PERFORMANCE EVALUATION OF ETHERNETS AND ATM NETWORKS CARRYING VIDEO TRAFFIC

İsmail Dalgıç and Fouad A. Tobagi

Technical Report No. CSL-TR-96-702

August 1996

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

Performance Evaluation of Ethernets and ATM Networks Carrying Multimedia Traffic

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

In this report the performance of Ethernets (10Base-T and 100Base-T) and ATM networks carrying multimedia traffic is presented. End-to-end delay requirements suitable for a wide range of multimedia applications are considered (ranging from 20 ms to 500 ms). Given the specific nature of the network considered and the maximum latency requirement, some data is lost. Data loss at the receiver causes quality degradations in the displayed video in the form of discontinuities, referred to as glitches. We define various quantities characterizing the glitches, namely, the total amount of information lost in glitches, their duration, and the rate at which glitches occur. We study these quantities for various network and traffic scenarios, using a computer simulation model driven by real video traffic generated by encoding video sequences. We also determine the maximum number of video streams that can be supported for given maximum delay requirement and glitch rate. We consider and compare the results for various types of video contents (video conferencing, motion pictures, commercials), two encoding schemes (H.261 and MPEG-1), and two encoder control schemes [Constant Bit Rate (CBR) and Constant-Quality Variable Bit Rate (CQ-VBR)], considering also scenarios where the traffic consists of various mixtures of the above. We show that when the video content is highly variable, both 100Base-T Ethernet and ATM can support many more CQ-VBR streams than CBR streams. When the video content is not much variable, as in a videoconferencing sequence, then the number of

CBR and CQ-VBR streams that can be supported are comparable. For low values of end-to-end delay requirement, we show that ATM networks can support up to twice as many video streams of a given type as Ethernets for a channel capacity of 100Mb/s. For relaxed end-to-end delay requirements, both networks can support about the same number of video streams of a given type. We also determine the number of streams supportable for traffic scenarios consisting of mixtures of heterogeneous video traffic sources in terms of the video content, video encoding scheme and encoder control scheme, as well as the end-to-end delay requirement. We then consider multihop ATM network scenarios, and provide admission control guidelines for video when the network topology is an arbitrary mesh. Finally, we consider scenarios with mixtures of video and data traffic (with various degrees of burstiness), and determine the effect of one traffic type over the other.

Key Words and Phrases: Network Performance, Ethernet, ATM, 10Base-T, 100Base-T, Video, Multimedia, Constant Bit Rate (CBR), Variable Bit Rate (VBR), Statistical Multiplexing, Admission Control.

1 Introduction

Many multimedia applications are distributed in nature and involve networking and communications. Examples of such applications are: video-on-demand for training and education, conferencing, and computer supported collaborative work. In this report, we address the performance of Ethernets and ATM networks supporting multimedia traffic.

The characteristics of multimedia traffic differ substantially from those of traffic encountered in more traditional data applications. These differences pertain to traffic patterns, data rates, delay and reliability requirements, modes of communication, and the need to integrate services for all media in the same networks. In turn, these differences in traffic characteristics place new requirements on the networks that are to support multimedia applications.

The focus in this report is on audio/video traffic either alone or in the presence of data traffic generated by traditional data applications. Audio/video traffic is stream oriented, i.e., it consists of a more-or-less continuous flow of information, and thus requires a certain amount of network bandwidth to be available on a somewhat continuous basis. Data traffic, on the other hand, tends to be bursty, and no preconceived idea of a required data rate is ever associated with a source or a session. Furthermore, the data rate associated with the audio/video traffic (and particularly video traffic) generated by a source is much higher than the average data rate corresponding to a single data source.

Another important new requirement that multimedia applications introduce is the maximum end-to-end delay requirement; end-to-end delay is defined as the time from when the information is generated at one end until the time it is available at the other end; depending on the interactiveness of the application, the maximum end-to-end delay requirement may be anywhere from 100 ms (as is the case for videoconferencing) to upwards of one second (as may be the case for one-way broadcasting). Traditional data applications, on the other hand, place no specific delay requirement on the network. With some multimedia applications such as data conferencing, it is important to achieve at the same time both timely delivery of information and full reliability, in which case it is important to guarantee that all information reach the destination(s) in time and that lost information be recovered within the delay constraint. With audio/video applications, on the other hand, a certain amount of information loss is acceptable. In this case, no end-to-end error/loss detection

and recovery mechanism is used, and some packets may be allowed to exceed the maximum end-to-end delay. Information that is received at the destination past the maximum allowed delay cannot be displayed (i.e., is considered lost), and thus leads to quality degradation in the displayed video in the form of discontinuities referred to as glitches. One could use packet loss rate as a measure of such quality degradation, but this would not constitute a good choice. First, it does not translate easily to what a viewer perceives as a result of the information loss. Secondly, due to the dependence that exists in the video encoded bit stream, the loss of information contained in a single packet has impact that extends beyond just that information; conversely, given that packet loss is often bursty, the loss of several packets may account for the same perceived degradation as that due to the loss of a subset of these packets. Accordingly we choose to express the quality degradation in terms of glitch statistics.

Given the complexity of the problem at hand, it is not possible to provide accurate analytical models for the evaluation of networks carrying video traffic. Accordingly, our approach is to use computer simulation. Furthermore, given the dependencies that exist in the encoded video bit stream (which differ from one compression scheme and its syntax to another), and given the interest to derive glitch statistics, the simulation model is driven by real video traffic generated by encoding video sequences, as opposed to artificially generated traffic using analytical models [1, 2].

In this report, we obtain glitch statistics for various network and traffic scenarios. We cover a wide range of end-to-end delay requirements (20 ms–500 ms), different video contents, two video encoding schemes (H.261 and MPEG-1) and three encoder control schemes (Constant Bit Rate (CBR), Open-Loop Variable Bit Rate (OL-VBR) and Constant-Quality Variable Bit Rate (CQ-VBR)). We also determine the maximum number of video streams supportable for given maximum delay requirement and maximum tolerable glitch rate.

Given that multimedia applications would normally share the same network with traditional data applications, we also consider scenarios where audio/video traffic is mixed with data traffic. We determine the effect of data load and data burst size on the number of audio/video streams supportable. We also determine the effect of audio/video traffic on the average and standard deviation of data traffic delay, and data packet loss rate.

The remainder of this report is organized as follows. In Section 2, we briefly describe the prior work on the performance evaluation of networks carrying multimedia traffic.

In Section 3, we describe the system under consideration: that is, the network scenario, the components in the system and their contribution to the end-to-end delay, the CBR, OL-VBR, and CQ-VBR encoder control schemes, and the transport layer functions at both the senders and the receivers. In Section 4, we give some preliminary performance results for Ethernet carrying CBR video. In that section, we do not use real video traffic traces; instead, we consider video traffic to consist of fixed size packets transmitted at constant intervals, and ignore the delay that would be incurred in the rate control buffer. Furthermore, we evaluate the network performance in terms of packet loss rate. This allows us to study the tradeoff between packet formation time and network delay, independently of the specifics of the video traffic. In Section 5, we show the inadequacy of packet loss rate as a measure of degradations in the displayed video. We develop more accurate and relevant measures, and characterize the statistics of glitches that occur due to packet loss. In Section 6, we examine the network performance for CBR and Constant-Quality VBR video traffic, using glitch rate as the primary performance measure, and considering that all the video streams in the network are of the same type. We give the number of streams supportable for both CBR and CQ-VBR traffic, and compare them at the same encoded video quality. We show that up to twice as many CQ-VBR streams as CBR streams can be supported under given quality and delay constraints for both ATM and Ethernet (for a network bandwidth equal to 100 Mb/s). We also show that for low values of end-to-end delay requirement, ATM networks can support up to twice as many video streams of a given type as Ethernets. For relaxed end-to-end delay requirements, both networks can support about the same number of video streams of a given type. In Section 7, we give the number of streams supportable for traffic scenarios involving a mixture of different types of video traffic, the differences pertaining to the video encoding scheme and its control, the video content, and the end-to-end delay requirement. We then consider multihop network scenarios for ATM networks in Section 8, determining the number of streams supportable for an arbitrary mesh topology. We show that the number of streams that can be supported on a given link is very insensitive to the number of hops that the streams travel. In Section 9, we consider scenarios where there is a mixture of video and data traffic on the network. We examine the effect of one traffic type over the other. Our conclusions are presented in Section 10.

2 Prior Work

The approach in the prior work assessing the network performance carrying multimedia traffic is usually based on computer simulation driven either by video frame size traces, or by analytical models. The network performance is measured by packet loss rate, which is given as a function of the load offered to the network. In the following, we review the relevant work, grouped according to the network type considered.

2.1 Ethernet

In [3], Carrier Sense Multiple Access (CSMA) networks carrying audio streams are simulated. It is shown that there exists an optimum packet size for which the number of streams supportable by the network is maximum. The maximum number of streams that can be supported on the network is given for various stream rates, and maximum end-to-end delay and packet loss rate requirements.

In [4], audio streams and non-bursty data traffic carried on an Ethernet (both 10 Mb/s and 100 Mb/s) are simulated. The maximum number of streams that can be supported in the presence of non-bursty data traffic is presented, given the stream rate, stream delay and loss requirements, as well as the data load. The voice packet size is assumed to be dynamically adjusted to cope with network contention. Similar results are also given in the same paper for Expressnet [5] and IEEE 802.4 Token Bus[6], and are compared with the results for Ethernet. It is shown that the Expressnet and the Token-Bus networks yield better performance compared to Ethernet, especially when the network bandwidth and the network diameter get large.

In [7], mixtures of video streams and non-bursty data traffic (with Poisson packet arrivals) are simulated on a 10 Mb/s Ethernet. Video traffic is artificially generated using analytical models, and the packet size is chosen arbitrarily to be a certain fixed fraction of a frame (e.g., one-third of a frame). Video and data traffic are assumed to generate an equal amount of load. The total load is varied, and the maximum number of streams supportable is determined given an end-to-end delay constraint and a maximum tolerable packet loss rate.

In [8], again mixtures of video streams and data traffic are simulated on a 10 Mb/s Ethernet. The video traffic is generated using similar models as in [7], and the packet size

is again chosen arbitrarily to be a fixed fraction of a frame. However, the data traffic is generated according to packet traces obtained through measurements on a network. It is shown that for the same average data load, the data traffic based on the packet traces has a more severe effect on the video performance compared to Poisson data traffic.

In [9], mixtures of CBR video streams and Poisson data traffic are simulated on a 10 Mb/s Ethernet. The video streams are generated using a 300-frame videoconferencing trace encoded at 64 kb/s; each frame is sent in a separate packet; (given the low bit rate, each frame is small enough to fit in a single Ethernet packet.) It is shown that when the end-to-end delay constraint is around 45 ms, and the tolerable packet loss rate is 1%, about 50% network utilization can be achieved.

In [10], also mixtures of mixtures of CBR video streams and Poisson data traffic are simulated on a 10 Mb/s Ethernet. CBR video streams of 200 kb/s are considered, and the traffic is generated simply by creating packets of a constant size (520 bytes), and a given fixed rate (50 packets per second). It is shown that a good level of network utilization can be reached (i.e., 60–70%) while meeting the delay and loss constraints of the video streams.

In addition to these simulation studies, in [11], the performance of a number of video streams sent by a single server in the presence of artificially generated background traffic is subjectively assessed in an experimental setup consisting of several workstations connected by a 10 Mb/s Ethernet. The background data traffic is generated by a single station. Under these conditions, it is shown that around 70-80% of network utilization can be achieved on an Ethernet segment before glitching starts to occur in the displayed video.

2.2 ATM

There is also a great deal of prior work studying the statistical multiplexing of video traffic on an ATM multiplexer.

In [12], an ATM multiplexer is simulated, driven by frame size traces of a 30-minute videoconferencing sequence. The multiplexer output bandwidth is taken to be 45 Mb/s, and the buffer size is chosen to correspond to a maximum delay of 4–5 ms in the buffer. 16 video sources are multiplexed, and the cell loss rate for each individual source is given. It is shown that with FIFO scheduling, the cell loss rate varies largely from source to source due to periodicity. Smoothing the traffic at the source over one frame period is shown to reduce this effect, but not completely eliminate it.

In [13], VBR video transmission is simulated over an ATM multiplexer. The multiplexer output bandwidth is taken to be either 45 Mb/s or 155 Mb/s. Several buffer sizes are chosen, resulting in a range of maximum delays from 0.5 ms to 16 ms in the buffer. The video traffic is generated according to a Markov chain model, whose parameters are determined based on videoconferencing sequences. It is shown that after the cell loss rate becomes nonnegligible, it increases very sharply with the network load. It is also shown that cell losses occur in clusters, even for very low cell loss rates such as 10^{-9}.

In [14], the main focus is to determine models that capture the statistical multiplexing characteristics of video sources. Accordingly, an ATM multiplexer is simulated, driven both by video traces for several long video sequences, and by artificially generated traffic according to analytical models that match the characteristics of the video sequences under consideration. The multiplexer output bandwidth is taken to be 45 Mb/s, and the buffer size is chosen to correspond to a maximum delay of 4–5 ms in the buffer. It is shown that there is no single model that can accurately represent the statistical multiplexing characteristics for all the video sequences considered.

In [15], the focus is also to determine models that capture the statistical multiplexing characteristics of video sources. A model is developed that matches the data for one 23-minute video sequence. The model is validated by simulating an ATM multiplexer driven by either the video frame size trace, or the analytical video traffic model. The multiplexer buffer size is chosen in a range from 50 to 300 cells, and the multiplexer output bandwidth is varied to achieve a given network utilization for a given number of video sources. It is shown that the simulation results agree for the trace and model driven simulations for the buffer sizes considered. However, as the buffer size increases, the model-based simulation starts to underestimate the cell loss rate.

In [16], several video sources are simulated over a bufferless multiplexer with an output bandwidth of 155 Mb/s. No delay constraint is specified, and cells are considered lost whenever the instantaneous traffic rate exceeds the channel capacity. The video sources are modelled as periodic ON-OFF sources, whose peak and average rates are chosen to match some OL-VBR videoconferencing sequences. It is shown that smoothing the video sources at the source can significantly improve the multiplexing performance.

In [17], an ATM multiplexer is simulated, with traffic generated according to an analytical model for video traffic. The channel capacity is taken to be 155 Mb/s. The multiplexer

buffer size considered is quite larger compared to the studies described above; it corresponds to a maximum buffer delay of 100 ms (i.e., about 36000 cells at 155 Mb/s). Cell loss rate is given as a function of the traffic load offered to the network. Furthermore, average buffer overflow duration is given for various traffic loads, and it is shown that the buffer overflow duration gets larger when the load is increased. It is also shown that reducing the peak traffic rate by smoothing helps reduce both the cell loss rate and the average buffer overflow duration.

In [18] and [19], MPEG-1 video performance over ATM networks is investigated. A simple model for MPEG-1 video is devised, and the performance of an ATM multiplexer is simulated, driven by video traffic generated according to the developed model. Two traffic scenarios are considered. In the first one, all video sources generate the same type of frame (i.e., I, P, or B) at the same time. In the second scenario, the alignment of frame types among multiple sources are chosen at random (corresponding to random starting times for the sequences). It is shown that the performance is much worse in the former scenario, since the I frames contain many more bits than the other frames, which create large traffic peaks in the former scenario.

In [20], a Markov-chain model for video traffic is developed, and the performance of an ATM multiplexer is studied by means of computer simulations driven by traffic generated according to the devised video traffic model. No statistical multiplexing is considered, and each video stream is assumed to request, and be granted a certain amount of bandwidth, which may vary in time. Three bandwidth allocation schemes are considered. In the first one, each video source requests a fixed bandwidth equal to the mean plus standard deviation of the source traffic. In the second one, the sources request a variable amount of bandwidth on a frame by frame basis; the amount of bandwidth to be requested is predicted by the source based on the previous frame size. Assuming that such variable bandwidth allocation is possible, it is shown that this scheme performs better than the first one. In case variable bandwidth allocation is not feasible, a third scheme is presented which assumes that the video can be encoded such that it can be divided into high and low priority portions. A fixed bandwidth is allocated for each source, and the video is coded such that the high priority portion just fits in the allocated bandwidth.

2.3 Open Problems

In all of the prior work described above, packet loss rate is used as the measure of video quality degradation due to packet losses. However, for reasons explained in the introduction, this is not a good choice, and the network performance carrying video traffic must be evaluated using more accurate and relevant performance measures.

Furthermore, in these papers the video quality degradation due to compression is not considered. Hence, the data rates for video are chosen arbitrarily. Moreover, the results from the prior work make it impossible to compare network performance for CBR and VBR video; indeed, it is still an open problem to determine whether video must be encoded CBR or VBR for transmission over networks.

Finally, the use of different video sequences and different models for generating the video traffic, and the different application delay and packet loss rate requirements considered make it difficult to compare the results provided in these papers so as to be able to contrast the performance of different networks.

We address those problems in this report.

3 System Description

In this section, we describe the network scenario under consideration. We then describe the components traversed by a video signal from its source where it is generated to its destination where it is displayed, and identify the contribution to the end-to-end delay of each of these components. We also describe the CBR, OL-VBR, and CQ-VBR encoder control schemes considered. We finally describe the functions performed by the video transport layer.

3.1 Network and Traffic Scenarios

For Ethernets, the network scenario under consideration is a 10Base-T or 100Base-T segment consisting of a number of stations connected to a multiport repeater by means of unshielded twisted pair cables. (See Figure 1.) The network bandwidth is denoted by W, and is clearly 10 Mb/s for 10Base-T and 100 Mb/s for 100Base-T. We consider that all station cables are 100 meters long, the maximum allowed by the 10Base-T and 100Base-T

standards. The propagation delays are computed using 0.6 times the speed of light, as appropriate for twisted-pair wires.

As for the ATM networks, the main scenario under consideration is an ATM switch interconnecting a number of workstations. (See Figure 2.) We consider that the switch is non-blocking, and therefore the only shared resources in the switch are the buffers at the output ports, and the output links. We focus in particular on one output port. We denote the output port buffer size by M (bits), and the capacity of the output link again by W (bits/sec).

For both network types, we consider that there are N_v video stations each of which generates a video stream destined to another station on that output port. We consider that there are also N_d data stations generating data traffic; each such station is considered to generate fixed size messages comprising M_d bits each, with uniform interarrival times, the average of which is denoted by u. The aggregate data load (in bits/sec.) for all N_d stations is denoted by G_d, whereby $G_d = M_d N_d / u$.

For ATM networks, we also consider multihop scenarios, and determine the maximum number of video streams that can be supported on a switch output link as a function of the number of hops that the streams travel. The particular multihop traffic and network scenarios considered are described in Section 8.

3.2 End-to-end Delay Components

We define the end-to-end delay D_{ete} for a video pixel as the time from when that pixel is scanned by the camera until it is displayed at the receiver. Therefore, if we denote the end-to-end delay constraint by D_{max}, then the inequality $D_{ete} \leq D_{max}$ must be satisfied for each pixel that is displayed at the receiver; any pixel that exceeds this delay constraint cannot be displayed, and is considered lost.

In Figure 3, we show the components traversed by a video signal from its source to its destination. In Figure 4, we show the associated timing diagram, identifying the delay introduced by each component. As it can be seen in Figure 3, a video frame is first scanned by the camera, and converted to an analog signal. The analog signal is then converted to digital form by the digitizer and passed to the encoder. We denote the time from when a point in the analog signal enters the digitizer until the corresponding digital information is passed to the encoder as D_{dig}. The digitized data is then compressed by the encoder, and

placed at the encoder's output buffer. For DCT-based encoding schemes used in MPEG and H.261, the smallest unit of information that the encoders and decoders operate on is called a *macroblock*, which is a group of 16x16 pixels [21, 22]. Since all the pixels contained in a macroblock experience the same delays, we define the following delays on a macroblock basis. We denote the delay for encoding information corresponding to a macroblock, say k, as $D_{enc}(k)$ (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). The data in the encoder's output buffer is retrieved by the host and packetized for transmission over the network.

The data retrieved by the host is packetized at the transport layer. Let us first consider the case that the network under consideration is Ethernet. We let $D_{pack}(k)$ denote the packet formation delay experienced by macroblock k; that is, the delay from when the bits corresponding to macroblock k enter the encoder's output buffer, until the packet containing those bits is formed. (If the bits corresponding to a given macroblock k are placed in multiple packets, then the packet formation delay $D_{pack}(k)$ is defined for the portion of the macroblock placed in the packet that is formed last.) The packets that are formed are then placed in the transport layer's transmit queue for transmission over the network. We define the network delay incurred by a packet, and thus for any macroblock k contained in the packet (denoted as $D_{net}(k)$,) as the time elapsed from when the packet has been formed, until it is received at the destination. $D_{net}(k)$ is composed of two components: (i) the queueing delay at the transport layer, denoted by D_q, which is the time from when the packet is formed until it is submitted to the MAC layer for transmission, and (ii) the MAC delay, denoted by D_{MAC}, which is the time spent at the MAC layer due to contention, transmission over the channel, and the propagation delay. We do not consider any protocol processing delays at the sender or the receiver.

In the case of ATM networks, the packets are passed to the ATM Adaptation Layer, which encapsulates them into Protocol Data Units (AAL-PDU). Then the AAL-PDU's are segmented into cells. In this case, we define $D_{pack}(k)$ as the cell formation delay experienced by macroblock k; that is, the delay from when the bits corresponding to macroblock k enter the encoder's output buffer, until the ATM cell containing those bits is formed. (Just as in the Ethernet, if the bits corresponding to a given macroblock k are placed in multiple cells, then the cell formation delay $D_{pack}(k)$ is defined for the portion of the macroblock placed

in the cell that is formed last.) The cells that are formed are then sent over the network. Again, we define the network delay incurred by a cell, and thus for any macroblock k contained in the cell (denoted as $D_{net}(k)$,) as the time elapsed from when the cell has been formed, until it is received at the destination.

At the destination, the received video information is decoded and displayed. We denote the decoding delay for macroblock k by $D_{dec}(k)$, from the time all the bits corresponding to macroblock k enter the decoder, to the time the macroblock is decoded, and is ready to be displayed. In general, the decoding of a macroblock and its display are not synchronized, and thus there is a delay $D_{disp}(k)$ corresponding to the time from when it is decoded until it is displayed. Thus, the total end-to-end delay incurred by a macroblock k is

$$D_{ete}(k) = D_{dig}(k) + D_{enc}(k) + D_{pack}(k) + D_{net}(k) + D_{dec}(k) + D_{disp}(k).$$

In this report, we consider that the digitizer/encoder and decoder/display components of the system are streamlined to the full extent possible in order to minimize the delays associated with those components. The digitizer samples and quantizes the analog video signal in real time, without any delay. The resulting samples (i.e., pixels) are passed to the encoder as soon as all the pixels corresponding to a macroblock are digitized. Therefore, the $D_{dig}(k)$ for any macroblock k is equal to the time it takes to produce 16 rows of pixels by the digitizer (which are placed in a buffer at the output of the digitizer, and passed to the encoder as a single block). As an example, for digitizing NTSC video (30 frames per second) using SIF frame format (i.e., 352x240 pixels), this delay is equal to (16/240)$\times$(1/30 seconds)=2.2 ms for every macroblock. (Note that typically such digitization is done by sampling only the lines in the odd fields of the analog video, in which case there is no additional delay due to interlacing of the lines in the analog video frame.) We consider that the macroblocks are encoded and placed at the encoder's output buffer one macroblock at a time. Therefore, in order to be able to keep up with the incoming data, the encoder must be able to encode each macroblock within $\tau \stackrel{\triangle}{=} 1/FM$ time, where F is the rate of the encoded frames, M is the number of macroblocks in a frame (i.e., for SIF resolution, M=330; thus, at 30 fps, τ =101 μs). We consider that the encoder processes and outputs one macroblock at regular time intervals equal to τ. Thus, $D_{enc}(k) = \tau$ for all k, which is small enough to be neglected. Similarly, 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 τ. Therefore, $D_{dec}(k)$ is also negligible. As for the displaying delay, we

consider that the decoder and the display are synchronized such that the display scanning begins when the first line of macroblocks is decoded; therefore, $D_{disp}(k)$ is equal to the decoding time of one line of macroblocks, (e.g., 2.2 ms for SIF). To summarize, under the assumptions of a highly streamlined system, $D_{dig}(k)+D_{enc}(k)+D_{dec}(k)+D_{disp}(k) \approx 4.4$ ms for all macroblocks.

