
	CBR				OL-VBR			
	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.	Min.	Max.
Commercials	3.56	0.58	1.0	4.70	4.1	0.27	3.26	4.56
Raiders	4.00	0.75	1.0	4.53	3.96	0.31	3.00	4.33
Star Trek VI	4.04	0.31	2.91	4.69	3.93	0.40	2.45	4.40
Terminator 2	3.92	0.33	3.16	4.51	3.96	0.16	3.67	4.32
Videoconferencing	4.06	0.23	3.63	4.51	4.1	0.18	3.9	4.3

(a) Encoded at the same average rate as the CQ-VBR sequences with $\hat{s}_{target}$=4.0

	$\hat{s}$							
	CBR				OL-VBR			
	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.	Min.	Max.
Commercials	4.41	0.18	3.11	4.75	4.52	0.17	3.82	4.72
Raiders	4.49	0.09	4.03	4.67	4.56	0.06	4.35	4.71
Star Trek VI	4.46	0.13	4.10	4.72	4.54	0.16	4.15	4.70
Terminator 2	4.50	0.11	4.29	4.66	4.51	0.07	4.28	4.61
Videoconferencing	4.52	0.06	4.39	4.70	4.57	0.04	4.37	4.67

(b) Encoded at the same average rate as the CQ-VBR sequences with $\hat{s}_{target}$=4.5

Table 8: Quality statistics for all five sequences, H.261, CBR and OL-VBR, encoded at the same average rate as their CQ-VBR counterparts.

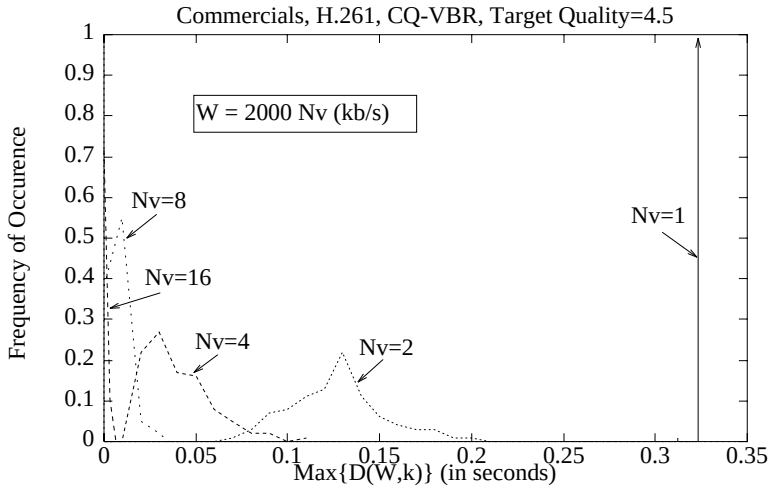
	Comm.	Raiders	Star Trek	Term. 2	Videoconf.
CQ-VBR $\hat{s}_{min}$	3.8	3.8	3.8	3.8	3.8
CQ-VBR Avg. Rate (kb/s)	600	330	360	300	270
CBR Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	1200	510	540	360	300
OL-VBR Avg. Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	900	450	540	300	270
OL-VBR q_0	14	14	10	17	22

(a) $\hat{s}_{target}$=4.0

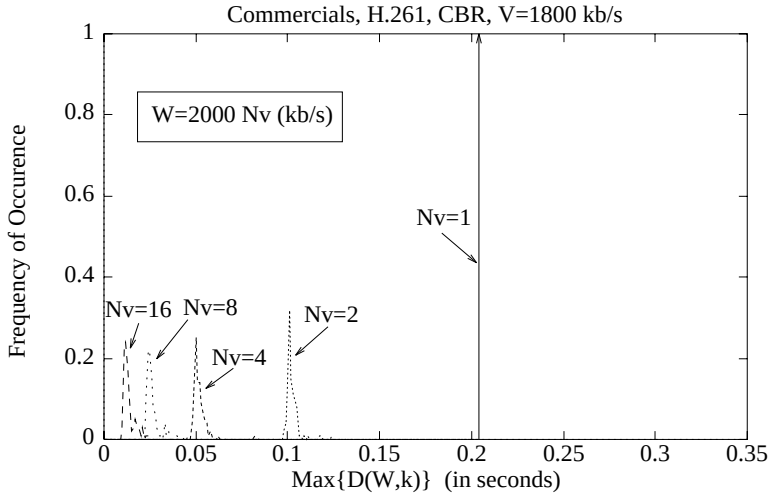
	Comm.	Raiders	Star Trek	Term. 2	Videoconf.
CQ-VBR $\hat{s}_{min}$	4.3	4.4	4.3	4.4	4.3
CQ-VBR Avg. Rate (kb/s)	1050	960	630	540	450
CBR Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	1800	900	660	510	480
OL-VBR Avg. Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	1500	750	750	510	450
OL-VBR q_0	10	8	6	10	16

(b) $\hat{s}_{target}$=4.5

Table 9: Comparison of average bit rates of CQ-VBR, CBR, and OL-VBR encoded sequences for the same $\hat{s}_{min}$.



(a) CQ-VBR, $\hat{s}_{target}$=4.5 (Avg. rate=1050 kb/s)



(b) CBR at the same $\hat{s}_{min}$ (V=1800 kb/s)

Figure 137: Histogram of maximum delay incurred at the multiplexer buffer for the Commercials sequence, CQ-VBR at $\hat{s}_{target}$=4.5 and CBR at the same $\hat{s}_{min}$ as in CQ-VBR; $W = 2000N_v$ (kbits).