Given that $D_{dig}(k)$, $D_{enc}(k)$, $D_{dec}(k)$, and $D_{disp}(k)$ are considered here to be constant for each macroblock, the end-to-end delay requirement for macroblock k can be expressed as

$$D_{pack}(k) + D_{net}(k) \leq D_{max} - (D_{dig} + D_{enc} + D_{dec} + D_{disp})$$

[1]. The packetization and network delays are in general variable, and for some macroblocks their sum might exceed the bound given in the right hand side of the equation; in this case, the macroblock is dropped. If it is equal to the bound, then the macroblock must be decoded and displayed immediately. If it is less than the bound, then it is delayed by an amount equal to $D_{max} - D_{ete}(k)$, so that the continuity of the video playback is maintained. So, for such macroblocks an additional delay equal to $D_{max} - D_{ete}$ (referred to as D_{buf}) is incurred in a playback buffer at the receiver. Thus a macroblock is either displayed at exactly D_{max} following the time it is scanned by the camera at the sender, or dropped.

Note that for MPEG, when B frames are used, a delay equal to $1/F$ times the number of consecutive B frames in a group of pictures (GOP) must be added to D_{enc}, since the B frames can only be encoded after the subsequent I or P frame is encoded, and, a delay equal to $1/F$ must be added to D_{dec}, in order to maintain the continuity of the displayed video. As an example, for a GOP structure of IBBPBB... and for F=30 fps, the additional delay due to having B frames is 100 ms.

3.3 Encoder Control Schemes

Most of the existing video encoders are designed for CBR encoding, where the rate of the encoded video is targeted to be kept at a given rate V at all times by dynamically adjusting the quantizer scale [23, 24]. This is achieved by placing a *hypothetical rate control buffer* of

[1]Note that many encoders and decoders today operate on a frame-by-frame basis. In this case, the quantities $D_{dig}(k)$, $D_{enc}(k)$, $D_{dec}(k)$, and $D_{disp}(k)$ may again be considered constant for all macroblocks, and each one of these delays would be equal to at least $1/F$. Therefore, the equation given here for the end-to-end delay requirement applies to that case as well.

size B_{max} at the output of the encoder which gets drained at rate V, and ensuring that the buffer does not underflow or overflow by inserting stuffing bits, or dropping macroblocks, whenever necessary. In order to reduce the likelihood of such underflows or 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 $k+1$. (See Figure 5.) 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.,

$$q(k+1) = \begin{cases} \left\lceil \frac{q_{max}}{\alpha} \frac{b(k)}{B_{max}} \right\rceil & \text{if } b(k) < \alpha B_{max} \\ 31 & \text{otherwise} \end{cases}$$

where α is a constant which is recommended to be equal to 0.4 [23, 24]. The buffer occupancy $b(k)$ can be expressed as $b(k) = \sum_{i=1}^{k}(m_i - V\tau)$ where m_i is the number of bits for macroblock i.

Note that with this scheme, there are still short-term variations in the traffic. When CBR video is to be sent over the network, V and B_{max} must be chosen appropriately so that an acceptable quality is maintained while not producing a large number of excessive bits. The values of V and B_{max} to achieve a given quality objective depend on the video content [25]. Therefore, in order to achieve a given quality objective at all times with CBR, during some scenes more bits are produced than the minimum possible amount.

One can also use variable bir 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 q_0 at all times, as illustrated in Figure 6. 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. We have shown in [25] that for any given q_0, both the quality and rate of the encoded video vary according to the content, and it is difficult to specify a clear objective when selecting q_0.

In order to produce only as many bits as needed to achieve a given quality objective, we have proposed a video encoder control scheme which maintains the quality of the encoded video at a constant level $\hat{s}_{target}$, referred to as Constant Quality VBR (CQ-VBR) [26]. This scheme uses a feedback control mechanism to adjust the encoder parameters as shown in Figure 7. 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 have used such a measure that has been developed at the *Institute for Telecommunication Sciences (ITS)*[27]. This measure is denoted by $\hat{s}$ and takes values in the range between 1 and 5 (1 being very annoying and 5 imperceptible). The measure $\hat{s}$ is designed so that it correlates well with subjective tests covering a wide variety of video quality impairments. The correlation coefficient between $\hat{s}$ and subjective scores was 0.94 for the subjective tests conducted by the authors of [27], indicating that there is a good fit between the estimated and the subjective scores.

In the CQ-VBR scheme the bit rate is allowed to vary so as to produce a constant level of quality. Since in a movie there are many scenes with different characteristics, the CQ-VBR encoding produces traffic with more variations compared to CBR traffic. However, as shown in [25], the average bit rate produced by the CQ-VBR computed over several such scenes is typically smaller than the CBR rate V when both schemes are to maintain the same minimum quality level $\hat{s}_{min}$ at all times (where the quality $\hat{s}$ is measured over one-second intervals, as appropriate to the rate of changes in the scene content and the response time of the human visual system). Therefore, an important issue that we address in this report is to compare the number of CQ-VBR and CBR video streams that can be supported on an Ethernet segment for a given $\hat{s}_{min}$.

3.4 Video Transport Layer Functions

The functions of the video transport layer at a sending station are: (i) packetization, and (ii) packet queueing and transmission, and possibly (iii) discarding of delayed video information (beyond the maximum allowed). The functions of the video transport layer at the receiving station are (i) de-packetization, (ii) timely delivery of macroblocks to the decoder, and (iii) discarding of delayed video information.

In order to perform these functions, we consider all stations to be time synchronized by means of a network time protocol such as Network Time Protocol (NTP) [28]. Furthermore, we assume that the video transport layer is aware of the macroblock boundaries in the video bit stream, and the timing associated with each macroblock. It is also assumed that the transport layer appropriately time-stamps each packet payload so that the delay incurred by each macroblock when the packet is received at the destination may be accurately deduced. (An appropriate method for timestamping would be to use a timestamp value

corresponding to the time when the first macroblock in the packet has been scanned by the camera; since the time between the generation of successive macroblocks is τ, the timing of all other macroblocks in the packet can also be deduced from the timestamp[2].) Thus, both at the sender and the receiver, the transport layer knows the time at which a macroblock is generated, and the delay incurred at each point of its journey. We also consider that at the receiver's transport layer, as well as at the AAL and ATM layers, appropriate mechanisms exist to detect the loss of any information.

The transport layer functions at the sending station can be described as follows.

(a) Packetization Function:

Macroblocks placed at the encoder's output buffer are retrieved by the transport layer to be packetized. Many packetization schemes can exist, depending on the underlying network type, when packets are formed, and how much data is placed in them. In the following, we describe packetization schemes considered here, first for Ethernet, and then for ATM.

(i) Ethernet:

For CBR, an immediately obvious method of packetization is to use packets of constant-size (denoted by P_v bits,) which are generated at regular intervals (denoted by T_f seconds) such that $P_v/T_f = V$ bits/sec; the first packet in a stream is formed T_f seconds after the encoder starts putting the bits it generates into the output buffer. We refer to this method as *Constant Size and Rate Packetization (CSRP)*. Note that for this packetization method, there are always at least P_v bits available at the encoder's output buffer just prior to the formation of each packet. Indeed, let $t = 0$ be the time when the encoder generates its first bit. Clearly, just prior to $t = 0$ the hypothetical rate control buffer is empty. From there on the buffer can never be empty, since the CBR scheme guarantees that the rate control buffer does not underflow. Therefore, at time T_f, the encoder must have already produced at least $P_v = T_f V$ bits. Thereafter, a similar argument applies at every point in time that are integer multiples of T_f. This packetization method represents constant bit rate traffic (grouped in

[2]Note that for encoders and decoders operating on a frame-by-frame basis, the timestamps are also on a frame-by-frame basis.

blocks of P_v bytes) offered to the network; thus, it would be the only appropriate method if the network was a circuit with a bandwidth equal to V. Note that in CSRP, once a packet is formed, no new data ever gets appended to it, nor any data is ever removed from it.

Another method of packetization is as follows. Let $t = 0$ be the time when the encoder generates its first bit. If before a predetermined time T_f elapses the buffer occupancy level reaches 1500 bytes (i.e., Ethernet's maximum packet payload size), then the entire content of the buffer is removed and placed in a single packet. Otherwise, fewer than 1500 bytes exist at the encoder output buffer at the time T_f, and a packet is formed with all the bytes in the buffer. Any time a packet is formed, the clock is reset, and the same procedure is repeated. We refer to this method as *Variable Size and Rate Packetization (VSRP)*. The idea behind this method is to make use of Ethernet's capability to accommodate variable size and rate packets in order to reduce the packet formation delay. Indeed, while for CSRP $D_{pack} \geq T_f$, for VSRP $D_{pack} \leq T_f$ for all the packets formed. For this scheme too, once a packet is formed, no new data ever gets appended to it, nor any data is ever removed from it.

A third packetization method, based on the idea of dynamically modifying the content of a packet as time progresses, is as follows. The packets are formed just as in the VSRP scheme, except for the case when the time T_f elapses after the previous packet has been formed, and the previous packet has not yet been transmitted (i.e., it is either in the MAC layer, or in the transport layer's transmit queue). In that case, formation of the new packet is delayed until the previous packet has been transmitted, or 1500 bytes accumulate in the encoder's output buffer; at that time, a single packet is formed with all the bytes in the encoder's output buffer, and the same procedure is repeated. With this method, no extra formation delay compared to VSRP is induced, but the network delay is minimized since there would be fewer packets that are queued for transmission. We refer to this packetization method as *Dynamic Packetization (DP)*.

It is important to note that for all three packetization methods, there is a trade-off between the network delay and T_f. As T_f gets smaller, the packet size gets smaller; therefore, the number of packets contending for the shared channel

increases, and thus the network delay increases. Conversely, the opposite is true as T_f gets larger. This suggests the existence of an optimum packet size that minimizes the packet formation plus network delays. In Section III a more detailed discussion of this effect is given.

(ii) ATM:

As far as packetization of video is concerned, the difference of ATM from the Ethernet is the use of fixed-sized cells, and the fact that the data is first packetized into AAL-PDU's, which are then segmented to ATM cells.

In accordance with our consideration of minimizing the end-to-end delay, here we consider that the streaming mode [29] is used when forming the AAL-PDU's so that as soon as a cell payload is delivered to the AAL, that payload is placed into a cell and transmitted. Another approach could be to use the messaging mode, so that an entire AAL-PDU is passed at once from the transport layer to the AAL. This would result in a greater packet formation time, but it would reduce the number of interrupts, which is an important issue in software-based protocol processing. We consider that approach as well, for purposes of comparison with the streaming mode approach. Note that we do not consider here any particular AAL protocol, and ignore the overhead due to AAL-PDU headers or trailers.

For CBR, a method of packetization analogous to the CSRP method can be devised for ATM, whereby cells are generated at regular intervals (denoted by T_f seconds) such that $(48\ bytes)/T_f = V$ bits/sec; again, the first packet in a stream is formed T_f seconds after the encoder starts putting the bits it generates into the output buffer. We refer to this method as *Constant Rate Cell Formation (CRCF)*. Note that using this method, and allocating the bandwidth V to the video stream, lossless transmission over an ATM network can be achieved. However, the cell formation time becomes equal to the delay incurred in the hypothetical rate control buffer plus T_f.

Another method of packetization, analogous to the VSRP method in Ethernet, is to simply form a cell as soon as the encoder generates 48 bytes of data. (At the data rates appropriate for video traffic, the cell formation delay does not ever become excessive.) We refer to this method as *Variable Rate Cell Formation (VRCF)*.

(b) Packet Queueing and Transmission Function:

(i) Ethernet:

We consider here that the LAN card implementing the MAC protocol can accommodate only one packet at a time. Therefore, the transport layer may submit a packet to the MAC layer only when service for the previous packet has been completed (i.e., the packet is either successfully transmitted or has exceeded its maximum number of collisions).

If at the MAC layer a packet exceeds the maximum number of collisions as specified by the 802.3 protocol (i.e., 16), then the MAC layer drops the packet and notifies the transport protocol. In that case, the transport protocol may resubmit the same packet to the MAC layer. In this report, we consider that a packet is resubmitted to the MAC layer as long as some macroblocks contained in the packet have not exceeded D_{max}. (Another possibility could be to additionally impose a limit on the number of resubmissions; we do not consider that case here.)

(ii) ATM:

The packets formed by the transport layer are passed to the AAL. If they are formed at an instantaneous rate greater than the peak rate allowed for that particular video stream, they are buffered at the transport layer and passed to the AAL at the appropriate time. In this report, we consider the limit on the stream peak rate to be equal to the channel capacity W.

(c) Discarding of Delayed Video Information Function:

Prior to submission or resubmission of a packet to the MAC layer, the transport layer checks the delay experienced by the packet. If all the macroblocks in the packet exceed their delay bound D_{max}, then the packet is dropped. (One could also discard only those macroblocks in a packet that exceed their delay bound; since we assume here that once a packet is formed it is not altered, we do not consider that approach).

Now consider the receiving station. For the ATM networks, we consider two data discarding modes at the receiver. In the first mode, we assume that the AAL delivers the received parts of an AAL-PDU to the transport layer even if some cells contained in it are

lost in the network; (this requires that the AAL operates in the *non-assured* mode [29]). In the second mode, any AAL-PDU which is not received in its entirety is discarded.

The functions of the receiver's transport layer can be described as follows. Upon receipt of a packet, the transport layer at the receiving station checks the time-stamp associated with the packet, and de-packetizes the information. If the first macroblock in the packet has not exceeded its delay bound, then all the data contained in the packet is received on time, and gets placed in the receiver's playback buffer. Otherwise, the transport layer parses the data contained in the packet, discarding those macroblocks that are late, and placing those that are on time in the playback buffer. As described above, each macroblock in the playback buffer is delivered to the decoder such that it is displayed at exactly D_{max} following the time it is scanned by the camera.

Note that one could also discard the contents of an entire packet if the first macroblock in the packet is received late. In such an approach, the transport layer at the receiver need not be aware of the macroblock boundaries. However, as we show in Section 5, with that approach significantly more macroblocks are not displayed for the same pattern of packet delays.

4 Preliminary Performance Results of Ethernets for CBR Video Based on Packet Loss Rate

In this section, we present a preliminary performance evaluation of Ethernet for CBR video packetized according to the CSRP scheme. We make two simplifying assumptions: first, we ignore the rate control buffer delay, and therefore consider the packet formation delay to be always equal to T_f (i.e., packets are considered lost if $T_f + D_{net} > D_{max}$); secondly, we give the network performance in terms of packet loss rate. Under these conditions, the simulation results do not depend on the particular video encoding scheme and video content; therefore, the trade-off between packetization and network delays can be studied independently of the specifics of the video traffic.

We follow here the same treatment done in [3], and the results given in this section have previously been presented in [30]. We first characterize the trade-off between the packet size and the network delay, and show that there exists an optimum packet size that minimizes the delay for a given packet loss rate. Furthermore, we show that the choice of

the packet size affects the network performance significantly. We then give the maximum number of streams supportable given V, D_{max}, and a maximum tolerable packet loss rate L_{max}. We assume that for a given run of the simulation, all streams have the same rate, packet size, delay and loss requirements. We simulate the network for a fixed period of simulation time, and repeat this until the desired confidence levels are reached (i.e., within $\pm 10\%$ with a probability of 90%).

We now show the existence of an optimum packet size. Let k denote the number of packets sampled in the simulation, and d_i $(i = 1, 2, \ldots, k)$ be the delay incurred by the i'th packet (i.e., $d_i = T_f + D_{net}(i)$). Let $d_1^{'} < d_2^{'} < \ldots < d_k^{'}$ be the ordered sequence of delay samples. We let $\max\{d|L_{max}\} \stackrel{\triangle}{=} d^{'}_{k(1-L_{max})}$, that is, the value of delay which is exceeded by exactly a fraction L_{max} of all samples. We can similarly define $\max\{D_{net}|L_{max}\}$. Clearly, $\max\{d|L_{max}\}=T_f+\max\{D_{net}|L_{max}\}$. When P_v is large, T_f is proportionally large. On the other hand, as P_v gets smaller, the number of packets contending increases, causing $\max\{D_{net}|L_{max}\}$ to increase. The combination of these two effects lead to an optimum packet size for which $\max\{d|L_{max}\}$ is minimum. This effect is illustrated in Figure 8 for V=384 kb/s. The solid lines show $\max\{d|L_{max}\}$ versus P_v, for L_{max}=0.001, and various values of N_v. The packet formation time, T_f, versus P_v is shown as a dotted line. The figure shows that given V, D_{max} and L_{max}, there exists a unique optimum value of P_v (denoted by P_{opt}) and a maximum value of N_v (denoted by N_{max}) which satisfy the constraint $\max\{d|L_{max}\} \leq D_{max}$.

Note that this definition of optimum packet size implies minimizing $\max\{d\}$ given that the stream experiences a loss rate equal to L_{max}. An alternative definition would be to choose the packet size that minimizes the loss rate L given that $\max\{d\}=D_{max}$. This alternative definition seems more appealing from a system designer's point of view, since the playback buffer at the receiving station is set such that packets exceeding D_{max} are discarded. However, we have determined that the optimum packet size is the same with either definition in all the cases that we have considered. (As a representative case, see Figure 9, where we plot $L|(\max\{d\}=D_{max})$ versus P_v for V=384 Kb/s, and compare it with Figure 8.)

In Figure 10, we plot the maximum number of streams supportable versus stream packet size, for W=10 Mb/s, V=384 kb/s, L_{max}=0.001, and D_{max}={20,100}ms. The figure illustrates that the choice of packet size has an important effect on the maximum

number of streams supportable by the network. Also, as we can see from the figure, the maximum number of streams is achieved over a range of packet sizes. The range of those feasible packet sizes are in some cases limited by the maximum allowable Ethernet packet size, particularly for larger values of D_{max}.

In Table 1, we show P_{opt} for $V = \{64, 384, 1536\}$kb/s, D_{max}=$\{20,100\}$ms, and L_{max}=$\{0.01$ %, 0.1%,1%$\}$. Observe from the table that P_{opt} increases with increasing stream rate. The reason is clear; with higher rate streams, a given packet formation time is attained by using a larger packet size. This allows the higher rate streams to use larger packets, which are handled more efficiently by the CSMA/CD protocol. Also note that for a small D_{max}, small packets need to be used due to the limit imposed on the packet formation time. As D_{max} increases, larger packets can be used, which are more efficiently handled by the CSMA/CD protocol. Thus, for a given stream rate, P_{opt} increases with increasing D_{max}.

In Table 2 we show N_{max} (and the corresponding maximum achievable network utilization in parenthesis) for D_{max}=$\{20,100\}$ms, L_{max}=$\{0.01\%,0.1\%, 1\%\}$, and V=$\{64,384,1536\}$ kb/s. As shown in the table, Ethernets can adequately support audio and video streams when the delay and loss constraints are within the ranges given above. It is also clear that more streams can be supported when the delay and/or loss constraints are more relaxed.

Recall that higher rate streams have a larger optimum packet size. Thus, they achieve the same network utilization with fewer packets. Also, with higher rate streams, there are fewer stations contending over the network for the same load, reducing the likelihood of contention and the average time to resolve contentions. As a result of these two effects, the CSMA/CD protocol operates more efficiently at high stream rates. Thus, at higher stream rates, even though fewer users are supported, more of the network bandwidth is utilized for useful traffic. As shown in Table 2, this is especially apparent for small D_{max}, because the low rate streams are forced to use very small packets.

We have also simulated a 100Base-T segment, and obtained similar results to those found for 10 Mb/s Ethernets. In Table 3, we show P_{opt} for V=$\{384,1536\}$kb/s, D_{max}=$\{20,$ $100\}$ ms, and L_{max}=$\{0.01\%,0.1\%, 1\%\}$ when there are only streams on the network. Comparing Table 3 with Table 1, we observe that the optimum packet sizes for 100Base-T are about the same as those for 10Base-T for given V, D_{max}, and L_{max}.

In Table 4, we show N_{max} (and the corresponding maximum achievable network utilization in parenthesis) for D_{max}=$\{20,100\}$ ms, L_{max}=$\{0.01\%,0.1\%, 1\%\}$, and V=$\{384,1536\}$

kb/s. By comparing this table with Table 2, we observe that 100Base-T can support about 10 times as many streams as in 10Base-T. This result does not immediately follow from the fact that 100Base-T has a 10 times higher transmission speed. There are several competing factors, some working for the advantage of 100Base-T, and some to its disadvantage. First of all, 100Base-T has a 10 times smaller slot size compared to 10Base-T. This works to the advantage of 100Base-T, since for a given D'_{max}, 10 times more slots are available in 100Base-T for a packet to resolve contention and be transmitted successfully. On the other hand, in 100Base-T, since packets are transmitted 10 times faster, the parameter a (i.e., propagation delay divided by the packet transmission time) is 10 times larger compared to that in 10Base-T for a given network span, and a given packet size. In CSMA/CD networks, performance degrades as a increases. Moreover, to achieve the same network utilization, 10 times more streams must exist in 100Base-T. For a given traffic load, the performance of a CSMA/CD network decreases as the number of stations which generate that load increases.

Note also that for 10Base-T, the delay that a packet experiences in the MAC layer after exceeding the maximum number of collisions is around 300-400 ms, while in 100Base-T, that delay is around 30-40 ms due to the 10 times smaller slot size of 100Base-T. Therefore, for 100Base-T, when D_{max}=100 ms, it is very useful to resubmit of a packet to the MAC layer when it exceeds the maximum number of collisions. Indeed, we have also simulated the case when the transport layer did not resubmit after excessive collisions, and seen that for W=100 Mb/s and D_{max}=100 ms, about 10% fewer streams could be supported for a given D_{max}.

The results given in this section are useful in understanding the trade-off between the packet formation time and network delays. However, as stated in the introduction, cell loss rate is not a good measure of video quality degradation. First, it does not translate easily to what a viewer perceives as a result of the information loss. Secondly, due to the dependence that exists in the video encoded bit stream, the loss of information contained in a single cell has impact that extends beyond just that information; conversely, given that cell loss is often bursty, the loss of several cells may account for the same perceived degradation as that due to the loss of a subset of these cells. Thus, in the next section, we introduce more accurate measures of quality degradation due to packet loss based on glitch statistics.

5 Glitching as a Measure of Video Quality Degradation due to Packet Loss

In this section, we introduce several glitch statistics as measures of video quality degradation due to packet loss, and investigate the effect of video encoding, video content, traffic load, and end-to-end delay constraint on the glitch statistics. We begin by showing that packet losses indeed occur in bursts. Furthermore, the burstiness of packet loss depends on several factors, such as the traffic load, end-to-end delay constraint, etc. We then characterize the discontinuities in the displayed video that packet loss causes. As a result of this characterization, we develop accurate and relevant measures, corresponding to the discontinuities that the viewer sees as a result of information loss.

In order to determine the statistics of discontinuities in the displayed video, the dependencies that exist in the encoded video bit stream must be considered (those dependencies differ from one compression scheme and its syntax to another). Accordingly, here we have simulated the network with video traffic obtained from encoding real video sequences, recording the lost packets and the macroblocks contained in them. The particular results given in this section are obtained by using the same video sequence to generate all streams for a given run of simulation. We use random starting times for each stream so as to decrease the correlations between the streams. We repeat the runs multiple times (with different starting times in each run), until the desired confidence levels are reached (again, within $\pm 10\%$ with 90% probability).

We have simulated the network with video traffic obtained from encoding several video sequences according to the H.261 and MPEG-1 standards. In particular, we use 3 video sequences, each of them about 1-minute long. The first one is 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. Given the small amount of motion, this sequence can be encoded at relatively low bit rates without incurring significant quality degradation. The second sequence is from the motion-picture "Star Trek VI: The Undiscovered Country"; it contains a combination of fast action scenes and other, slower-moving scenes. This sequence requires somewhat more bits to be encoded to achieve acceptable quality. The third sequence contains 3 different commercial advertisements; of those, the first two have very fast movement, and contain animated scenes, which are complex to

encode (i.e., they require more bits to be encoded for a given level of quality). For all three sequences, the frame format used is SIF (i.e., 352x240 pixels), and the frame rate is 30 fps.

5.1 Clustering of Packet Loss

Typically, the effect of multiple lost packets in a cluster is perceived as a single discontinuity in the displayed video. Therefore, compared to the packet loss rate, a more refined measure of video quality degradations could be the rate of packet loss clusters. In Figure 11, we show for 10Base-T a typical trace of the packets received and lost for one source-destination pair for V=384 kb/s, N_v=18, T_f=40 ms, D_{max}=100 ms. (Note that in this case, the packet loss rate is 0.14%.) It is clear that the packet losses occur in clusters of several successive packets. To further quantify the clustering of packets, consider Figure 12, where we show the histogram of the number of successive packets lost for the same simulation scenario. As the figure indicates, more than 65% of all the lost packets are in fact clustered.

Note that, the statistics of packet loss clustering potentially depends on the traffic load, and D_{max}. For example, if we consider the same scenario as above, but relax D_{max} to 500 ms, then 96% of all packet losses occur as single ones. This is because at such relaxed D_{max} values, packet loss becomes mainly due to exceeding the maximum number of collisions. Therefore, those packets that are queued behind a lost packet still have about the same chance as any other packet to get through, since the delay due to queueing becomes largely irrelevant. Therefore, in general, packet loss rate does not have a one-to-one correspondence with the frequency of packet loss clusters. As for the traffic load, consider again the case where V=384 kb/s, D_{max}=100 ms, and T_f=40 ms, but this time for N_v=15 and 20. (For these N_v values, the packet loss rate is 3×10^{-5} and 9.4×10^{-3}, respectively.) In Figure 14, the histograms of number of successive packets lost are shown for these two N_v values. From this figure and Figure 12, it is clear that the packet loss clustering is roughly independent of the network load; it is the frequency of packet loss clustering that increases as the network load is increased. We have made the same observation for other video contents, video encoding and encoder control schemes, and other D_{max} values, for both 10Base-T and 100Base-T networks. (By contrast, for ATM networks the cell loss cluster statistics do depend on the load as shown below.)

Now consider an ATM multiplexer. Let us first examine how the cell loss rate varies with the traffic load. In Figure 15, we show the cell loss rate as a function of N_v for

various values of M for the *commercials* sequence, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, N_v=65, D_{max}=25 ms, W=100 Mb/s. The first observation to be made is that cell loss rate increases exponentially, and at a very fast rate; indeed, it increases by three orders of magnitude by increasing N_v from 63 to 69. Therefore, one should operate just below the "knee" where cell losses start to occur, and achieve a good video quality as well as good network utilization. However, it is still of interest to study the pattern of cell losses, and to determine the effect of cell losses on the displayed video.

The second observation to be made regarding Figure 15 is that for all N_v values, the smallest cell loss rates occur for M=4500 cells, and the difference between the cell loss rates for M=4500 cells and $M = \infty$ is about a factor of two. M=4500 cells corresponds to a maximum delay of about 19 ms in the multiplexer buffer. Considering that the cell formation time is typically around 1 ms for this sequence (and the time for encoding and decoding is assumed to be 4.4 ms), at this value of buffer size only those cells that would have been received late at the receiver are dropped. For a smaller buffer size, some cells are dropped unnecessarily, and for a larger buffer size, some cells are included in the buffer despite them having exceeded their delay constraints; those cells in turn cause other cells to be delayed excessively, and thus get dropped at the receiver. We have observed for all D_{max} values considered that the optimum buffer size which minimizes the cell loss rate is roughly equal to $W \times D_{max}$ minus a few hundred cells.

Now let us consider the clustering of cell loss in an ATM network. It has been observed before that the cell losses in an ATM multiplexer carrying video traffic occur in clusters [13]. Typically, the effect of multiple lost cells in a cluster is perceived as a single discontinuity in the displayed video. Therefore, compared to the cell loss rate, a more refined measure of video quality degradations could be the rate of cell loss clusters. In Figure 16, we show for an ATM multiplexer a typical trace of the multiplexer buffer occupancy for $\hat{s}_{target}$=4.5, D_{max}=25 ms, and W=100 Mb/s; part (a) of the figure is for N_v=65, and part (b) is for N_v=67. (Note that the respective cell loss rates are 1.8×10^{-5} and 5.5×10^{-4}.) It is clear that the cell losses occur in clusters which last several tens of milliseconds. Those clusters are separated by tens or hundreds of seconds. Furthermore, within a cluster, not all the cells are lost; this is different from the Ethernet, where the cell losses within a cluster usually occur in succession. Note that the cluster for N_v=67 is longer than the one for N_v=65. This is in general true, and it is as expected since increasing the offered load to

the network prolongs the congestion period. For these two scenarios, we have determined that the time between each subsequent cell loss per source occurs within 10 ms for 95% of the cell losses. For the remaining 5%, the time between subsequent cell losses is in a range from a few seconds to several hundreds of seconds. Therefore, an appropriate definition of a cell loss cluster would be a group of lost cells where the time between subsequently lost cells is less than 10 ms.

The above discussion suggests that packet/cell loss cluster statistics could be used as a measure of video quality degradation. However, it is also interesting to know the amount of information loss in each discontinuity. Therefore, we map the packet losses into information loss in the video bit stream, and in the following subsection define several quantities to characterize the perceived effect of such information loss.

5.2 Definition of Glitch Statistics and Some Illustrative Examples

We first define a *glitch* as an occurence which begins when a portion of a frame is unavailable while its preceding frame is fully displayed, and continues as long as each subsequent frame contains an unavailable portion. Note that the "subsequent" frames here refer to the transmission order of the frames, which in case of MPEG may be different from the display order due to B frames.

Note that if the "lost" cells are in fact late arrivals to the receiver, they can be used to recover from the glitch at an earlier time. Clearly, this requires that the decoder is fast enough to catch-up with the extra information that it has to process. We do not consider such early recovery schemes in this paper, and assume that the only means of recovery from an information loss is intraframe coding. We also do not consider any forward error correction schemes appropriate for video transmission, such as PET [31, 32]. Neither do we consider any error concealment schemes, and assume that the unavailable portions of the received video are simply shown as black regions. Consideration of such improvements is for further study.

The information loss in the video bit stream resulting from cell loss depends in general on the interframe/intraframe coding patterns used by the video encoding scheme. For example, in H.261, frames are divided into a number of Group of Blocks (GOBs); (a frame

consists of 3, 10, or 12 GOBs, depending on the frame format). Typically, one GOB in each frame is intracoded, and the intracoded GOB is rotated cyclically among all the GOBs from frame to frame. In this case, when a glitch which affects only one GOB occurs, it will be recovered in the frame where the affected GOB is intracoded. Therefore, assuming that no other loss occurs in the middle of the glitch, its duration will be anywhere from one frame to the number of GOBs in a frame. Similarly, when a glitch which affects multiple GOBs occurs, it will be recovered one GOB at a time as the affected GOBs are intracoded in successive frames. Taken to the extreme, if the glitch initially affects an entire frame, it will take at least as many frames to fully recover from it as the number of GOBs in a frame. By contrast, in MPEG, when a glitch begins in an I or P frame, it affects all the B and P frames dependent on that frame. When a glitch begins in a B frame, it does not propagate; thus the duration of such a glitch is always 1 frame.

The statistics of the information loss also depend on the network load, the encoder control scheme, video content, and the end-to-end delay constraint. We now define several quantities that characterize the information loss in a glitch, and show several example cases taken from our simulations to demonstrate how the factors described here affect the glitch statistics. In the next subsection, we investigate in more detail the effect of these factors.

In Figure 17, we show a typical glitch for H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, N_v=65, M=4500 cells. In the figure, the black regions correspond to macroblocks that are contained in the lost cell, the gray regions correspond to macroblocks that depend on the lost macroblocks, and the white regions correspond to the macroblocks that are received and available to be displayed. In this particular glitch, information in 13 cells are lost. The first quantity of interest we define is the number of macroblocks that are contained in the lost cells that cause the glitch (denoted by L_I). The larger this quantity, the greater the initial area lost in a frame at the beginning of a glitch. For the example glitch considered here, L_I=80 macroblocks. The second quantity of interest is the *glitch duration* d, defined as the number of frames that are part of the glitch. In this particular case, the glitch duration is 10 frames. Two other quantities of interest are the *maximum and average number of undisplayed macroblocks per frame* (denoted by A_{max} and A_{av}, respectively). These two quantities indicate how large a display area the glitch covers. In this example, $A_{max} = 183$ macroblocks, and $A_{av} = 76$ macroblocks. The final quantity of interest is the *total number of undisplayed macroblocks in a glitch* L_{tot}. This quantity is particularly interesting since it

combines the duration and area information in a simple way; therefore, if a single number is used to quantify the perceived effect of a glitch, L_{tot} is a good candidate. In this example, $L_{tot} = 760$ macroblocks. In addition to these quantities that are defined per glitch, another quantity of interest is the *glitch rate* g, which is defined as the number of glitches that occur per unit time. For this particular simulation scenario, the glitch rate is 0.17 per minute.

We have observed that in ATM, when the network load is increased, the correlation between cell losses increases; therefore, the glitches become longer and affect larger portions of the frames. As an example to the increased network load, consider the same scenario as above, but this time with N_v=67. In Figure 18, we show a typical glitch for that case. In this particular glitch, information in 131 cells is lost. Accordingly, the glitch affects a larger area and lasts longer. In particular, L_I=576 macroblocks, d=16 frames, $A_{max} = 325$ macroblocks, $A_{av} = 107$ macroblocks, $L_{tot} = 1709$ macroblocks. Similarly, in Figure 19, we show a representative example glitch for N_v=69. For this glitch, L_I=846, A_{max}=327, A_{av}=188, and L_{tot}=2448 macroblocks, and d=13 frames.

Now consider some example scenarios for Ethernet. In Figure 20, we show a typical glitch for H.261, V=384 kb/s, D_{max}=100 ms, $T_f = 40\ ms$, N_v=18. In this particular case, information in four successive packets is lost. Each packet contains a region roughly corresponding to a full frame. The first lost packet causes two-fifth of frame 1 to be unavailable. Packet losses continue in frames 2 to 4. Starting with frame 4, the displaying is recovered GOB by GOB in every subsequent frame until frame 12, when the recovery is completed. For this glitch, L_I=962 macroblocks, which is roughly the area of 3 frames. The glitch duration is 12 frames, $A_{max} = 330$ macroblocks, $A_{av} = 195$ macroblocks, and $L_{tot} = 2345$ macroblocks.

For the above simulation scenario, in Figure 21 we show the histogram of L_I. About 30% of the glitches $L_I \geq 330$ macroblocks, or a full frame. In all those cases an entire frame becomes unavailable in the beginning of a glitch. For reasons given above, the duration of a glitch in H.261 is correlated with L_I. In Figure 22, we show the histogram of the glitch durations. About 80% of the glitches occur with a duration from 10 to 15 frames, as expected from the L_I statistics. In Figure 23, we show the histogram of A_{max}. As the figure indicates, the maximum area lost in a frame is usually quite large for this scenario, which is again as expected. In Figure 24, we show the histogram of L_{tot}. L_{tot} is on average equal to 1318 macroblocks, and it is mostly distributed between 0 and 3000. Note that

for the same scenario, we have also considered dropping entire packets both at the sender and the receiver whenever the first macroblock in the packet exceeds its delay bound. In Figure 25, we show the histogram of L_{tot} for that case. It is clear that the L_{tot} values are much larger in this case; indeed, the average L_{tot} is now equal to about 2000 macroblocks. Therefore, in this scenario, dropping of only those macroblocks that exceed their delay bound as opposed to dropping entire packets results in an improvement of about 35% in terms of average L_{tot}. We have made the same comparison for various other scenarios as well, and observed that the improvement in general ranges from 20% to 50%.

The above results indicate that the perceived effect of the glitches in this scenario would be quite severe, even when only those macroblocks that exceed their delay bound are dropped. To confirm this further, we apply the ITS video quality measure $\hat{s}$. In Figure 26, we show the histogram of $\hat{s}$ values, where $\hat{s}$ is measured for every glitch, including several frames prior to and after the occurence of the glitch such that the total period over which $\hat{s}$ is measured is 30 frames (i.e., one second) for each glitch. It is interesting to note that all the glitches cause a quality degradation of 2.8 impairment units or less. Therefore, all the glitches in this case would be perceived as being annoying.

In the scenario considered above, 98% of all packets have a size of 1500 bytes, and the average frame size is 1600 bytes. We now consider another scenario where V=1536 kb/s, D_{max}=25 ms, and N_v=4. In this case, T_f=8 ms is used in order not to incur too large a packet formation time compared to D_{max}. The resulting packet sizes are on average 1200 bytes, and the average frame size is 6400 bytes. Therefore, the packet size to frame size ratio is about 4 times smaller in this scenario compared to the previous one. (Therefore, a packet typically contains 2–3 GOBs.) Therefore, all the glitch quantities we have defined are much smaller compared to the previous case. For example, the total number of macroblocks affected by a glitch is on average 327 macroblocks, which is about 1/4'th of the previous case. Therefore, the glitches in this scenario would be expected to have a less severe effect compared to the previous scenario. In Figure 27, we show a typical glitch for this scenario, (In this case, the packet loss rate is about 0.02%, and the average time between packet loss clusters is about 50 seconds.) In the particular example shown, a single packet is lost, which contains information for 2 GOBs. One of these GOBs happens to be the last GOB in frame 1, and the other one is the first GOB in frame 2. Then in frame 4, one of the GOBs is recovered, and in frame 5, the other one is also recovered, ending the glitch. This

figure suggests that one could also define the duration of a glitch as the maximum number of frames that any macroblock at a particular location is not displayed. For this particular example, this quantity represents more accurately the perceived duration of a glitch (i.e., 3 frames instead of 4); however, for simplicity, we use the original definition given above. As for the other glitch quantities in this example, L_I=22 macroblocks, A_{max}=40 macroblocks, A_{av}=30 macroblocks, and L_{tot}=120 macroblocks. In Figure 28, we show the histogram of the $\hat{s}$ values for the same scenario. What is interesting is that even though the packet losses cover a much smaller area for this case compared to the previous one, the $\hat{s}$ statistics are only marginally better, and the quality values are always less than or equal to 3.5 units. Therefore, even for this case where the lost region covers a relatively small portion of the frame, all the losses have an important impact.

We now show three examples for MPEG-encoded video transmission over Ethernet. The examples are taken from the scenario where the video is encoded at V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18, W=10 Mb/s. (For this case, the packet loss rate is about 0.17%, and the average time between packet loss clusters is about 42 seconds.) The Group Of Pictures (GOP) pattern is IBBPBBPBBPBBI....

The first example glitch for this scenario is shown in Figure 29. (The frames are shown in the display order.) As the figure indicates, a packet loss occurs in the middle of an I frame. The glitch affects the two preceding B frames, as well as all the succeeding B and P frames in the same GOP, until finally it is recovered by the next I frame. While the shape of the affected region somewhat changes from frame to frame due to nonzero motion vectors, the overall area affected per frame remains roughly the same. The glitch quantities in this case are d=14 frames, L_I=67, A_{max}=80, A_{av}=70, and L_{tot}=991 macroblocks. Note that the values of A_{max} and A_{av} are very close in this case, indicating that the glitch area does not change very much from one frame to another. This is generally true for all the glitches which occured for this scenario, indicating that the motion vectors in this sequence are not very large.

In Figure 30, another example glitch for MPEG (with the same simulation scenario) is shown. In this case, two successive packets are lost, which contain regions from frames 1, 2, and 6. Note that due to having B frames, the frames shown are in fact transmitted in the order 3,1,2,6,4,5,...; therefore, when they are displayed, the first two frames show discontinuities, then a full frame is displayed, and then another 8 frames have discontinuities.

Having such a fully displayed frame in the middle of a glitch is quite a common case in this scenario, taking place in about 30% of all the glitches.

Now consider a third example glitch for MPEG from the same scenario, which is shown in Figure 31. In this particular example, the packet loss only affects a single B frame. Since no frames use a B frame as reference, the glitch in this case is confined only to this particular frame.

Finally, note that sometimes a packet loss may occur in the middle of a glitch, and depending on where it occurs, it may prolong the glitch. An example to this is shown in Figure 32 from the first scenario given in this section (i.e., H.261, V=384 kb/s, D_{max}100 ms, N_v=18). First, some packet losses occur during the first 3 frames of the glitch; then, the gradual recovery from the glitch happens until frame 10, and then from frame 11 to 18, another series of packet losses occur, from where on the glitch is gradually recovered until frame 27, where the recovery is complete.

5.3 Effect of Various Factors on Glitch Statistics

Having defined the glitch quantities of interest and shown several example scenarios, we now investigate the effect of various factors on the glitch statistics. The particular factors we focus on are the video encoder scheme, the video encoder control scheme, the traffic load (i.e., the number of video sources multiplexed), video content, D_{max}, and for ATM multiplexers, the switch output buffer size M.

First consider Ethernets. In Table 5, we show for 100Base-T the packet loss rate L, glitch rate g, and the average values of d, L_{tot}, and A_{max} (denoted by $\overline{d}$, $\overline{A_{max}}$, and $\overline{L_{tot}}$, respectively) for various video encoder schemes, encoder control schemes, contents, network loads, and D_{max} values. In the first two rows, we have the *commercials* sequence, H.261 CQ-VBR encoded, and D_{max}=25 ms. The two rows correspond to N_v=44, and 51. For those N_v values, the glitch rates are 0.08, and 0.92, respectively. However, in both cases, the per-glitch statistics are about the same. We have observed that in Ethernets, in general, the per-glitch statistics are independent of the network load as long as the glitch rate is not more than a few glitches per minute. As for the effect of D_{max}, the third row indicates that when D_{max} is increased to 500 ms, the per-glitch statistics increase very significantly. This is because at such large values of D_{max}, the packets that are dropped at the MAC layer due to exceeding the maximum number of collisions are resubmitted several times;

this may lead to prolonged periods of congestion due to the backlog caused at the stations. As for the video encoder control scheme, the encoding scheme, and the video content, we have found that the per-glitch statistics are fairly independent of those factors, as indicated by the results shown in Table 5. We have found similar results also for 10Base-T. Since the per-glitch statistics for Ethernet do not depend on the network load, we can use the glitch rate as the main measure of network performance.

Now consider an ATM multiplexer. We first show how the selection of the multiplexer buffer size M affects the glitches. In Figure 33 (a), we plot the glitch rate as a function of M for the *commercials* sequence, H.261 encoded using the CQ-VBR scheme with $\hat{s}_{target}$=4.5, N_v=65, and D_{max}=25 ms. As the figure indicates, the glitch rate decreases as M is increased when M is less than about 4500 cells (recall that this is the buffer size that minimizes the cell loss rate). Beyond that point, the glitch rate remains fairly constant. On the other hand, consider Figure 33 (b), where we plot the average L_I (denoted as $\overline{L_I}$) as a function of M; when M is increased beyond 4500 cells, $\overline{L_I}$ increases sharply. This causes both the glitch duration and the total number of undisplayed macroblocks per glitch to increase. Therefore, in order to obtain the lowest glitch rate, and the smallest per glitch quantities, the output buffer size M should be chosen such that the maximum delay incurred in the buffer is about D_{max} minus 5–6 ms. In the remainder of the paper, the results are given for that choice of M.

We now investigate for an ATM multiplexer the effect of video encoding scheme, video encoder control scheme, content, network load, and D_{max} on the glitch statistics. First, consider a baseline case, consisting of H.261, CQ-VBR, $\hat{s}_{target}$=4.5, *commercials* sequence, N_v=67, D_{max}=25 ms. In Figure 34, we show the histogram of L_I for that scenario. It is clear that the number of macroblocks that are contained in the lost cells can be as large as 1500-2000 per glitch; this indicates that many glitches would last longer than 10 frames (which is the number of frames at which loss in a single frame can be fully recovered). This is indeed confirmed in Figure 35, where the histogram of the glitch duration is shown; about 70% of all glitches have a duration greater than 10 frames. In Figure 36, we show the histogram of A_{max} for the same scenario. It is clear that about 65% of all glitches have an A_{max} of about 90% of a frame. Finally, we show the histogram for L_{tot} in Figure 37. The figure indicates that the total number of macroblocks not displayed due to a glitch may be up to 3500 in this scenario.

Having studied the baseline case, let us compare it with other cases by changing one parameter at a time. In Table 6, we show the cell loss rate L, glitch rate g, and the average values of d, L_{tot}, and A_{max} (denoted by $\overline{d}$, $\overline{A_{max}}$, and $\overline{L_{tot}}$, respectively) for various video encoder schemes, encoder control schemes, contents, network loads, and D_{max} values. In the first three rows, we show the effect of changing the network load from N_v=65 to N_v=69. It is clear that increasing the load by such a small amount not only increases the cell loss rate and glitch rate very sharply in this range, but also increases the other glitch quantities as well.

We then investigate the effect of changing D_{max}. In row 4 of the same table, we set D_{max}=500 ms. Note that for this value, N_v=67 results in no glitching. Therefore, we also increase the network load to N_v=75. It is interesting to note that the cell loss rate here is almost an order of magnitude greater compared to the baseline, and yet the glitch rate is smaller (0.58 per minute as compared to 0.90). Also, $\overline{d}$ and $\overline{L_{tot}}$ are much larger in this case. These differences are because the cell losses in this case tend to occur in larger clusters. Indeed, when we have a large D_{max}, we can put several more streams on the network without causing cell loss. But when a cell loss occurs, the cells arriving shortly afterwards have a greater likelihood of being lost because of the larger traffic load offered. Therefore, the cell losses tend to be more correlated. Occurence of such long glitches suggests that glitch rate by itself may not be a good measure of the quality degradation; one could instead consider weighting glitches according to, say, L_{tot}.

Now consider using Open-Loop VBR. In order to achieve the same minimum quality $\hat{s}_{min}$ at all times, this sequence has to be encoded at q_0=8 (found by iteration). At that q_0 value, the average rate of the OL-VBR encoded sequence is 1.64 Mb/s. Thus, in order to have the same network load as in the baseline case, here we use N_v=43. (See the fifth row of Table 6.) It is interesting to note that the cell loss rate and all the glitch statistics are quite similar in this case to the CQ-VBR. Therefore, the OL-VBR and CQ-VBR schemes give similar results, except for the fact that the number of streams supportable is less for the OL-VBR due to the larger data rate required to achieve the same minimum quality level.

As for comparing MPEG-1 and H.261, in the sixth row we give the glitch statistics for MPEG-1, using N_v=48, which results in about the same network load as in the baseline case. (Note that here D_{max} is chosen to be 125 ms, given the 100 ms additional delay

introduced by the use of B frames.) It is interesting to note that here the glitch rate is somewhat larger compared to the baseline case; however, $\overline{d}$, $\overline{A_{max}}$, and $\overline{L_{tot}}$ are somewhat smaller. This is mainly because some of the glitches are confined only to the B frames.

Finally, we consider using a different video content, namely, the *videoconferencing* sequence. In the seventh row of Table 6, we show the glitch quantities for that sequence. We have first tried setting the average network load equal to the baseline case. However, the *videoconferencing* sequence is less variable compared to the *commercials* sequence; as a result, there are no cell losses for that case. So, we have increased the network load until the knee of the cell loss rate occurs. The knee is around N_v=193–195 (i.e., a network utilization of about 85 Mb/s). We show in the table the packet loss rate and glitch statistics for N_v=194 and 195. For N_v=194, the cell loss rate is 4.0×10^{-4}, which is similar to the baseline case; however, the glitch rate is only 0.20 per minute. The three per glitch quantities are in this case similar to the baseline. For N_v=195, the cell loss rate is 1.9×10^{-3}, and the glitch rate is about 0.99 per minute. The three per glitch quantities are all greater than those for the baseline. This indicates that the cell losses are more clustered here compared to the baseline case. It is also interesting to note that the cell loss rate increases very significantly for such a small increase in the network load.

These results indicate that for ATM networks, the glitch patterns depend significantly on the network load, D_{max}, and video encoding scheme; to a lesser degree, they also depend on the video content and encoder control scheme. This is in contrast with an Ethernet segment, where there is almost no dependency on the network load (within reasonable limits), and the dependency on D_{max} is less significant. However, recall also that the glitch rate increases very rapidly in ATM as the network load is increased beyond a certain point (i.e., there is a very sharp knee). Thus, what is most important is to determine where the knee occurs, and to operate just below it. We can therefore still use glitch rate as the primary performance measure, as the per-glitch statistics are of secondary importance given the existence of such a sharp knee.

6 Performance of Ethernets and ATM Multiplexers Carrying Constant Bit Rate and Variable Bit Rate Video Traffic

Having characterized the glitch characteristics, and established that glitch rate g is an appropriate measure of network performance carrying video traffic, in this section we address the performance of CBR and VBR video transmission over an Ethernet segment and an ATM multiplexer. We first discuss the results for Ethernet, examining the effect of the packetization scheme, and comparing the number of streams supportable for CBR and VBR video streams while maintaining the same level of quality. We then discuss similar results for ATM, and compare the number of streams supportable in the two types of networks.

6.1 Ethernet

6.1.1 CBR Video Traffic on an Ethernet Segment

Here, we consider video rates of V=1536 kb/s and V=384 kb/s. As far as the choice of B_{max} is concerned, we have determined in [25] that as the B_{max} is increased, the encoded video quality first increases, and then reaches a plateau. Furthermore, the maximum and average values of delay in the rate control buffer increase linearly with B_{max}. Therefore, a reasonable choice of B_{max} is the "knee" point where the quality just reaches its plateau. Those B_{max} values for the three video sequences considered are shown in Table 7. The results presented in the remainder of this paper are for those values of B_{max} unless explicitly stated otherwise. Note that it is not possible to encode the *commercials* sequence at V values smaller than about 1 Mb/s without incurring severe quality degradation even for very large B_{max} values. Therefore, we only consider V=1536 kb/s for that sequence.

In Figure 38, we show the glitch rate as a function of the packet size P_v for the *video-conferencing* sequence, for V=384 kb/s, D_{max}=100 ms, and CSRP. As shown in the figure, around the optimum value of P_v, there is a flat region where g is fairly insensitive to P_v. The reason is that when the packet loss rate increases, the number of consecutive packets lost also increases; these consecutive lost packets all contribute to the same glitch. As a result of this effect, the rate of increase in the glitch rate is less than the underlying rate

of increase in the packet loss rate. Similar graphs are shown for VSRP in Figures 39 and 40 for D_{max}=25 ms and 100 ms, respectively. These figures show a similar flat region as well. As a result, it is clear that the choice of packet size does not need to be very precise. A guideline that can be given based on our observations is to use $T_f = 0.4D_{max}$ for VSRP. For CSRP, if the maximum delay in the rate control buffer cannot be estimated, then a conservative choice is

$$T_f = \min\{0.4(D_{max} - B_{max}/V), (1500\ bytes/V)\}$$

for $D_{max} > B_{max}/V$. If the maximum delay in the rate control buffer is known or estimated, then instead of B_{max}/V, the value of that delay can be substituted in the above equation. We have observed that by using these guidelines typically the same number of streams as with the true optimum T_f can be supported over the network.

Although there is not a one-to-one mapping between the packet loss rate and the glitch rate, we have seen that at the optimum T_f, packet loss rates ranging from 10^{-4} to 10^{-3} result in glitch rates on the order of 1 per minute. Similarly, packet loss rates ranging from 10^{-5} to 10^{-4} result in glitch rates on the order of 1 every 10 minutes.

Let g_{max} be the maximum tolerable glitch rate. In Figure 41, we plot the maximum number of streams supportable on a 10Base-T segment as a function of the stream packet size, for the *videoconferencing* sequence, at V=384 kb/s, B_{max}=38.4 kbits, g_{max}=0.1 per min, and D_{max}=$\{60,100\}$ms. Similarly to section 4, the figure indicates that the choice of T_f has an important effect on the maximum number of streams supportable by the network. In Figure 42, we plot the counterpart of Figure 41 for VSRP, and for D_{max}=$\{25,50,100\}$ms. As this figure indicates, for given D_{max} and g_{max}, more streams can be supported using VSRP as compared to CSRP; likewise, the range of T_f values for which the maximum is attained is greater using VSRP.

Now let us consider the effect of B_{max} on the glitch rate. In Figure 43, we plot g as a function of N_v for the *videoconferencing* sequence, V=384 kb/s, B_{max}=38.4 kbits, D_{max}=60 ms, CSRP, using T_f=3 ms at each point. As the figure indicates, there is a knee at N_v=13, beyond which the glitch rate increases very sharply. Furthermore, for this sequence and for D_{max}=60 ms, it was not possible to use a greater value of B_{max} without incurring a large rate of glitches, due to many packets exceeding D_{max} while still in the encoder's output buffer. In Figure 44, we again plot g as a function of N_v for the same sequence, at the same

rate and the same packetization scheme, but for D_{max}=100 ms, B_{max}=$\{$19.2,38.4,96$\}$kbits, and T_f=25 ms. As the figure indicates, here too the glitch rate depends on B_{max}. Now consider Figure 45, where we again plot g versus N_v for the *videoconferencing* sequence encoded at V=384 kb/s, for D_{max}=$\{$25,60,100$\}$ms, B_{max}=$\{$19.2,38.4,96$\}$kbits, but this time for VSRP. The figure indicates that with the VSRP scheme, the dependence on B_{max} is much smaller as compared to the CSRP scheme (in fact, it is almost negligible for the B_{max} values considered). This justifies our choice of B_{max} at the plateau of quality, since smaller B_{max} values would not improve the network performance, but reduce the encoded video quality. The figure also indicates that using VSRP, it is possible to transfer video even at very small D_{max} values such as 25 ms. This is especially important if the encoder and decoder are not streamlined as we have suggested, since their processing delay have to be subtracted from the end-to-end delay requirement to find the equivalent D_{max}.

In Table 8, we show the maximum number of streams supportable (N_{max}) and in parenthesis the corresponding maximum achievable network utilization for the three sequences, for D_{max}=$\{$25,60,100,250,500,600$\}$ms, g_{max}=$\{$0.1,1$\}$ per minute, V=$\{$384,1536$\}$ kb/s, and using the CSRP scheme, with the optimum T_f chosen for each operating point. As the table indicates, with this packetization scheme, N_{max} is dependent on the video sequence used, particularly when D_{max} is small. In particular, the *Star Trek* sequence at V=384 kb/s requires a relatively large B_{max} to be encoded with reasonable source quality; therefore in the CSRP scheme, end-to-end delays of 500 ms or less could not be achieved due to the delay in the encoder's output buffer. When D_{max} is greater than 600 ms, the number of streams supportable reaches a plateau for all three contents.

Table 9 is the counterpart of Table 8 for the VSRP scheme. Here, we observe that the dependency on the video content is much less significant. Moreover, at all points, the number of streams supportable is greater than or equal to its CSRP counterpart, the difference being larger for smaller D_{max}. Thus, VSRP scheme results in better performance as compared to CSRP.

Now compare VSRP with Dynamic Packetization. It is clear that the advantage of the DP method will be most apparent at low stream rates, and small D_{max}; (otherwise, packet sizes get limited by the maximum Ethernet packet size). Therefore, consider the case of CBR with V=384 kb/s, and D_{max}=25 ms. For that case, in Figure 46 we plot the glitch rate as a function of N_v for *videoconferencing*. The figure indicates that the DP scheme has

some small advantage over the VSRP scheme, especially as we get to greater N_v values, and therefore larger glitch rates. However, the maximum number of streams supportable given a maximum glitch rate of 0.1 per minute to 1 per minute differs only by 1 between the two packetization methods. For greater bit rates and greater D_{max} values, there would be even less of an improvement using the DP method. Therefore, we recommend the VSRP method to be used for its simplicity.

Now consider the performance of 100Base-T networks carrying video traffic. As far as the range of T_f values to attain the minimum glitch rate is concerned, the results were identical to 10Base-T. In Tables 10 and 11, we show the maximum number of streams supportable (N_{max}) and in parenthesis the corresponding maximum achievable network utilization for the three sequences, for D_{max}=$\{25,60,100,250\}$ms, g_{max}=$\{0.1,1\}$ per minute, V=$\{384,1536\}$ kb/s, with the optimum T_f chosen for each operating point. Table 10 is for CSRP, and Table 11 is for VSRP. As the tables indicate, for small D_{max}values, 100Base-T can support more than 10 times as many streams as 10Base-T, as a result of the 10 times smaller slot time, which allows the collisions to be resolved 10 times faster.

Finally, for comparison purposes consider that the encoder operates on a frame-by-frame basis. The bits corresponding to a frame are placed as a single unit in the encoder output buffer, from where they are retrieved immediately, placed into as many full size packets as needed, and queued for transmission. Since the encoder now postpones producing the first macroblock in a frame until all the macroblocks are ready, an additional 33 ms of end-to-end delay is introduced. In Table 12, we show N_{max} for different values of D_{max} for 100Base-T. Not surprisingly, the N_{max} values for this case and the streamlined operation case (with VSRP packetization) are about the same when D_{max} is greater by 33 ms in the frame-by-frame operation case.

6.1.2 VBR Video Traffic on Ethernet

Now consider that the sequences are CQ-VBR encoded. We use the VSRP scheme to packetize the CQ-VBR video data. For the CQ-VBR, we have found that the same packetization guideline as for the CBR VSRP scheme applies; that is, $T_f = 0.4D_{max}$ results in near-minimum glitch rates for a given N_v. In Table 13, we show for a 10Base-T segment the maximum number of streams supportable and the corresponding network utilization for CQ-VBR with the target quality values $\hat{s}_{target}$=4.0 and $\hat{s}_{target}$=4.5, for all three video

sequences, D_{max}={25,60,100,250}ms, g_{max}={0.1,1} per minute. The table indicates that for small D_{max} values, the number of CQ-VBR streams that can be multiplexed on a 10Base-T segment is fairly small, and the corresponding network utilization is fairly low (i.e., about 20–35%). For larger D_{max} values, the number of streams supportable increases, achieving network utilizations of about 50–60%. Table 14 is the counterpart of Table 13 for a 100Base-T segment. Here, even for small D_{max} values, the network utilization is fairly large compared to the 10Base-T segment. This is because a greater percentage of VBR streams can be multiplexed over a channel with a larger bandwidth.

It is also interesting to compare these results with the CBR results. In order to make a meaningful comparison, we consider the sequences to be encoded such that for both the CBR and CQ-VBR schemes the resulting video quality is always greater than or equal to a given minimum quality value $\hat{s}_{min}$. In Table 15, we show the resulting average rates for CQ-VBR, as well as the data rate for CBR when it achieves the same minimum level of quality $\hat{s}_{min}$ as the CQ-VBR scheme. Parts (a) and (b) of the table are for $\hat{s}_{target}$=4.0 and $\hat{s}_{target}$=4.5, respectively. The table indicates that for the *commercials* sequence, the CQ-VBR average rate is about one half of the CBR rate. For the *Star Trek* sequence and for $\hat{s}_{target}$=4.0, the ratio of CBR rate to the CQ-VBR average rate is about 1.5, and for the same sequence, for $\hat{s}_{target}$=4.5, the two rates are nearly the same. For the *videoconferencing* sequence, the data rates for CQ-VBR and CBR are nearly the same for both target quality values. The ratio of the average data rates for CQ-VBR and CBR represents the maximum amount of multiplexing gain that can be achieved when sending video over a network. The multiplexing gain over a network approaches this maximum for large network bandwidths and large D_{max} values. Given the numbers in Table 15, it can be expected that the greatest amount of gain in using CQ-VBR compared to CBR can be achieved for the *commercials* sequence, and almost no gain should be expected for the *videoconferencing* sequence.

Now consider Table 16, which is equivalent to Table 13, but for the corresponding CBR sequences (packetized using the VSRP scheme). It is interesting to note that the number of CQ-VBR streams supportable on a 10Base-T segment is smaller compared to its CBR counterpart for the *videoconferencing* and *Star Trek* sequences, and they are somewhat greater for the *commercials* sequence, especially for relatively large values of D_{max}. Now consider the 100Base-T. In Table 17, we show the maximum number of CBR streams (with equivalent quality to the CQ-VBR streams) supportable for the same cases as in Table 14.

It is interesting to note that for 100Base-T, up to twice as many CQ-VBR streams can be supported while maintaining the same minimum level of quality. In fact, roughly the same level of network utilization can be achieved by CQ-VBR streams as for CBR streams. Therefore, the theoretical multiplexing gain as given by the ratio of the CQ-VBR average rate to the CBR rate is achieved in 100Base-T.

6.2 ATM Multiplexer

6.2.1 CBR Video Traffic on an ATM Multiplexer

First, we consider transmitting CBR video traffic over an ATM network. As a preliminary step, consider that the CBR video is multiplexed over a circuit of bandwidth W, and as soon as a macroblock is generated, it is passed to the multiplexer buffer. We ignore any overhead due to framing. In this case, one possibility is to allocate each CBR stream a bandwidth equal to V; this would result in a delay in each source equal to the rate control buffer delay [25], but little or no delay would be incurred in the multiplexer buffer. Another alternative is to statistically multiplex the CBR streams by sending to the multiplexer all the bits corresponding to a macroblock as soon as they are produced by the encoder; the idea here is to reduce the end-to-end delay by allowing the multiplexer buffer to absorb the fluctuations in the instantaneous rate. We have shown in [25] that such fluctuations are of relatively small magnitude and short duration for CBR controlled video traffic, suggesting that statistically multiplexing CBR streams is useful even at relatively small values of W. 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 [25]. We have simulated several video sources using those sequences as the source, and choosing a different, random starting point in the sequence for each source in order to reduce correlations among them. We have repeated the simulation many times, 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 $\max_k\{D(W,k)\}$). In Figure 47, we show the histogram of $\max_k\{D(W,k)\}$ for both sequences, and for various values of W and N_v such that the network utilization is 96% in each case (i.e., $VN_v = 0.96W$). It is clear that even

at relatively small bandwidths, statistical multiplexing of CBR sources reduces the delay significantly.

For an ATM network where the bandwidth is considerably larger, and therefore many more streams can be multiplexed, the reduction of end-to-end delay would be expected to be even greater. Indeed, we have simulated statistically multiplexing several CBR sources by using the VRCF scheme using all three video contents, with data rates ranging from 270 kb/s to 2 Mb/s, rate control buffer sizes ranging from 38.4 kbits to 768 kb/s, and D_{max} values ranging from 25 ms to 500 ms. In all cases, we have found that the number of streams that can be multiplexed in the network without incurring any measurable cell loss is equal to $\lfloor \frac{48}{53} \frac{W}{V} \rfloor$ (where the term 48/53 is due to the cell header overhead). Clearly, with the CRCF scheme the same number of streams can be sent over the network, but if D_{max} is small, such as 25 ms, then a very small rate control buffer size must be chosen, which results in a significantly reduced source video quality [25]. Given these results, statistically multiplexing CBR streams is more advantageous compared to allocating bandwidth individually for them.

Now consider comparing the performance of Ethernets and ATM networks carrying CBR traffic. As stated above, a 100Mb/s ATM channel can carry about 90 Mb/s of CBR traffic. Comparing this with the results in Table 11, it is clear that for D_{max}=25 ms, a 100Mb/s ATM channel can carry about twice as many CBR streams as in a 100Base-T segment. As D_{max} becomes about 60 ms or larger, the difference between the two networks becomes small. This is because at D_{max}=25 ms, small packet sizes are forced to be used in Ethernet, and the performance of CSMA/CD is not very good when the packet size is small. At greater D_{max}, greater packet sizes can be used. Furthermore, resubmission of the packets that are dropped at the MAC layer due to excessive collisions also results in improved network utilization for large D_{max} values.

6.3 VBR Video Traffic on an ATM Multiplexer

Now consider the CQ-VBR video traffic. In Table 18, we show the maximum number of CQ-VBR streams supportable in an ATM network and the corresponding network utilization for various values of D_{max} and g_{max}, and for the three video sequences. It is interesting to note that N_{max} does not depend on D_{max} very significantly in the range considered. Furthermore, as stated in the previous section, N_{max} also does not depend significantly on

g_{max}, which suggests that N_v could be chosen just below the "knee" such that the glitch rate is negligible. In almost all cases, the number of CQ-VBR streams supportable is greater than that of CBR streams, in some cases up to a factor of 2. In fact, the theoretical maximum multiplexing gain (i.e., the ratio of the average CBR and CQ-VBR rates for the same $\hat{s}_{min}$) is attained for the relatively large D_{max} values.

We have also simulated an ATM multiplexer where the video stream is first packetized into AAL-PDU's at regular time intervals T_f, and then the AAL-PDU's are segmented into cells. We have used T_f values up to 33 ms. Note that if the streaming mode is used, this approach results in virtually the same performance as with mapping the stream directly into cells (as a result of the highly clustered cell losses). If the messaging mode is used, we have observed that the number of streams supportable is the same as in the streaming mode when T_f is subtracted from D_{max}.

Now compare the number of streams supportable in ATM and in Ethernet. Comparing Tables 14 and 18, it is clear that again for D_{max}=25 ms a 100Mb/s ATM channel can support about twice as many CQ-VBR streams as in 100Base-T. Again, the reason is that for such a small value of D_{max}, in Ethernet, a small packet size is forced to be used, which makes the CSMA/CD scheme inefficient. Indeed, for larger D_{max} values the difference between the two networks is small. In fact, for the Commercials, $\hat{s}_{target}$=4.5, the number of streams supported by the Ethernet is somewhat greater compared to ATM.

7 Statistical Multiplexing of Heterogeneous Video Sources

In the preceding section, we have considered that all the video sources are generated by using the same video sequence. Here, we relax that assumption, and investigate the effect of mixing video streams with different contents, encoding schemes, encoder control schemes, and D_{max} values.

First, consider an ATM multiplexer at which multiple video streams generated by two different video contents are mixed; both of the video sequences are H.261 and CQ-VBR encoded, and both types of streams have D_{max}=25 ms. In Table 19, we show various mixtures of Videoconferencing and Commercials sequences under those conditions. The numbers are chosen such that the cell loss rates remain roughly the same. The first point

to be made is that the glitch statistics also remain about the same over all the mixtures considered. Secondly, the network utilization for the mixtures is at intermediate points compared to the homogeneous cases.

Most importantly, the results indicate that one can specify a simple admission control criteria in case of heterogeneous video contents by means of an "effective bandwidth", which is defined for a given content, a given D_{max}, and a given g_{max} as follows. Let the average rate of the video stream with a given content be R_a bits/second, and the maximum network utilization that can be achievable by multiplexing a number of such streams (with the same content) while meeting their delay and glitch rate constraints be U bits/second. Then the effective bandwidth R_e is defined as $(W/U)R_a$. The admission control criteria is then to admit a new stream as long as the total equivalent bandwidth in the network does not exceed W. Indeed, for all the mixtures in Table 19, the sum of the effective bandwidths is equal to W=100 Mb/s. The importance of this result is that it allows one to determine whether a heterogeneous mixture of video streams can be supported or not by simulating the network under homogeneous scenarios, and determining the effective bandwidth for each stream.

Note that some researchers have previously shown that one can determine an effective bandwidth for artificially generated traffic based on various models. The first two papers to introduce the notion of effective bandwidth are [33] and [34]. In [33], the traffic is considered to be generated according to either independent Poisson streams (possibly with different means), or slotted independent burst arrivals. In [34], on-off sources are considered. It is later shown analytically in [35] that for sources with multiple time-scale variations, (as it is the case with variable bit rate video), an effective bandwidth can also be determined. All these papers consider the multiplexer buffer to be infinite, and use a large deviations approximation; therefore, for a finite buffer size and a finite number of streams, the effective bandwidth is only an approximate notion. Our experimental results indicate the validity of this approximation for real video traffic, and for the number of streams and the buffer sizes considered in this paper.

Similar results apply for mixing video sequences with different encoding schemes, and encoder control schemes. Examples to these are shown in Tables 20 and 21, respectively.

As for mixing streams with different delay constraints, consider first that the multiplexer buffer is still a FIFO, with no priority among streams with different delay constraints. In

this case, the network utilization should not exceed the value that would satisfy the delay and glitch rate requirements for the streams with the most stringent delay requirement. Another possibility is to assign a high priority to those streams that have a more stringent delay requirement. This allows more of the streams with the relaxed delay requirement to be admitted without degrading the performance for the other streams; however, the additional network utilization that can be gained is fairly small, considering that the number of streams supportable does not increase significantly with D_{max}.

The next issue then is to determine what buffer size to use. If one uses a buffer size that is optimum for the stringent delay requirement, then the streams with a relaxed requirement also incur the same type of cell losses as those with a stringent delay requirement. If, on the other hand, one uses a buffer size that is optimum for the relaxed delay requirement, then the streams with a stringent requirement would experience more packet losses, which translate into glitches with larger d and L_{tot}; but the streams with a relaxed requirement would experience much less cell loss than the former case. Therefore, the buffer size in this case is a trade-off between the cell loss rate for the streams with a stringent and a relaxed delay requirement. In Table 22, we show an example case for Commercials, H.261, CQ-VBR, and for D_{max} values of 25 ms and 500 ms. In this example, the buffer size is chosen to be 117500 cells, which is optimum for D_{max}=500 ms. It is clear that the streams with D_{max}=25 ms suffer much longer glitches compared to the case where the optimum M for D_{max}=25 ms is used. On the other hand, when the two types of streams are mixed, the streams with D_{max}=500 ms experience no cell loss.

As for the Ethernets, again a similar effective bandwidth approach applies, which is as expected given the insensitivity of network utilization to the video content, encoding scheme, and encoder control scheme. For streams with different D_{max} values, again the admission control criteria should correspond to the most stringent D_{max} value, since prioritization is not possible among streams transmitted by different sources over an Ethernet segment.

8 Multihop ATM Networks

Since ATM networks are envisioned to be deployed both in the local and wide area environments, it is also interesting to study cases where the video streams traverse multiple ATM switches. The general problem that we address can be stated as follows. Consider a number of ATM switches connected in an arbitrary mesh topology, over which some video traffic is currently being sent (according to an arbitrary traffic matrix). When a new video stream is requested to be transmitted over the network, the problem is to determine whether or not the stream can be carried while meeting the delay and quality requirements of all current video streams, as well as the newly requested one.

Some researchers have studied multihop ATM scenarios in the past either by means of analytical models, or computer simulations driven by artificially generated traffic [36, 37]. An important conclusion that is reached in those papers is that when a burst of cells belonging to a source passes through a multiplexer, the spacing between the cells belonging to the burst becomes greater at the output of the multiplexer compared to its input. Thus, traffic offered to a multihop network has the highest degree of burstiness in the first hop traversed, and the burstiness decreases as more and more hops are traversed.

This result allows us to construct a worst-case scenario in terms of the number of streams that can be supported in a multihop environment. As shown in Figure 48, in this scenario multiple ATM switches are connected in tandem, interconnected by channels with a bandwidth of W bits/s. We denote the number of switches by H. We assume that N_{ete} streams go through H hops, thus traversing through all the switches. In addition, in every hop i there are N_{loc}^{i} streams that are generated by stations directly connected to the switch i; i.e., they enter and exit the network at switch i. When $N_{ete} = 1$, this is a worst case scenario from the point of view of the single video stream that travels multiple hops, since it encounters video traffic that is the most bursty. Thus, if we denote by N_{max} the value of $N_{loc}^{i} + 1$ for which still the video streams' delay and loss requirements can be satisfied, then for any arbitrary mesh topology, at least N_{max} streams can be supported on every link as long as all the streams travel at most H hops.

We have simulated the network for this scenario where all the video streams are generated using the *commercials* and *videoconferencing* sequences, H.261 encoded using the CQ-VBR control scheme at $\hat{s}_{target}$=4.5. We have chosen W=100 Mb/s, and H=$\{$2,4,8,16$\}$.

We consider N_{loc}^i to be the same in every hop, and denote it by N_{loc}.

In Figure 49, we plot the glitch rate for the stream that travels the multiple hops as a function of the number of video streams per hop (i.e., N_{loc}+1) for *commercials*, H=$\{$1,2,4,8,16$\}$. An interesting observation is that the glitch rate increases sharply beyond a knee point in the multihop cases, just as in the single hop case. Furthermore, the knee point does not change too much as the number of hops are varied, suggesting that the maximum number of streams supportable per hop is not significantly dependent on the number of hops. This can be explained by examining the distribution of delay in each of the hops. In Figure 50, we plot the complement of the cumulative distribution function of delay that the cells experience in individual hops and end-to-end for *commercials*, H=8, N_{loc}=59, N_{ete}=1. As it can be seen in the figure, about 99% of the cells experience delays less than 0.1 ms in each hop; in fact, the average delay per hop is around 0.04 ms. Therefore, the average end-to-end delay is equal to $H \times 0.04$ ms plus the average cell formation time (0.02 ms), which is very small compared to D_{max} even for a large number of hops such as 16. The delay distribution in each hop is fairly independent; thus, the end-to-end delay is the convolution of the probability distribution function (pdf) of delay in each individual hop and the pdf of the packet formation time. Given the small average of the delay distribution for each hop, the tail of the end-to-end delay distribution does not increase very much. Furthermore, since the cell loss rate is very sensitive to the number of streams at the knee, decreasing the number of streams per hop by just a few results in a large decrease in the tail distribution, allowing many more hops to be traveled while meeting the delay and glitch rate constraints.

To further quantify the insensitivity of the number of streams per hop to H, in Table 23, we show $N_{max} = N_{loc} + 1$ for 1, 2, 4, 8, and 16 hop networks, for g_{max}=0.1 per minute, and D_{max}=$\{$20,100$\}$ ms. It is clearly seen that the number of streams supportable per hop decreases by a very small amount: from 64 for H=1 to 59 for H=16 when D_{max}=25 ms. When D_{max}=100 ms, the insensitivity is even more apparent: N_{max} is equal to 69 for H=1, and it is equal to 68 for H between 2 and 16.

It was not practical to simulate cases with a far greater number of hops than 16, given the memory and processing speed limitations. However, an estimate of the delay distribution can be obtained for such a large number of hops by convolving the distributions that we have obtained for a smaller number of hops. In Figure 51, we show the result for such a

convolution for H=32, N_{loc}=58, and N_{ete}=1. The cell loss rate obtained by the convolution for D_{max}=25 ms (including the 4.4 ms of encoding/decoding time) is around 3×10^{-6}, which would yield an acceptable glitch rate. Thus, even for H=32, 59 streams per hop can be supported.

For comparison purposes, in Table 24, we show the total number of streams supportable per hop for a similar scenario, but for $N_{ete} = 50$. It is interesting to note that these numbers are not very different from those obtained for N_{ete}=1.

Similar results also apply for the *videoconferencing* sequence, as shown in Table 25 for D_{max}=25 ms, and g_{max}=0.1 per minute.

For comparison purposes we have also considered another scenario, where the ATM multiplexers are organized into a ring for purposes of symmetry, as shown in Figure 52. There are G multiplexers; in every hop, N_v streams enter the multiplexer, and travel H hops. (Figure 52 is for H=2.) Therefore, the total number of streams that are multiplexed in every hop is $N_v \times H$. For this scenario too, we have considered all video streams to be *commercials*, H.261, CQ-VBR, $\hat{s}_{target}$=4.5. We have found that here too, the number of streams supportable per hop is similar to the previous scenario. For example, for D_{max}=25 ms, G=5, and H=3, we have N_v=20, giving $N_v \times H$=60, the same as in the first scenario.

These results indicate that the number of streams supportable per hop is only weakly dependent on the number of hops that the individual streams traverse. This is a useful result in terms of admission control: it suggests that in any arbitrary mesh topology, one can simply determine if a given video stream can be admitted by checking the number of streams in every hop independently.

9 Integrated Video and Data Services

In this section, we consider a mixture of video and data traffic to be present on the network. We first examine the effect of data traffic load and burst size on the video performance. We then examine the effect of video traffic on data.

As described in Section 3, we assume that data sources generate messages with a fixed size M_d, with uniform interarrival times, the average of which is denoted by u[3].

[3]The simulator was also run with exponentially distributed interarrival times, and exponentially dis-

If a message is larger than a full size packet (1500 bytes), it is divided into as many full-size packets as needed. In that case, the packets belonging to the message are placed into the transmit queue as a bulk arrival.

In this study, we have limited ourselves to 3 message sizes. To generate non-bursty data traffic, we let M_d=1 kbyte, in which case each message fits in a single packet. To generate bursty data traffic, we choose M_d=10 kbytes and M_d=100 kbytes. After packetization at the transport layer, these message sizes correspond to 7 and 67 back-to-back packets respectively.

The number of data sources in the network was fixed at two in the 10Base-T Ethernet simulations and twenty in the 100Base-T Ethernet and ATM simulations [4].

In the following, we give all our results using the *videoconferencing* sequence, CBR encoded using the VSRP scheme. As far as the effect of data traffic on video is concerned, this case is representative of all the other video contents, encoder control schemes, and packetization schemes considered in this paper.

In Figure 53, we show for the 10Base-T the glitch rate as a function of the data load G_d for N_v=10, D_{max}=100 ms, VSRP. As indicated by the figure, the non-bursty data behaves very similarly to a video stream (i.e., if it were replaced by a number of video streams such that the equivalent load generated is the same, then the effect would be roughly the same). On the other hand, the figure indicates that the bursty data has a very detrimental effect on the video traffic.

In Figure 54, we show N_{max} as a function of the data load for VSRP, g_{max}=0.1 per min. Part (a) of the figure is for D_{max}=25 ms, and part (b) is for D_{max}=100 ms. Figure 55 is similar to Figure 54, except that g_{max}=1 per min. As these figures indicate, the degradation due to increasing burst size is less apparent with less stringent delay and loss constraints. For example, as shown in Figure 54(a), when D_{max}=25 ms, g_{max}=0.1 per min., and G_d=0.5 Mb/s, 7 streams can be supported with a data message size of 1 kbyte. For the same data load, and for a data message size of 10 kbytes, no video streams can be supported.

tributed message sizes. The results did not differ significantly from those obtained with the uniform distribution with fixed size messages. The uniform distribution was chosen over the exponential distribution for its lower complexity and faster convergence properties.

[4] We have also simulated some test cases with 20 data sources for 10Base-T, and 2 data sources for 100Base-T. For a given data load and burst size, the number of streams supportable did not change as we changed the number of data sources from 2 to 20.

Compare this with Figure 55(b), where D_{max}=100 ms, and g_{max}=1 per min. Under those conditions, when G_d =0.5 Mb/s, and M_d=1 kbyte, 13 streams can be supported, whereas for the same data load, and for M_d=10 kbytes, 7 streams can be supported.

Consider now a 100Base-T segment carrying video and data traffic. In Figure 56, we plot N_{max} as a function of G_d, for M_d=$\{$1,10,100$\}$kbytes, for the VSRP scheme. Part (a) of the figure is for D_{max}=25 ms, and g_{max}=0.1 per min., and part (b) is for D_{max}=100 ms, and g_{max}=1 per min. Comparing these figures with Figure 54(a) and Figure 55(b), we can see that in 100Base-T, about 10 times larger burst sizes than those in 10Base-T cause an equivalent degradation. This is due to the 10 times larger transmission speed and 10 times smaller slot size, causing the same size burst to be transmitted about 10 times faster, and thereby affecting stream packets for a time about 10 times shorter in 100Base-T[5].

As for the effect of video traffic on data, the largest effects will occur when N_{max} video streams are being carried over the network. For a given data load and data burst size, N_{max} is greater for relaxed delay and glitch rate constraints; therefore, this will represent the worst case for data. In Table 26, we show the average and standard deviation of delay experienced by data packets for 10Base-T, G_d=1 Mb/s, M_d =$\{$1,10$\}$kbytes, and N_{max} CBR *videoconferencing* sequences being carried on the network using the VSRP scheme for D_{max}=100 ms and g_{max}=1 glitch per minute. As the table indicates, the average data traffic delay is quite low; in fact, most of the delay is experienced due to the data packets belonging to the same message being queued behind each other. Furthermore, in this scenario, no data packets were lost.

Similarly, In Table 27, we show the average and standard deviation of delay experienced by data packets as well as data packet loss ratio for 100Base-T, G_d=10 Mb/s, M_d =$\{$1,10,100$\}$kbytes, and under the same conditions for video as in the previous paragraph. Again, the data delay is quite low. This time, there is some data packet loss, but it is on the order of 10^{-5} to 10^{-4}; such amounts of loss should not cause any significant performance degradation in data applications due to the delays involved in retransmissions.

In Figure 57, we plot for ATM the cell loss rate for the video sources as a function of N_v for G_d=10 Mb/s, for M_d=$\{$50,100$\}$ kbytes, for the Videoconferencing sequence, H.261 CQ-

[5]Note that the parameter a does not have an important effect while a data burst is being transmitted; regardless of the value of a, the i'th packet of the burst will always collide with the packets that were deferring to the $i-1$'st packet of the burst owing to the 1-persistence of the Ethernet MAC protocol.

VBR encoded at $\hat{s}_{target}$=4.5, and for D_{max}=25 ms. It is clear that as M_d is increased, the cell loss rate also increases. In particular, for M_d=100 kbytes, video packets start to experience losses for much smaller values of N_v compared to the case with M_d=50 kbytes. We have also collected the glitch statistics for M_d=100 kbytes, and observed that the glitches in this case have much smaller duration and L_{tot}; for example, for N_v=45, g=1.9 per minute, d=5.5 frames, A_{max}=31.3 macroblocks, and L_{tot}=144 macroblocks. This indicates that the cell losses here are not as much clustered; hence, for a given packet loss rate, more glitches are experienced compared to the case for video alone, but the glitches affect only about one-tenth of a frame. The reason is that for a relatively small value of N_v such as 45, cell losses occur only when a data burst is being transmitted. Since the burst duration at M_d=100 kbytes is only 8 ms, cell losses occur only during a small portion of a frame.

Now consider Figure 58, where we plot N_{max} as a function of G_d for *videoconferencing*, D_{max}=25 ms, g_{max}=1 per minute, and various values of M_d. It is clear that for small M_d, the number of streams supportable decreases gradually as G_d is increased. However, for M_d=100 kbytes, it decreases rapidly as G_d is increased. These results suggest the usefulness of giving priority to video traffic over data traffic, as it would normally be the case in an ATM network. As for comparing the effect of data traffic on video for Ethernet and ATM, it is clear that in Ethernet, the number of streams supportable is more sensitive to the burstiness of the data traffic.

10 Conclusions

We have presented the performance of Ethernets and ATM networks carrying multimedia traffic. The evaluation is done by computer simulation, using real video sequences. The sequences are encoded using the H.261 and MPEG-1 video encoding standards, with Constant Bit Rate and Constant-Quality Variable Bit Rate encoder control schemes. We take into account all components of the end-to-end delay, measured from the time a frame is taken in the camera, to the time it is displayed at the receiver's screen. We measure the effect of cell loss on the displayed video in terms of glitch rate, duration, and spatial extent, which are measured by taking into account the dependence among frames.

We have characterized the quality degradation in the displayed video due to packet loss, and developed accurate measures in terms of statistics of glitches that the viewer perceives.

We have shown that for ATM the statistics per glitch depend significantly on the video load, the video encoding scheme, and the end-to-end delay constraint. On the other hand, for Ethernet, the per-glitch statistics are fairly independent of the video load. We have also shown that in ATM, once the packet losses become non-negligible, the rate of glitches increases very sharply as the video load increases. The rate of increase for the glitch rate is not as sharp in Ethernet.

We show that both Ethernets and ATM networks can support video well in the absence of bursty data traffic. In 10Base-T, the number of CBR and CQ-VBR streams supportable are comparable. On the other hand, 100Base-T and a 100Mb/s ATM multiplexer can support up to twice as many CQ-VBR streams as CBR streams for given maximum end-to-end delay requirement, minimum level of video quality at the source, and maximum glitch rate. We also compare the number of streams supportable in 100Base-T and 100 Mb/s ATM multiplexer. For low values of end-to-end delay requirement, we show that ATM networks can support up to twice as many video streams of a given type as Ethernets. For relaxed end-to-end delay requirements, both networks can support about the same number of video streams of a given type.

We also consider scenarios consisting of mixtures of heterogeneous video traffic sources in terms of the video content, video encoding scheme and encoder control scheme, as well as the end-to-end delay requirement. We show that when video streams with different content, encoding scheme, and encoder control schemes are mixed, the results are at intermediate points compared to the homogeneous cases, and the maximum number of supportable streams of a given type can be determined in the presence of other types of video traffic by considering an "equivalent bandwidth" for each of the stream types. As for mixing streams with different delay constraints, the multiplexer buffer size represents a trade-off between glitches of the streams with stringent and relaxed delay constraints.

We also consider multihop ATM network scenarios, and determine that the number of streams supportable on a given multiplexer in an arbitrary mesh topology is only weakly dependent on the number of hops that the multiplexed video streams traverse. Therefore, a simple admission control mechanism that treats each hop independently can be employed.

When data traffic is introduced in the network, we have observed that the number of streams supportable decreases significantly as the data burst size is increased. Bursty data is most detrimental in 10Base-T, where even 10 Kbyte bursts have a very severe effect.

In 100Base-T, 10 times as large bursts have the same effect as in 10Base-T. In ATM, the burst size has to be a significant fraction of the buffer size before the bursts start to have a large impact.

In order to avoid the effect of bursty data traffic on video in an ATM network, video traffic can be given a higher priority with respect to the data traffic. In an Ethernet, where prioritization is not possible, the data load on a segment must be kept limited by throttling the data sources, or avoiding mixing data and video through appropriate deployment of stations.

References

[1] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Transactions on Communications*, vol. 36, pp. 834–844, July 1988.

[2] R. Rodriguez-Dagnino, M. Khansari, and A. Leon-Garcia, "Prediction of Bit Rate Sequences of Encoded Video Signals," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 305–314, Apr. 1991.

[3] F. A. Tobagi and N. Gonzalez-Cawley, "On CSMA-CD local networks and voice communication," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Las Vegas, Nevada), pp. 122–127, IEEE, March/April 1982.

[4] T. A. Gonsalves and F. A. Tobagi, "Comparative Performance of Voice/Data Local Area Networks," *IEEE Journal on Selected Areas in Communications*, pp. 657–669, June 1989.

[5] F. A. Tobagi, F. Borgonovo, and L. Fratta, "Expressnet: A High-Performance Integrated-Services Local Area Network," *IEEE Journal on Selected Areas in Communications*, vol. 1, pp. 898–913, Nov. 1983.

[6] "ISO/IEC 8802-4:1990, Information Processing Systems – Local Area Networks – Part 4:Token-Passing Bus Access Method and Physical Layer Specifications," 1985.

[7] J. Zdepski, K. Joseph, and D. Raychaudhuri, "Packet transport of VBR interframe DCT compressed digital video on a CSMA/CD LAN," in *Proceedings of the Conference*

on Global Communications (GLOBECOM), (CA), pp. 886–892 (25.3), IEEE, Dec. 1989.

[8] F. Edwards and M. Schulz, "Performance of VBR Packet Video Communications on an Ethernet LAN: A Trace-Driven Simulation Study," in *Proceedings of the 1994 IEEE 13th Ann. Int. Phoenix Conf. on Computers and Communications*, (Phoenix, AZ), pp. 427–433, Apr. 1994.

[9] K. M. Nichols, "Network Performance of Packet Video on a Local Area Network," in *IEEE 11th Annual International Phoenix Conference on Computers and Communications*, (Scottsdale, Arizona), pp. 659–666, IEEE, April 1–3 1992.

[10] A. Desimone, R. Nagarajan, and Y. Wang, "Desktop and Network Performance Issues in Multimedia Conferencing and Collaboration," tech. rep., AT&T Bell Labs, Holmdel, New Jersey, 1994.

[11] E. Gay, H. Hin, and L. Kiong, "Performance Evaluation of Deploying Full Motion Digital Video in Ethernet," in *Proceedings of 1994 IEEE Region 10's Ninth Annual International Conference*, (singapore), pp. 754–758, Aug. 1994.

[12] 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.

[13] D. M. Cohen and D. P. Heyman, "Performance Modeling of Video Teleconferencing in ATM Networks," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 3, pp. 408–420, Dec. 1993.

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

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

[16] A. R. Reibman and A. W. Berger, "Traffic Descriptors for VBR Video Teleconferencing over ATM Networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 329–339, June 1995.

[17] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta, and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks," in *IEEE INFOCOM '93*, pp. 919–926, 1993.

[18] O. Rose, "Delivery of MPEG Video Services over ATM," Research Report Series 86, University of Wurzburg, Institute of Computer Science, Aug. 1994.

[19] O. Rose, "Approximate Analysis of an ATM Multiplexer with MPEG Video Input," Tech. Rep. 79, University of Wurzburg, Institute of Computer Science, Jan. 1994.

[20] P. Pancha and M. El Zarki, "Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 3, pp. 190–198, June 1993.

[21] "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.

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

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

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

[25] İ. Dalgıç and F. A. Tobagi, "Characterization of Video Traffic and Video Quality for Various Video Encoding Schemes and Various Encoder Control Schemes," Computer Systems Laboratory Technical Report, TR-96-701, Stanford University (in preparation), Aug. 1996.

[26] F. A. Tobagi and İ. Dalgıç, "Constant Quality VBR Video Encoder Control Scheme and its Traffic and Quality Characterization," Submitted to *IEEE JSAC*, special issue on Real-Time Video Services in Multimedia Networks.

[27] 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.

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

[29] "B-ISDN ATM Adaptation Layer (AAL) Specification," ITU-T Recommendation I.363, March 1993.

[30] İ. Dalgıç, W. Chien, and F. A. Tobagi, "Evaluation of 10Base-T and 100Base-T Ethernets Carrying Video, Audio and Data Traffic," in *Proceedings of the IEEE INFOCOM '94*, (Toronto, Canada), pp. 1094–1102, June 1994.

[31] A. Albanese, J. Blömer, J. Edmonds, and M. Luby, "Priority Encoding Transmission," Tech. Rep. TR-94-039, International Computer Science Institute, Berkeley, Aug. 1994.

[32] C. Leicher, "Hierarchical Encoding of MPEG Sequences Using Priority Encoding Transmission (PET)," Tech. Rep. TR-94-058, International Computer Science Institute, Berkeley, Nov. 1994.

[33] F. Kelly, "Effective Bandwidths at Multi-Class Queues," *Queueing Systems*, pp. 5–16, Sept. 1991.

[34] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 968–981, Sept. 1991.

[35] D. Tse, R. Gallager, and J. N. Tsitsiklis, "Statistical Multiplexing of Multiple Time-Scale Markov Streams," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1028–1038, Aug. 1995.

[36] Y. Ohba, M. Murata, and H. Miyahara, "Analysis of Interdeparture Processes for Bursty Traffic in ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 468–476, Apr. 1991.

[37] M. D'Ambrosio and R. Melen, "Evaluating the Limit Behavior of the ATM Traffic Within a Network," *IEEE/ACM Trans. Networking*, vol. 3, pp. 832–841, Dec. 1995.

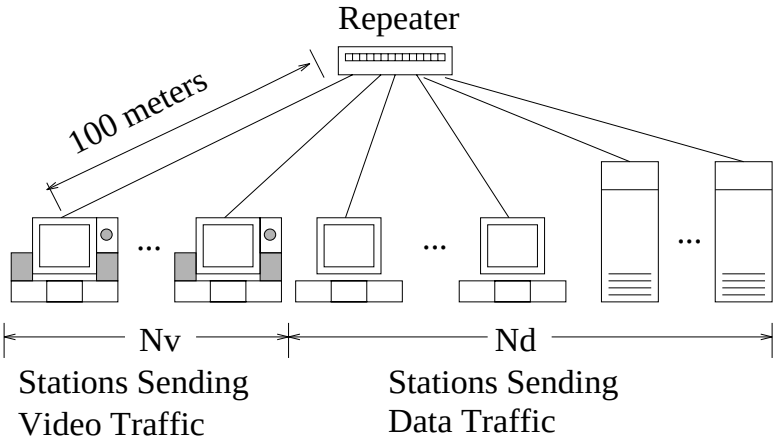
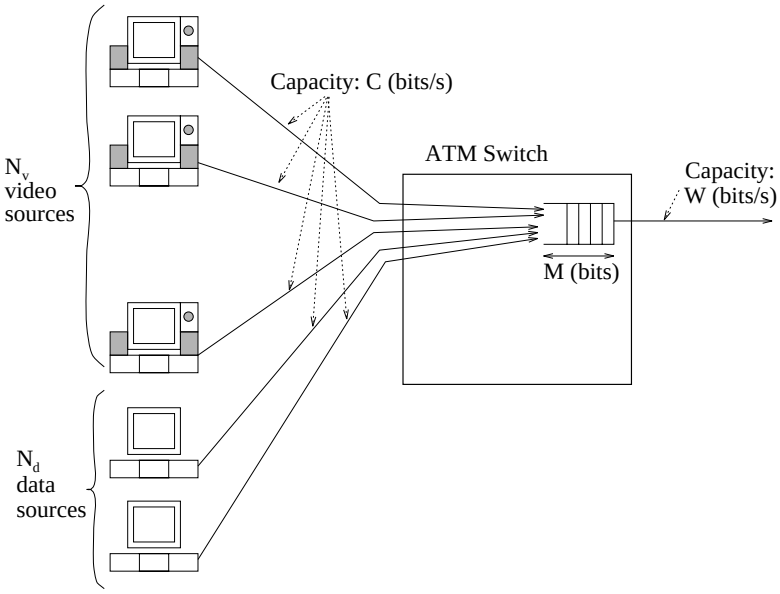


Figure 1: The network scenario under consideration for Ethernets.



(a) Single Hop

Figure 2: The network scenario under consideration for ATM.

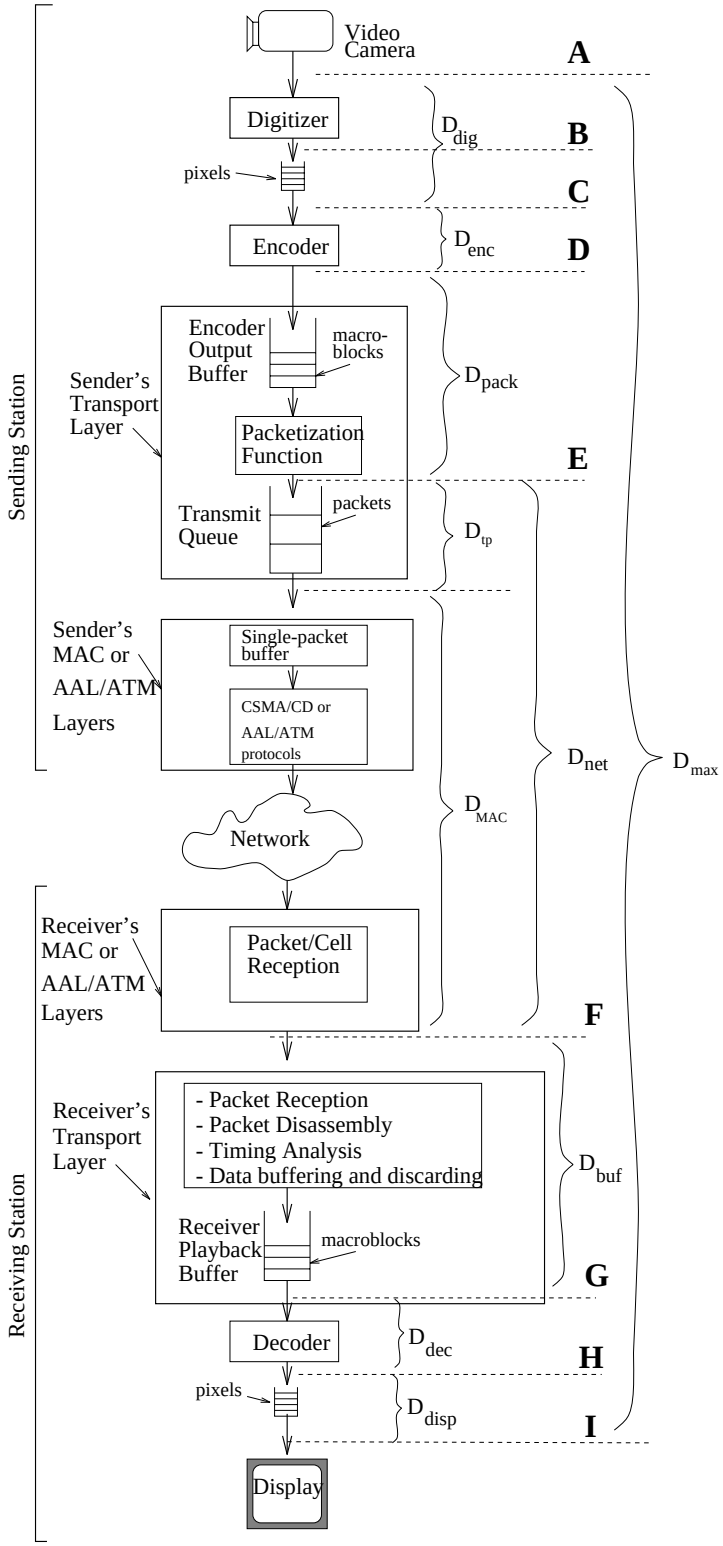


Figure 3: Components in the end-to-end data path traversed by the video signal and the corresponding delay components.

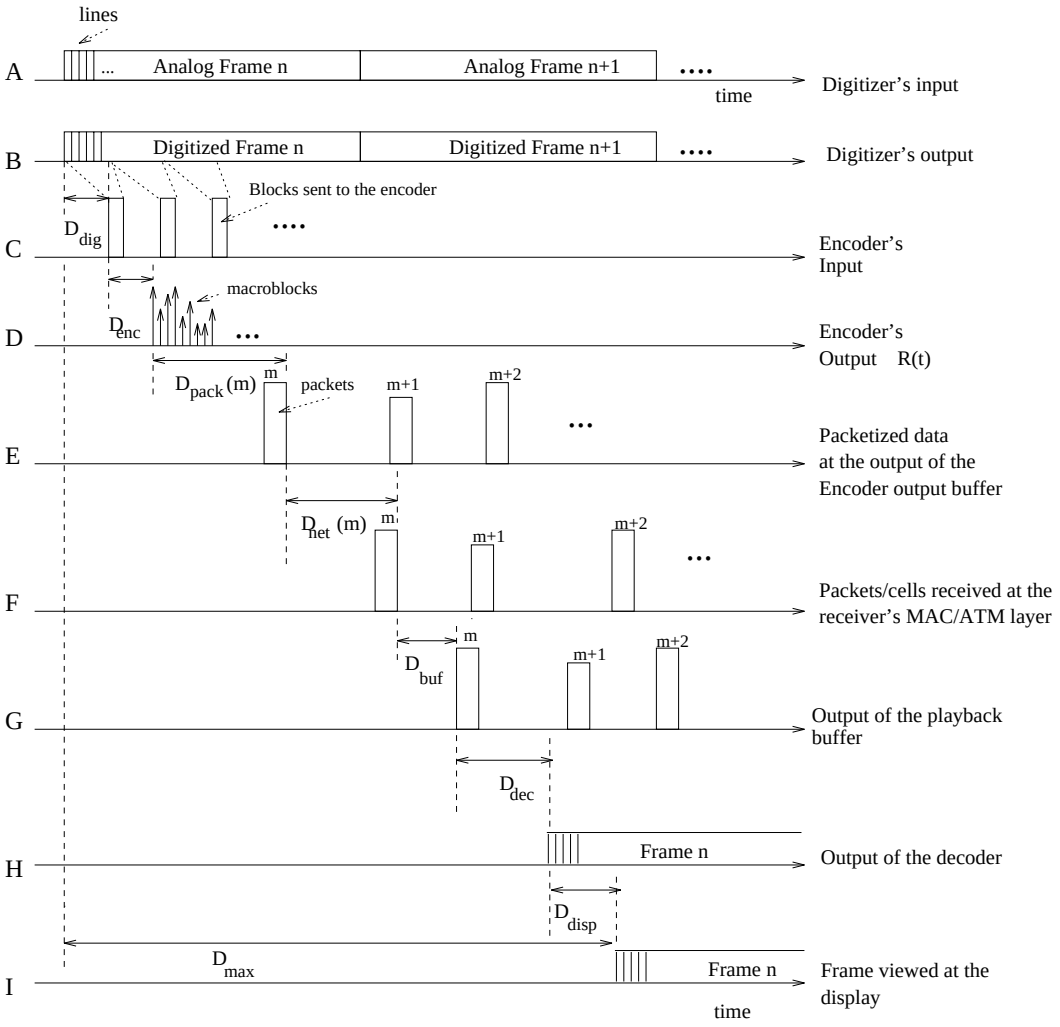


Figure 4: End-to-end timing diagram.

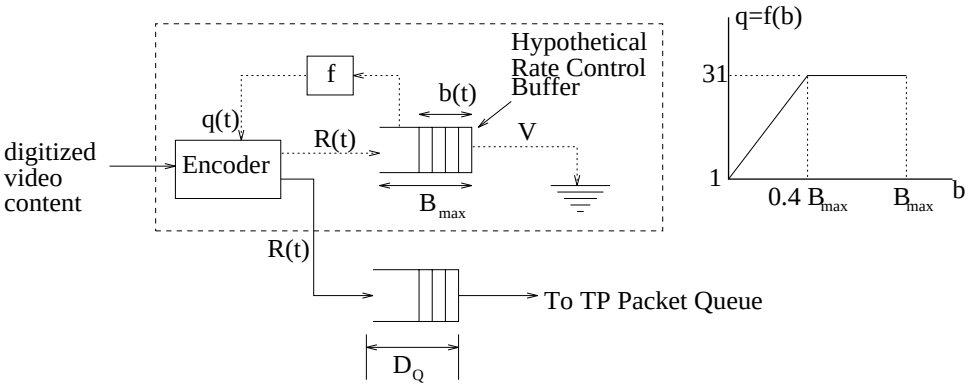


Figure 5: Block diagram of a Constant Bit Rate (CBR) encoder.

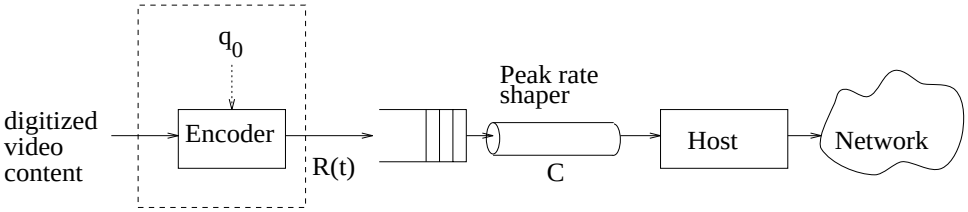


Figure 6: Block diagram of an Open-Loop VBR encoder.

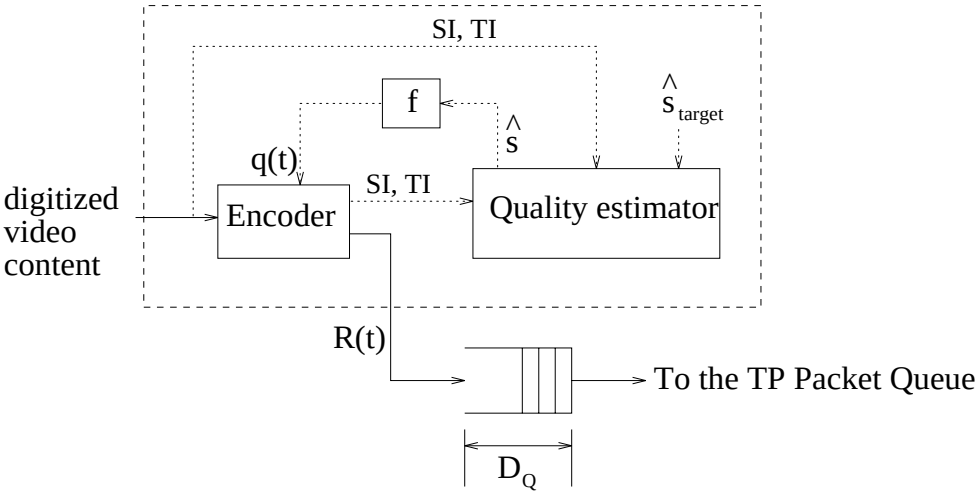


Figure 7: Block diagram of a Constant-Quality Variable Bit Rate (CQ-VBR) encoder.

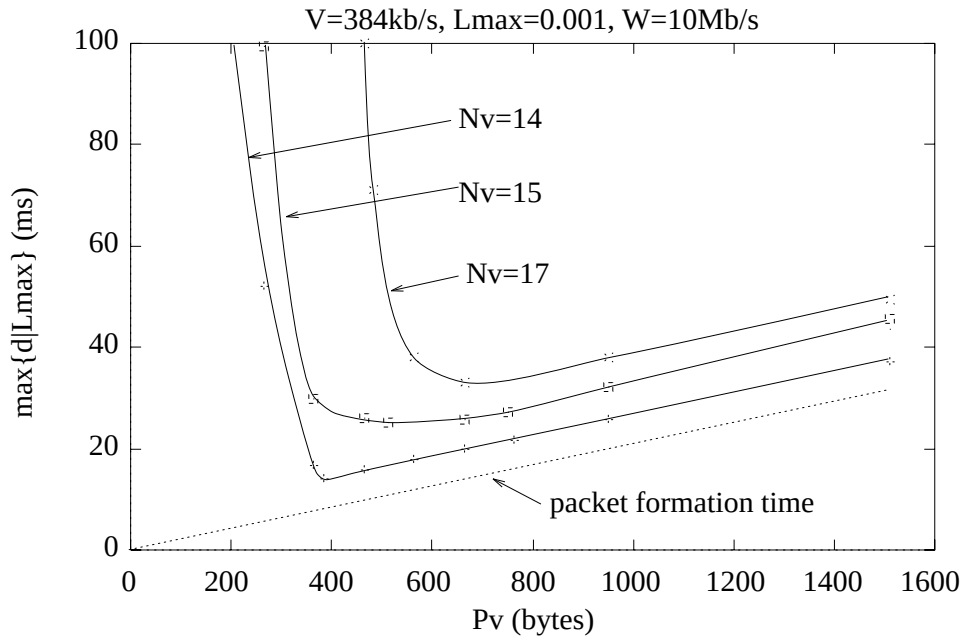


Figure 8: $\max\{d|L_{max}\}$ vs. P_v, V=384 kb/s, L_{max}=0.001, W=10 Mb/s

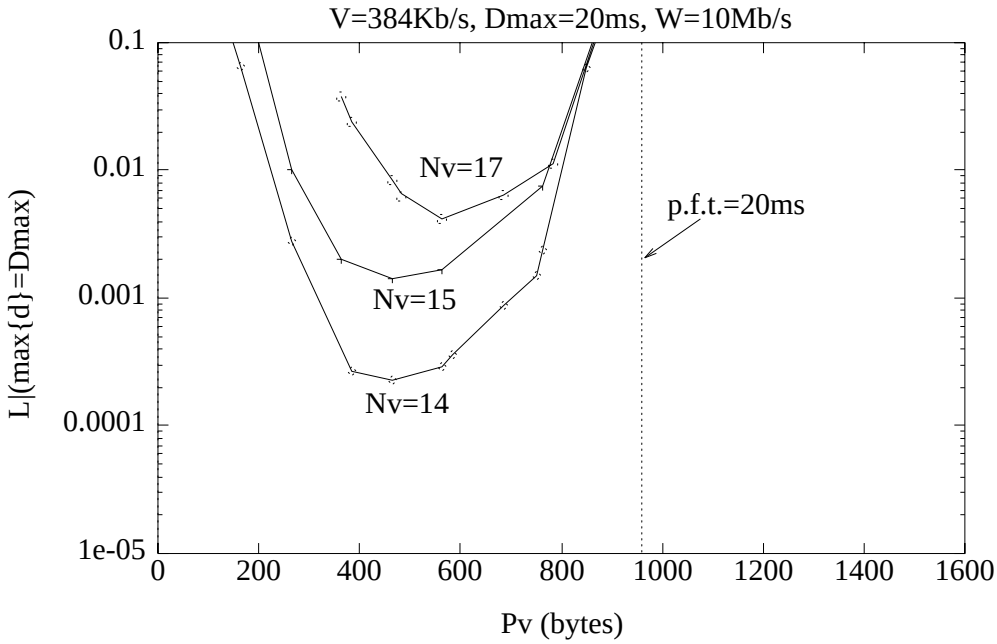


Figure 9: $L|(\max\{d\}=D_{max})$ versus P_v, V=384 Kb/s, D_{max}=20ms, W=10 Mb/s

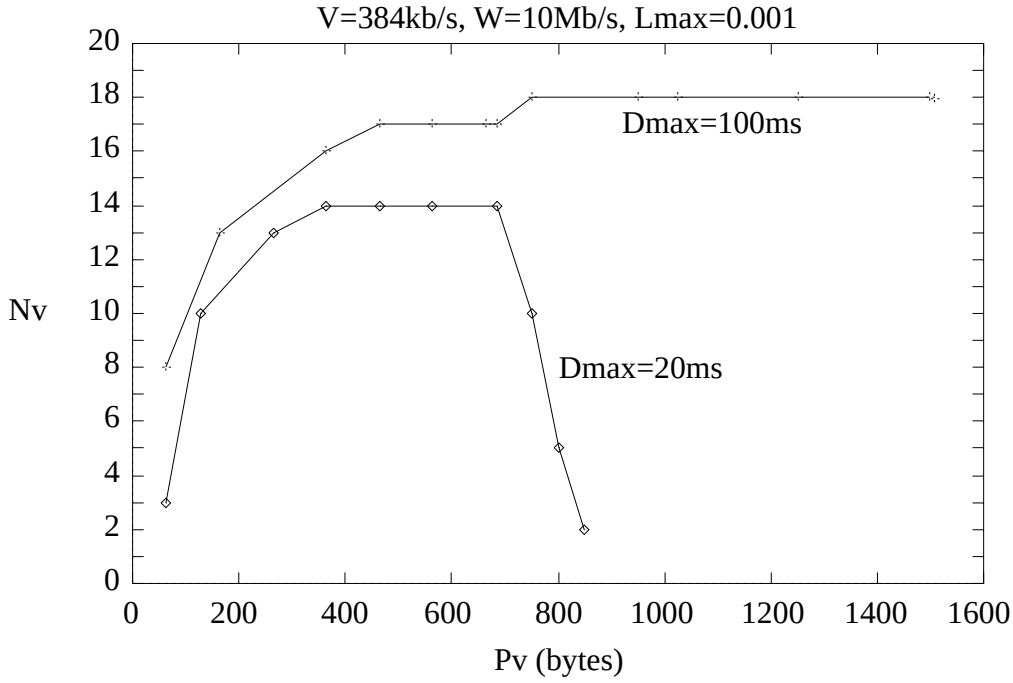


Figure 10: N_v vs. P_v for W=10 Mb/s, V=384 kb/s, L_{max}=0.001

D_{max} (ms)	L_{max}	64 kb/s	384 kb/s	1536 kb/s
20	0.0001	90	350	650
	0.001	100	450	800
	0.01	120	650	1500
100	0.0001	380	1250	650
	0.001	380	1250	1500
	0.01	480	1500	1500

Table 1: P_{opt} (bytes), W=10 Mb/s

D_{max} (ms)	L_{max}	64 kb/s	384 kb/s	1536 kb/s
		N_{max}	N_{max}	N_{max}
20	0.0001	38 (24%)	11 (42%)	4 (61%)
	0.001	55 (35%)	14 (54%)	4 (61%)
	0.01	64 (41%)	17 (65%)	5 (77%)
100	0.0001	78 (50%)	18 (69%)	5 (77%)
	0.001	89 (57%)	18 (69%)	5 (77%)
	0.01	104 (67%)	20 (77%)	5 (77%)
Bandwidth limit		156	26	6

Table 2: N_{max} and the corresponding network utilization, W=10 Mb/s

D_{max} (ms)	L_{max}	384 kb/s	1536 kb/s
20	0.0001	450	800
20	0.001	500	1000
20	0.01	650	1500
100	0.0001	1500	1500
100	0.001	1500	1500
100	0.01	1500	1500

Table 3: P_{opt} (bytes) for W=100 Mb/s

D_{max} (ms)	L_{max}	384 kb/s	1536 kb/s
		N_{max}	N_{max}
20	0.0001	130 (50%)	39 (60%)
20	0.001	138 (53%)	43 (66%)
20	0.01	160 (61%)	49 (75%)
100	0.0001	213 (82%)	54 (83%)
100	0.001	216 (83%)	55 (85%)
100	0.01	221 (85%)	55 (85%)
Bandwidth limit		260	65

Table 4: N_{max} and the corresponding network utilization for W=100Mb/s

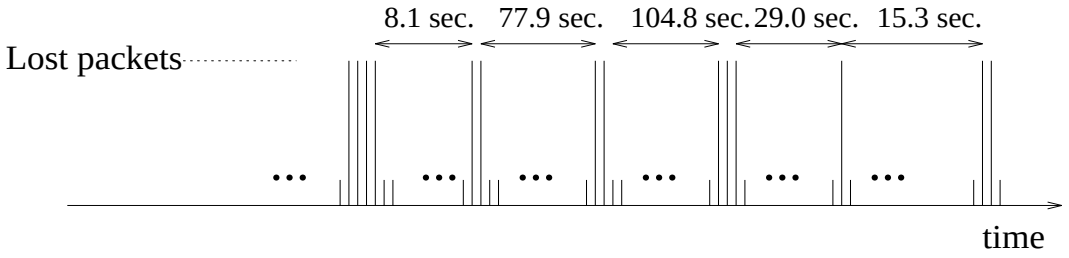


Figure 11: Clustering of lost packets for a source-destination pair for the *videoconferencing* sequence, CBR, V=384 kb/s, N_v=18, VSRP, T_f=40 ms, D_{max}=100 ms, 10Base-T.

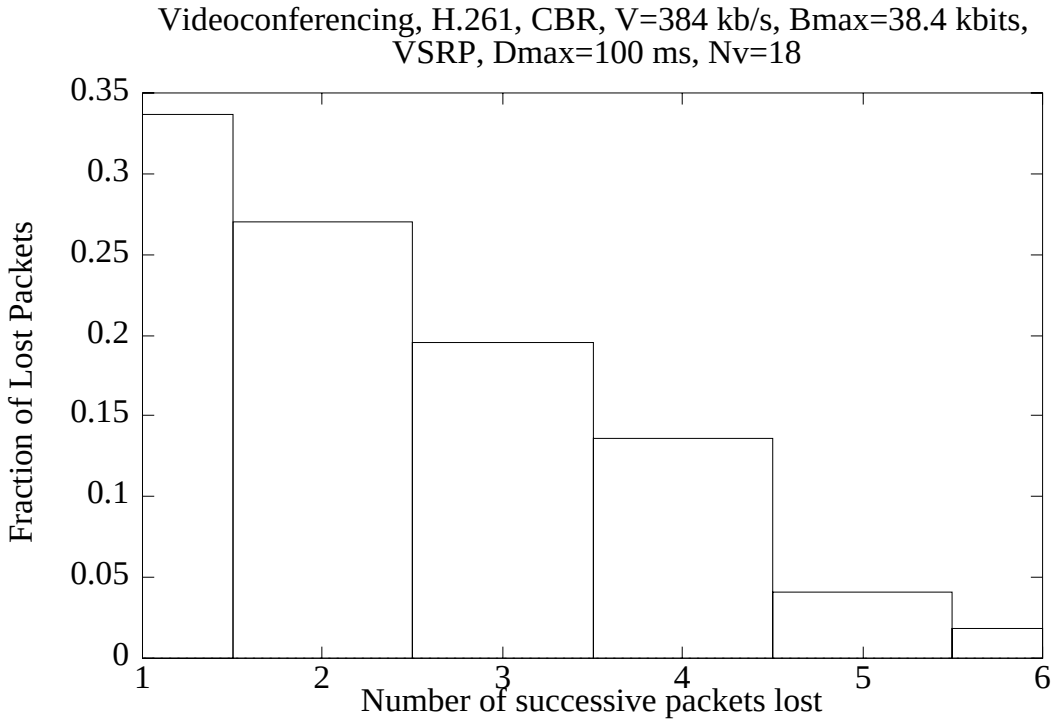


Figure 12: Histogram of number of packets lost in packet loss clusters for the *videoconferencing* sequence, CBR, V=384 kb/s, N_v=18, VSRP, T_f=40 ms, D_{max}=100 ms, W=10 Mb/s.

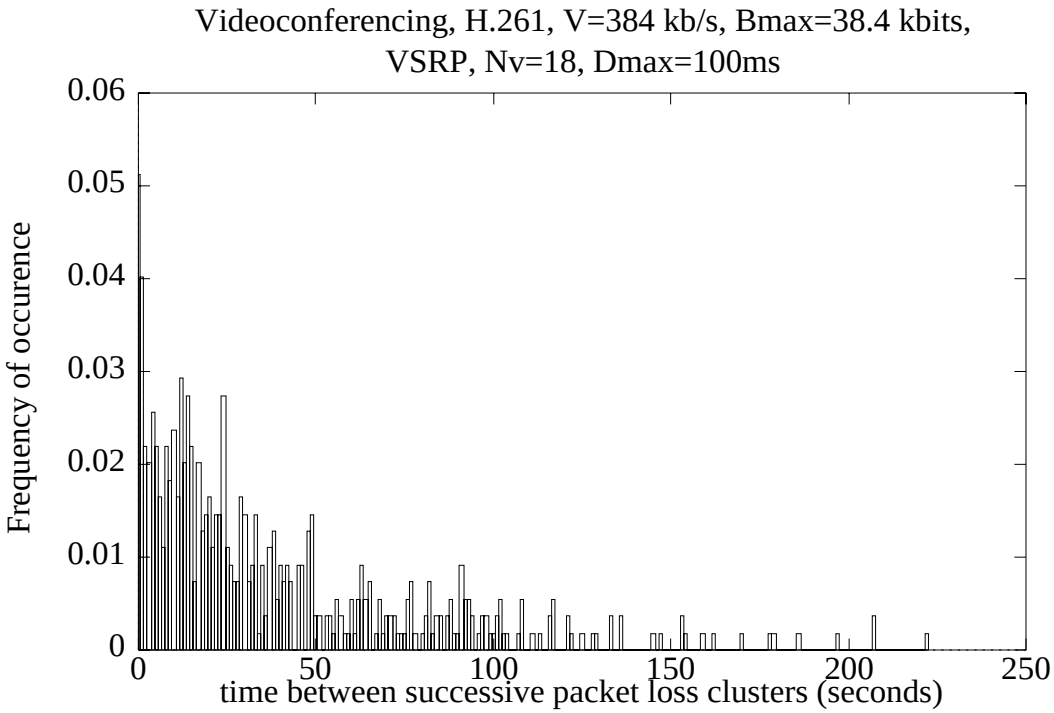
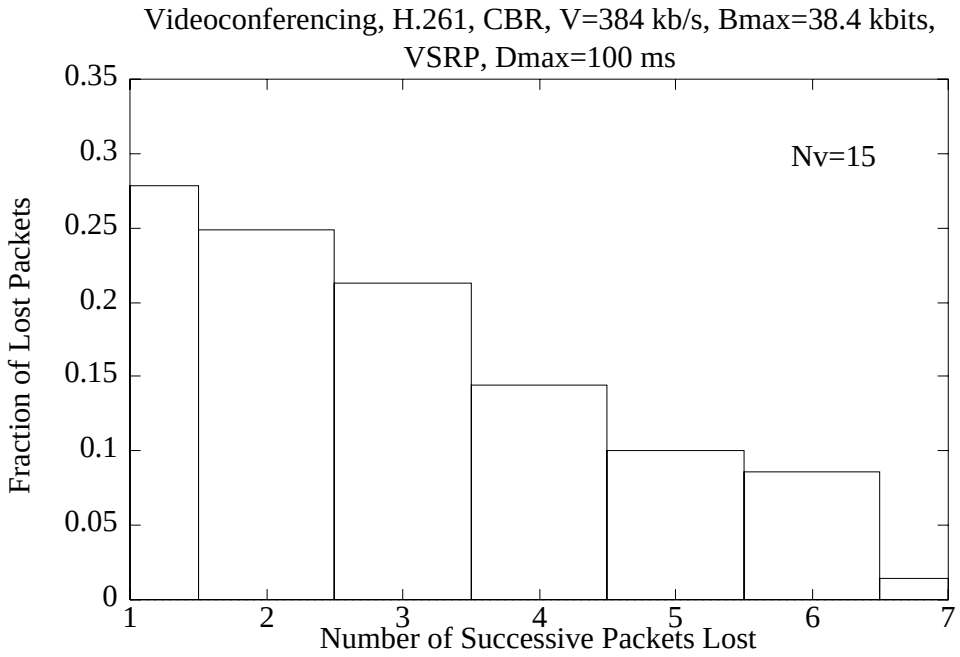
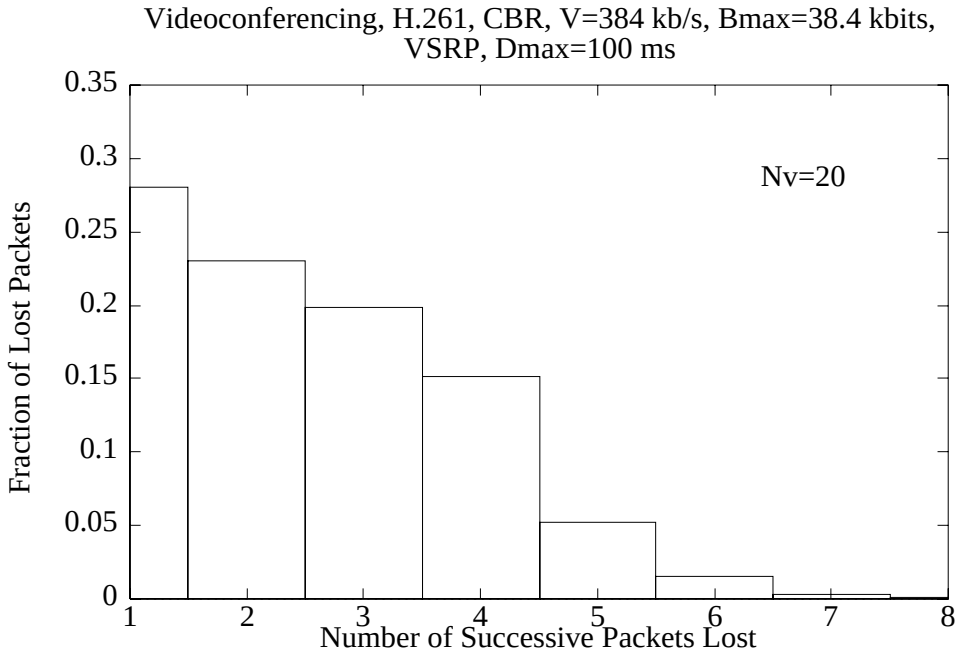


Figure 13: Histogram of time between packet loss clusters for the *videoconferencing* sequence, CBR, V=384 kb/s, N_v=18, VSRP, T_f=40 ms, D_{max}=100 ms, W=10 Mb/s.



(a) N_v=15



(b) N_v=20

Figure 14: Histogram of number of packets lost in packet loss clusters for different traffic loads, for the *videoconferencing* sequence, CBR, V=384 kb/s, VSRP, T_f=40 ms, D_{max}=100 ms, W=10 Mb/s.

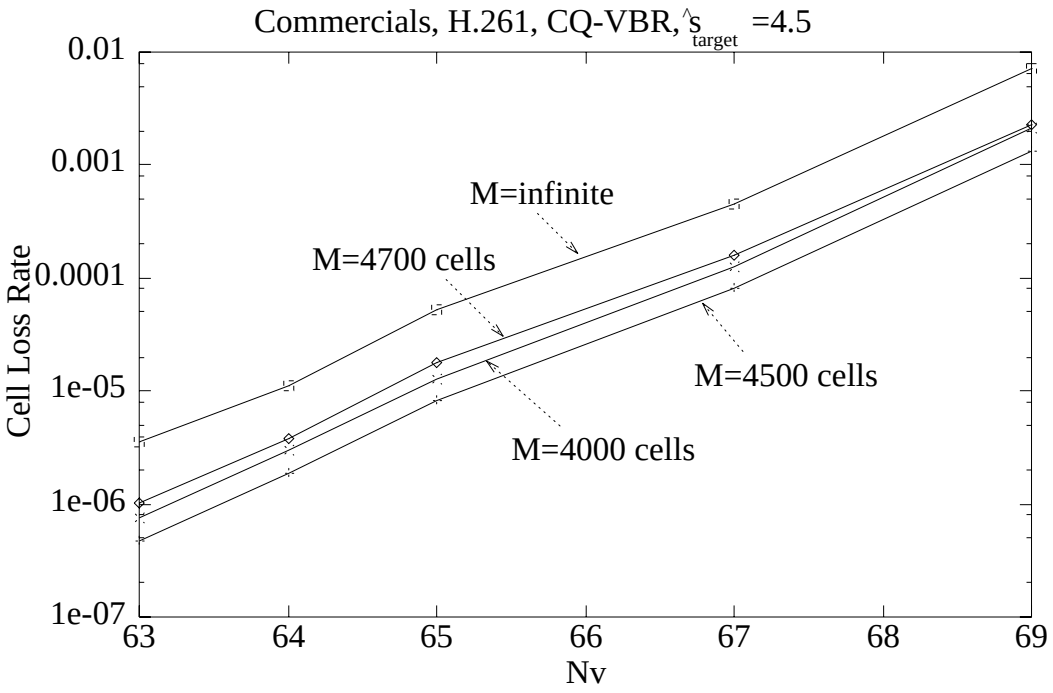
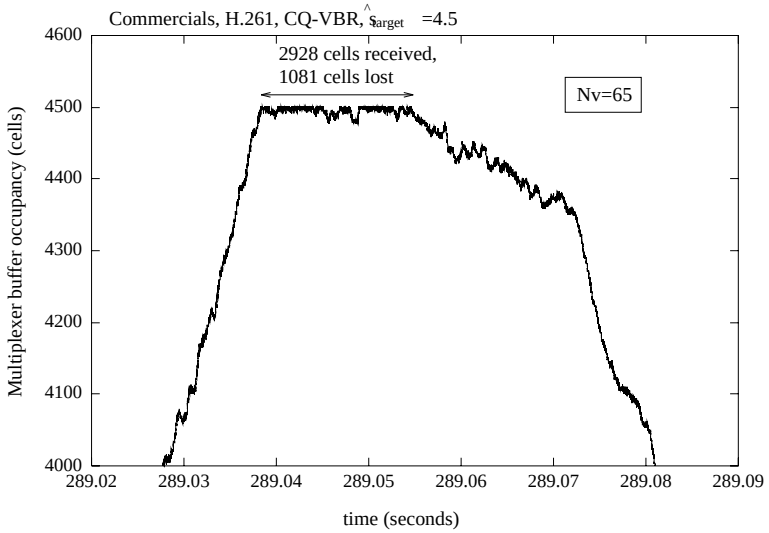
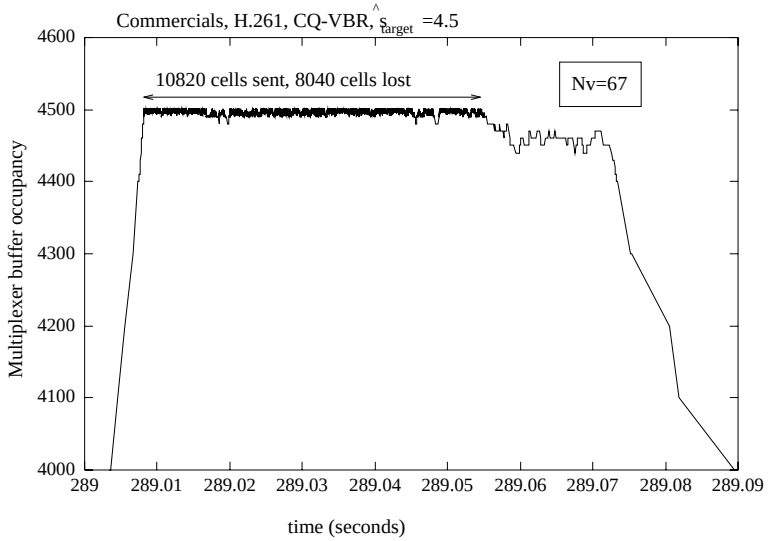


Figure 15: Cell loss rate as a function of N_v for various buffer sizes for the *commercials* sequence, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, N_v=65, D_{max}=25 ms, W=100 Mb/s.



(a) N_v=65



(a) N_v=67

Figure 16: Buffer occupancy during a congested period for the *commercials* sequence, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, W=100 Mb/s.

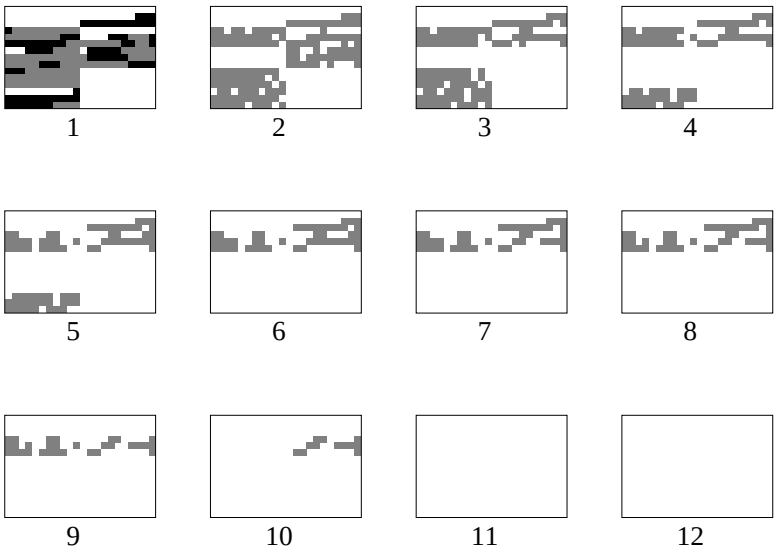


Figure 17: An example glitch for ATM, W=100 Mb/s, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 65$

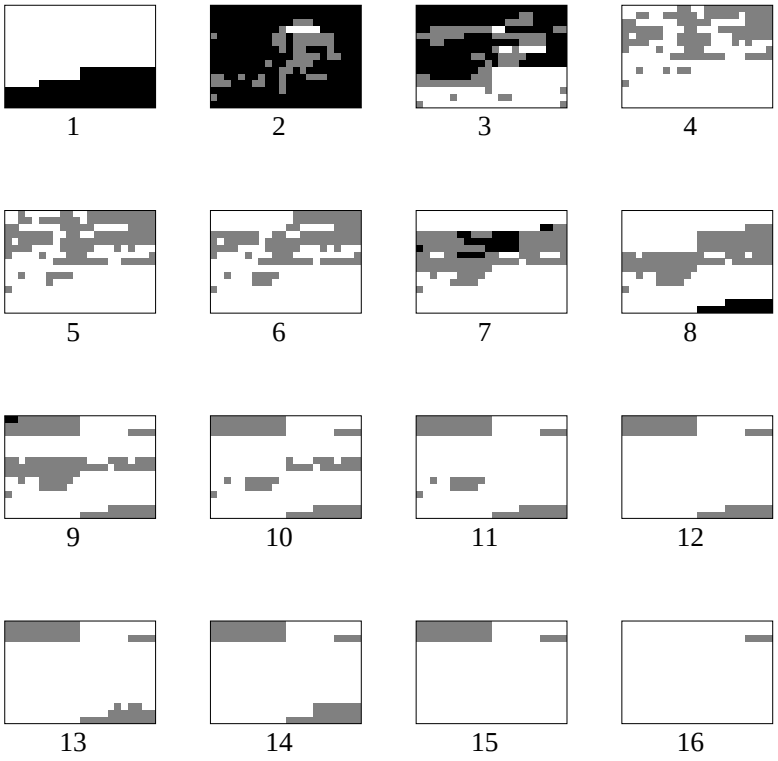


Figure 18: An example glitch for ATM, W=100 Mb/s, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 67$

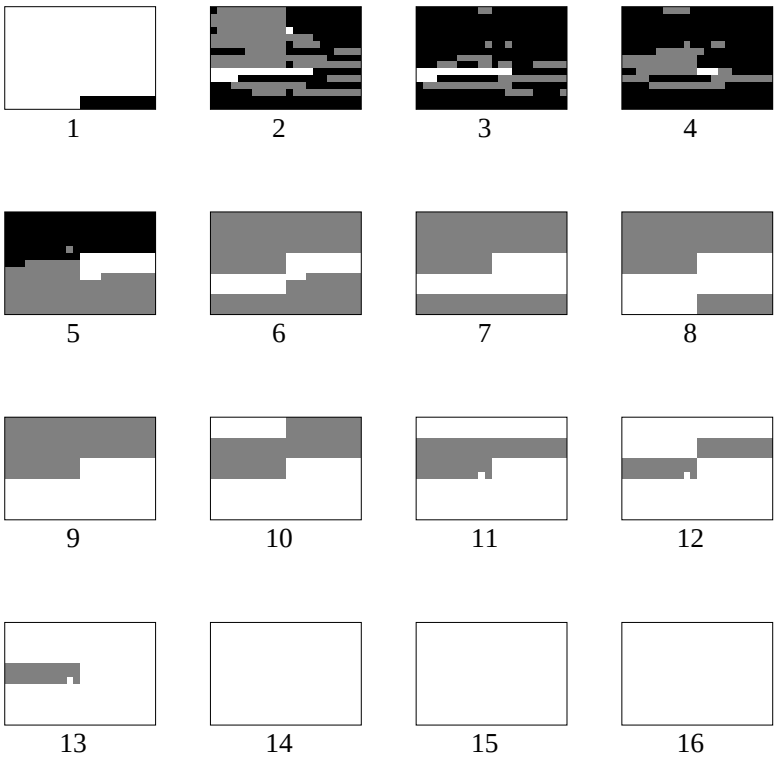


Figure 19: An example glitch for ATM, W=100 Mb/s, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 69$

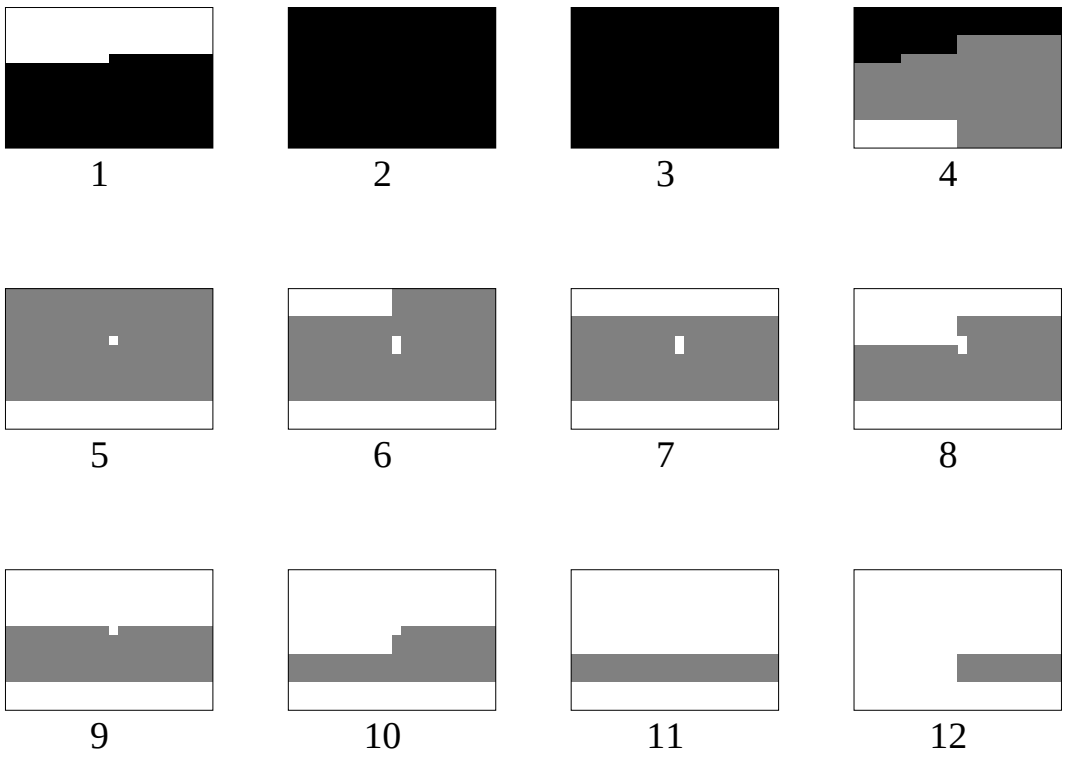


Figure 20: An example glitch for 10Base-T, H.261, V=384 kb/s, D_{max}=100 ms, $N_v = 18$

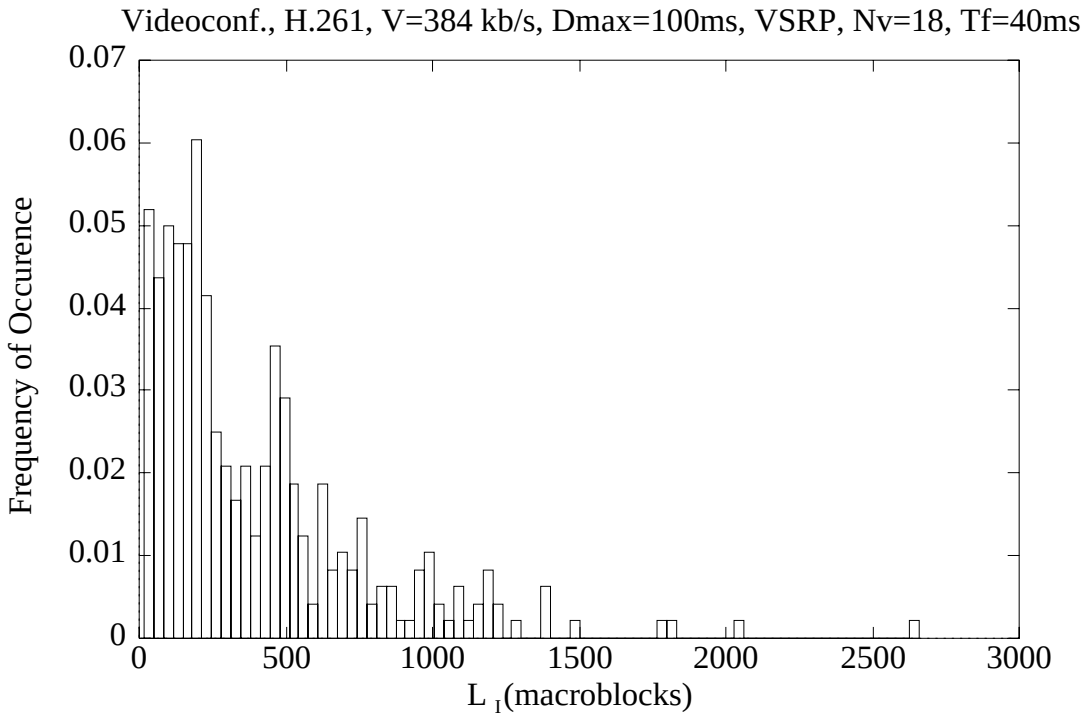


Figure 21: Histogram of L_I for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

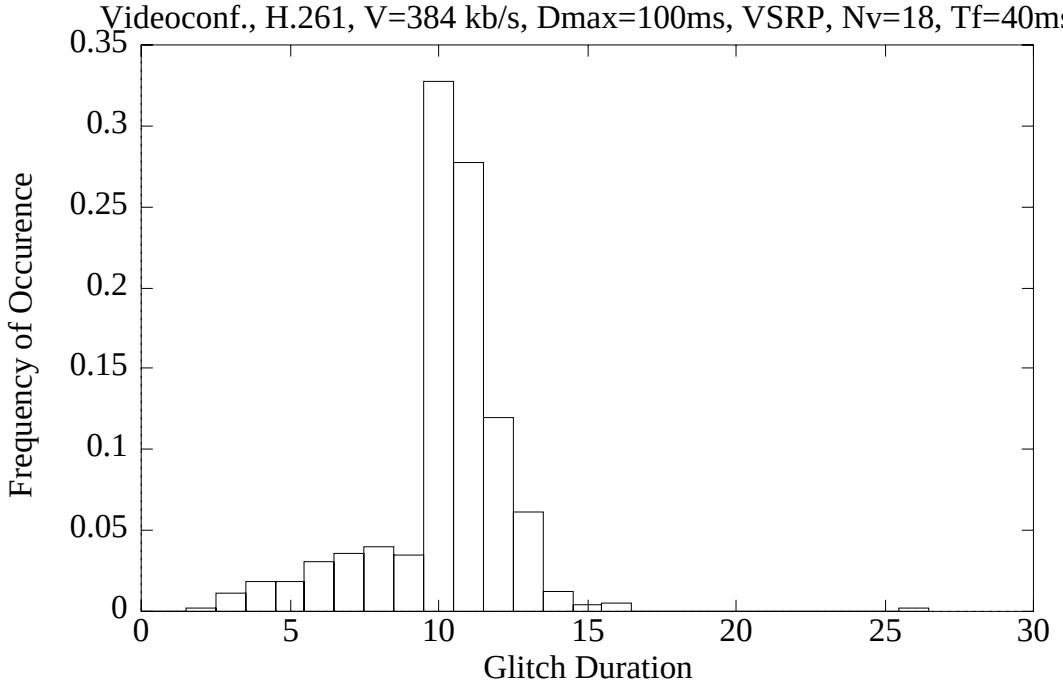


Figure 22: Histogram of glitch durations for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

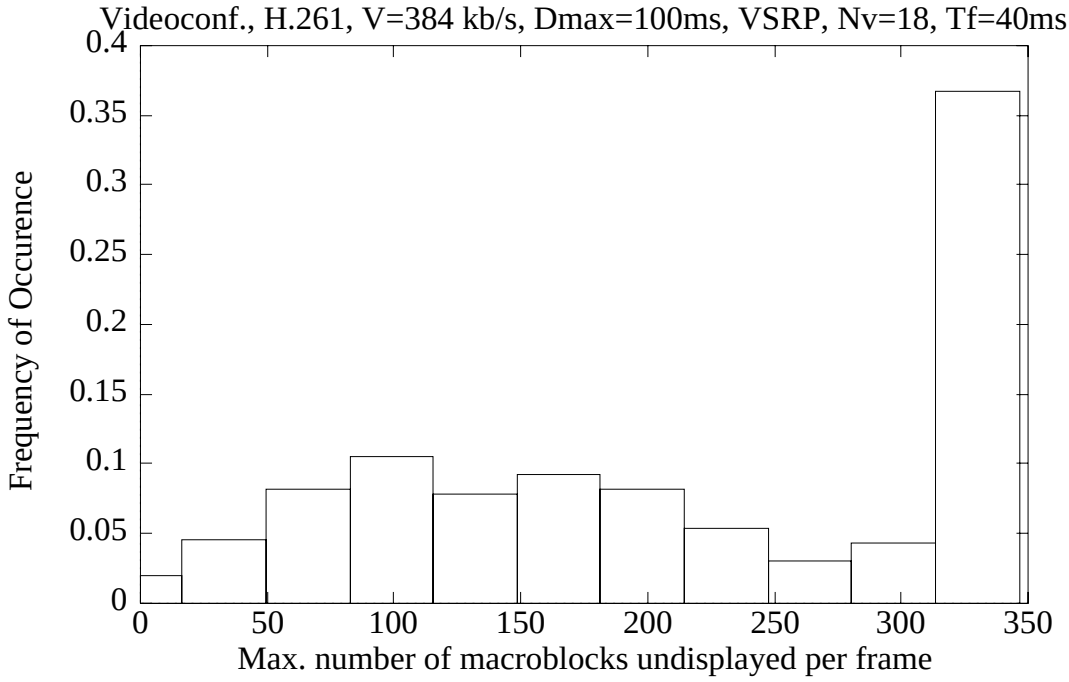


Figure 23: Histogram of A_{max} for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

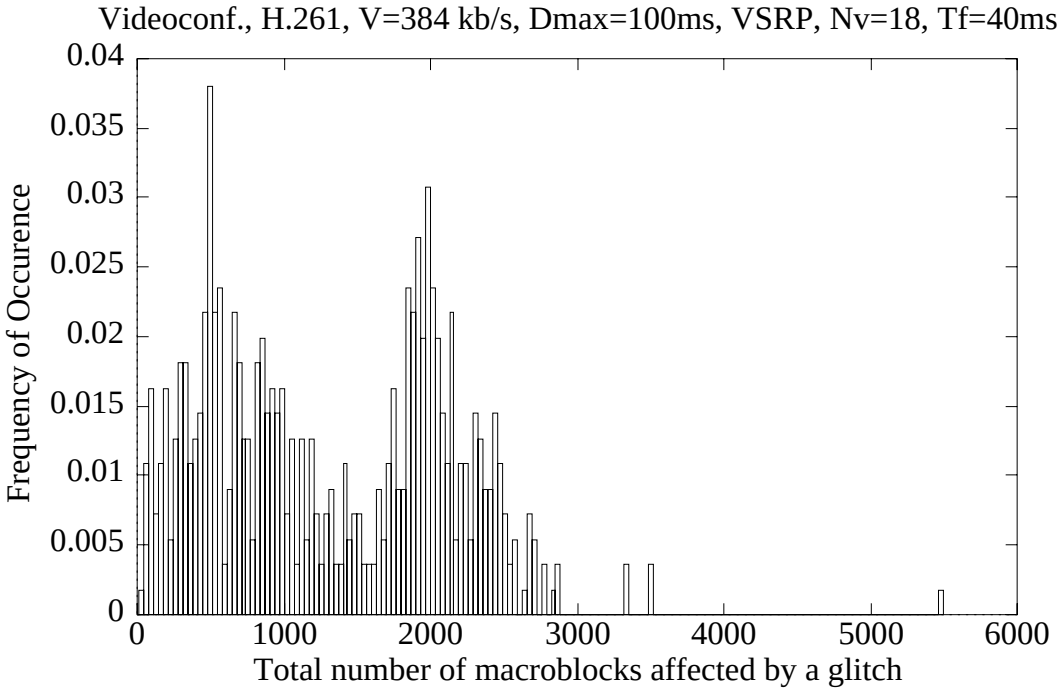


Figure 24: Histogram of L_{tot} for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

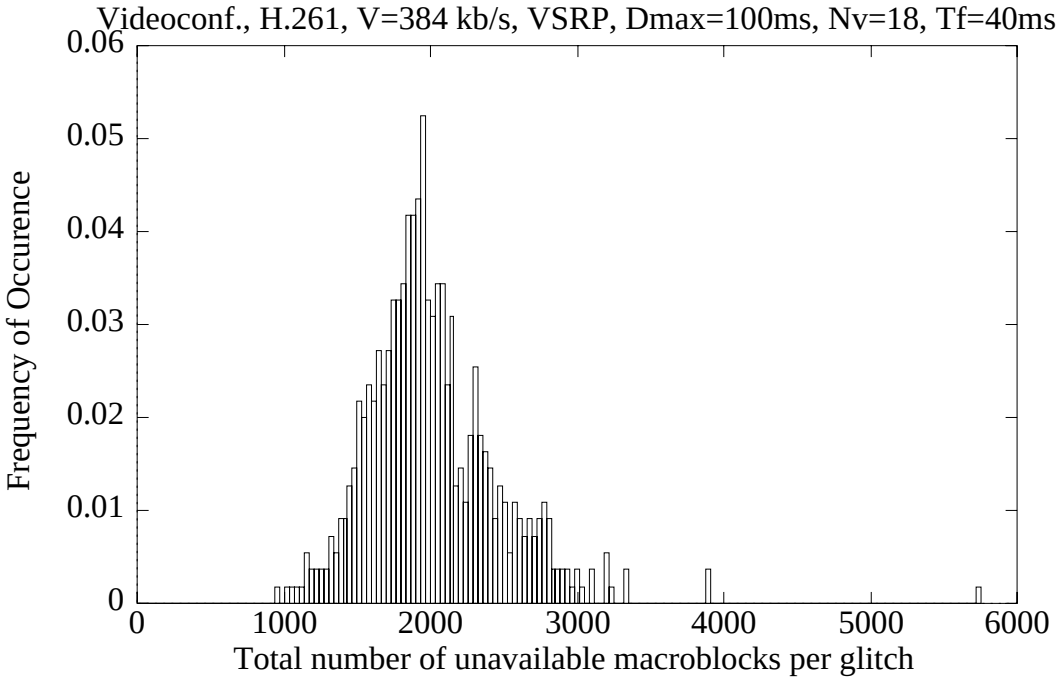


Figure 25: Histogram of L_{tot} when entire packets are dropped, for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

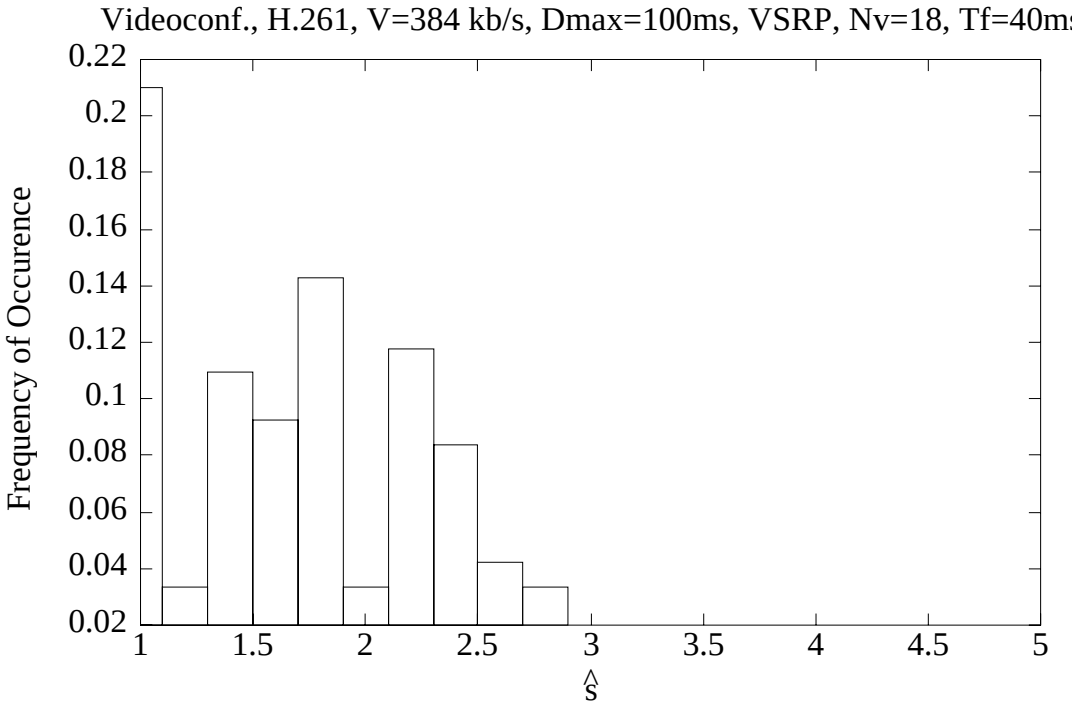


Figure 26: Histogram of $\hat{s}$ for 10Base-T, H.261, CBR, V=384 kb/s, D_{max}=100 ms, T_f=40 ms, N_v=18.

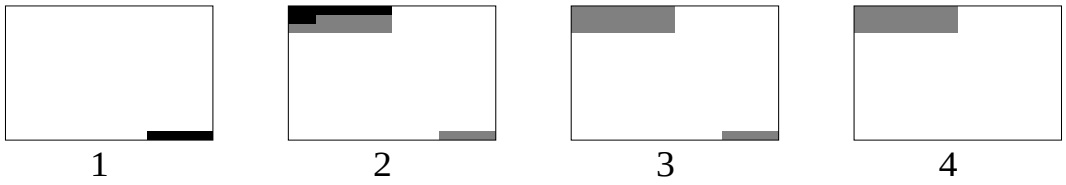


Figure 27: An example glitch for 10Base-T, H.261, V=1536 kb/s, D_{max}=25 ms, $N_v = 4$

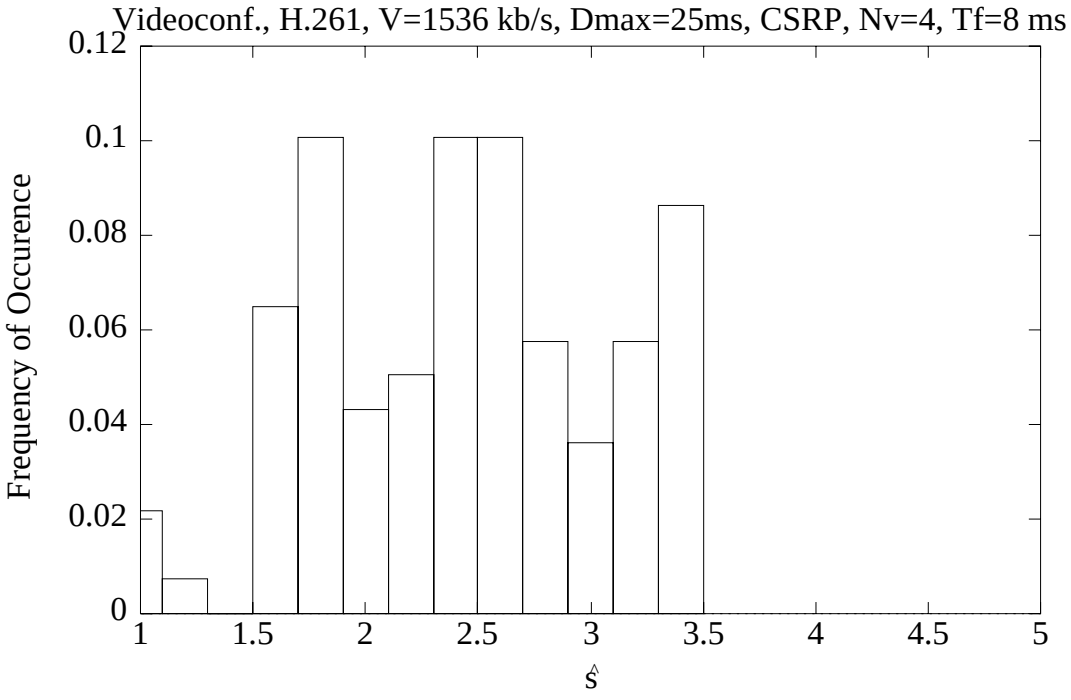


Figure 28: Histogram of $\hat{s}$ for 10Base-T, H.261, V=1536 kb/s, D_{max}=25 ms, T_f=8 ms, N_v=4.

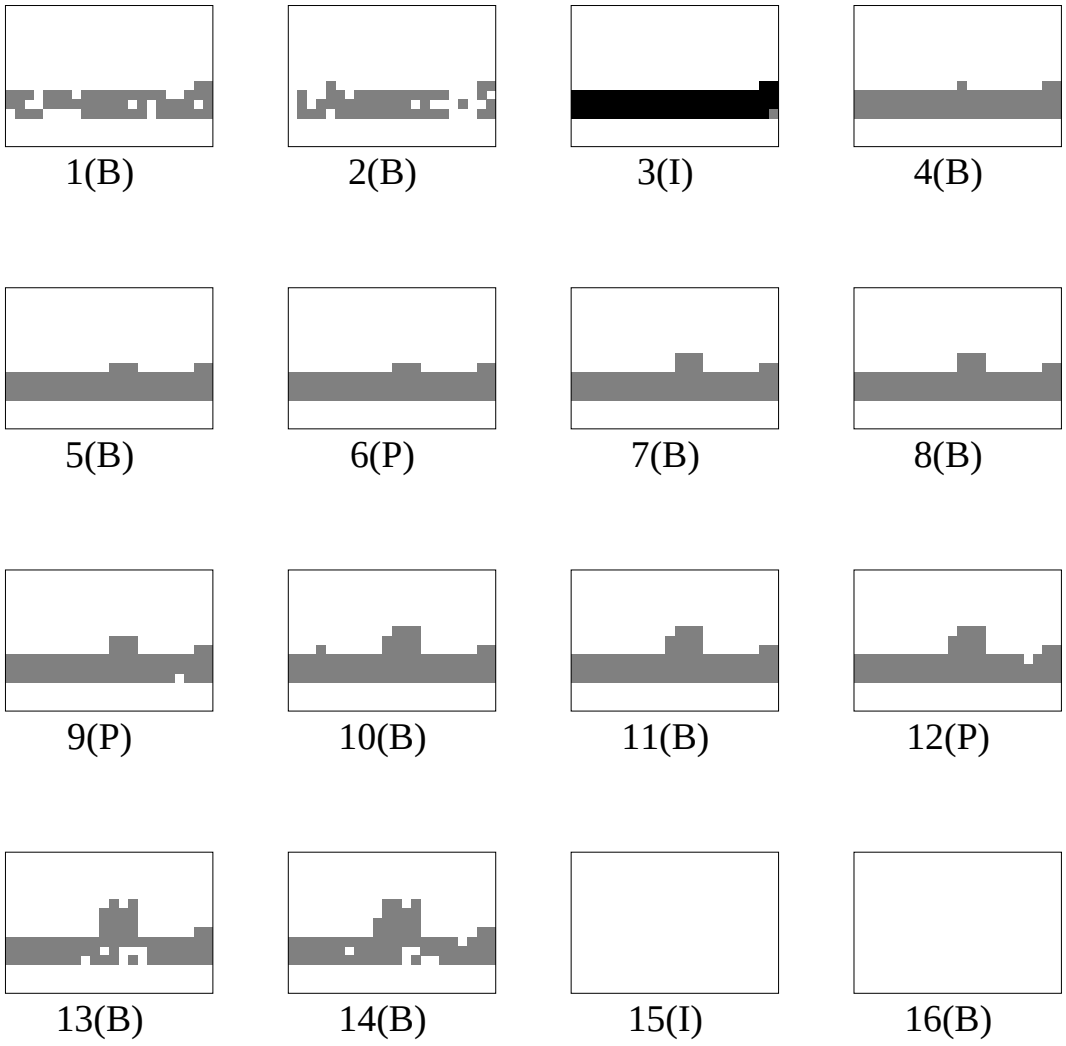


Figure 29: An example glitch for MPEG, V=384 kb/s, D_{max}=100 ms, N_v = 18, 10Base-T

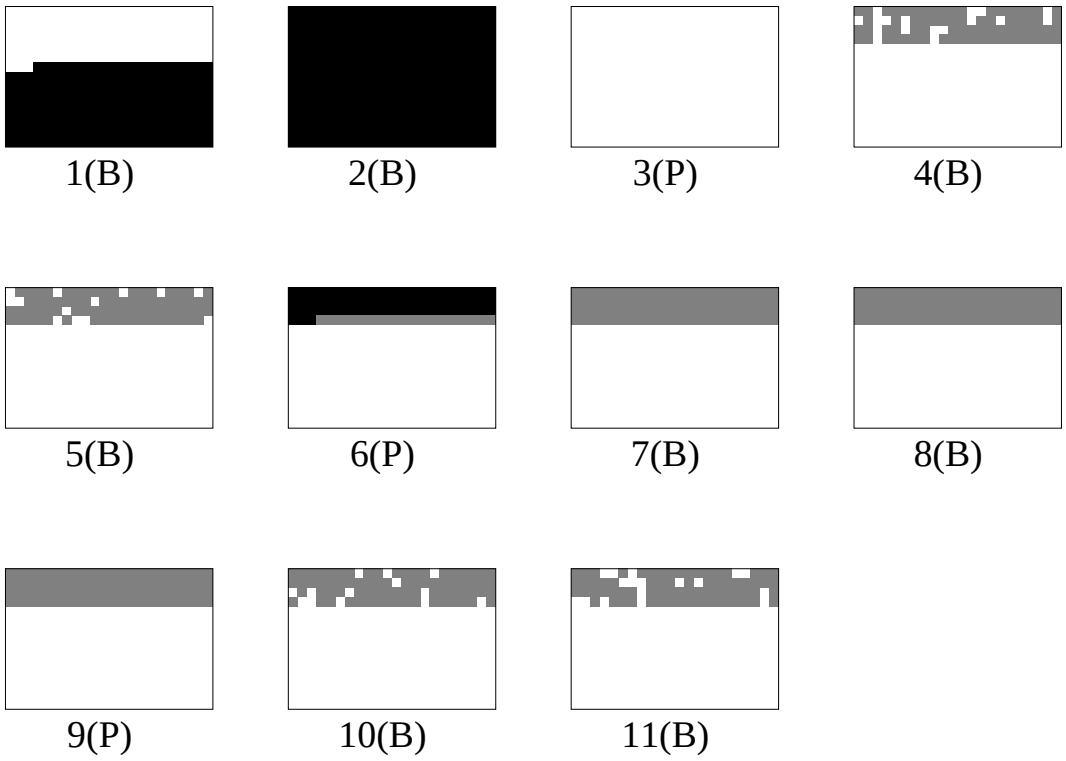


Figure 30: A second example glitch for MPEG, V=384 kb/s, D_{max}=100 ms, N_v = 18, 10Base-T

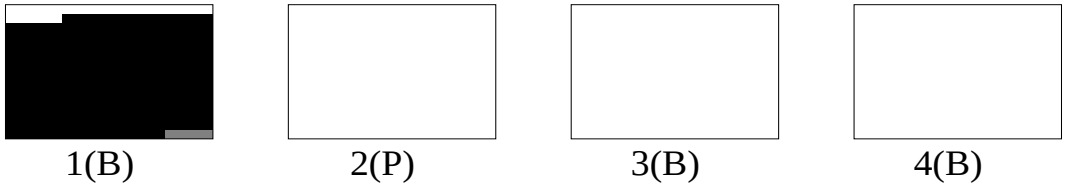


Figure 31: A third example glitch for MPEG, V=384 kb/s, D_{max}=100 ms, N_v = 18, 10Base-T

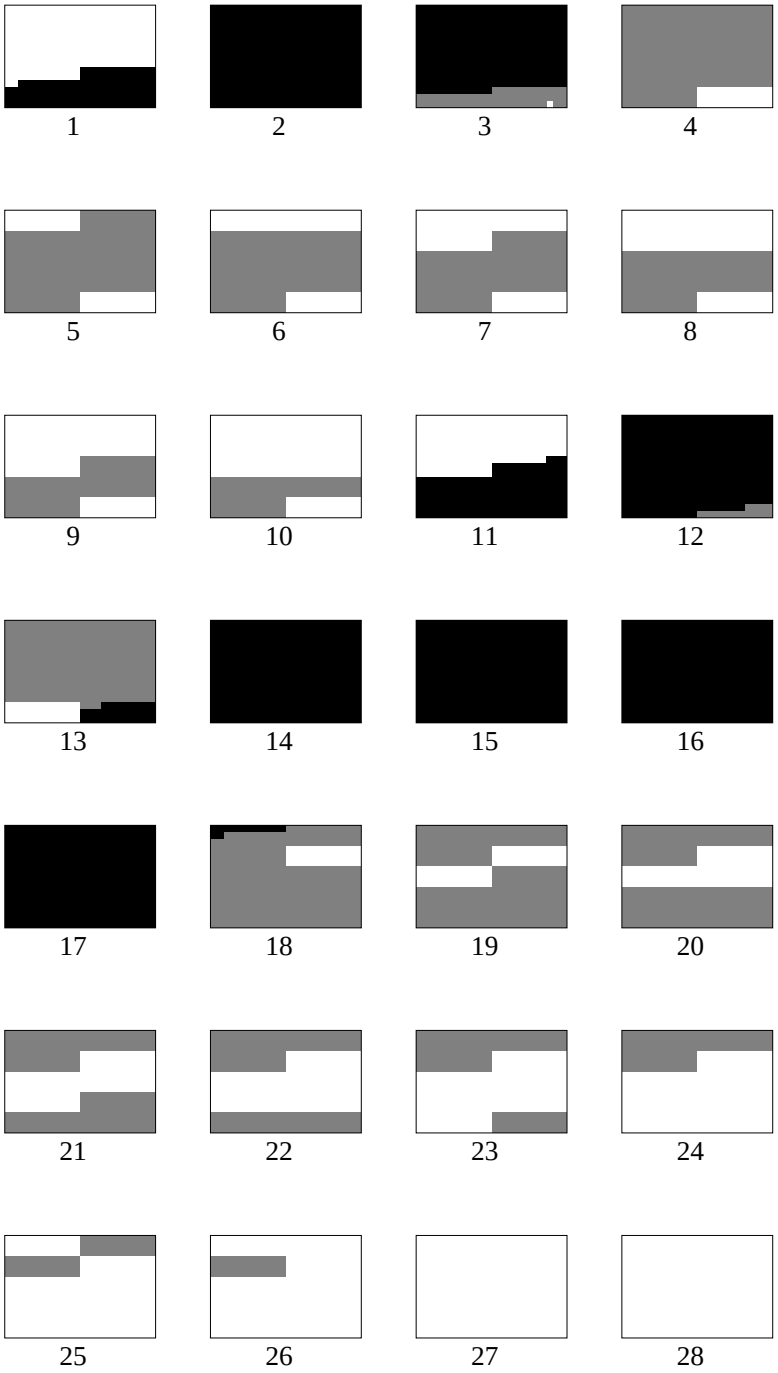
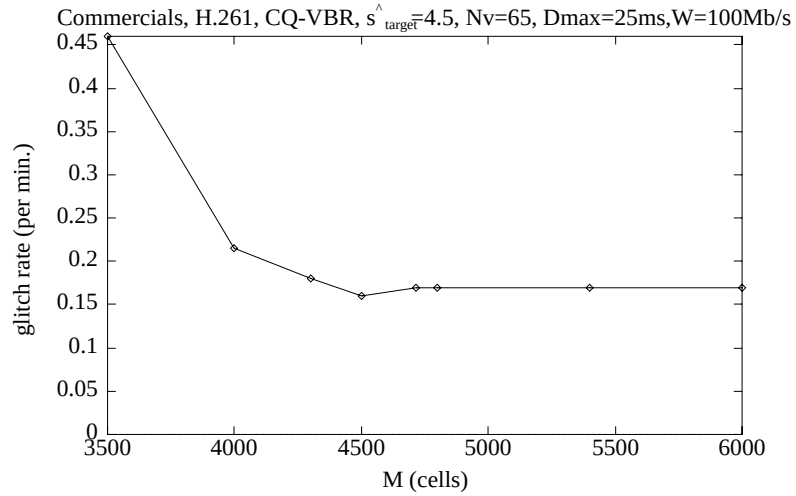


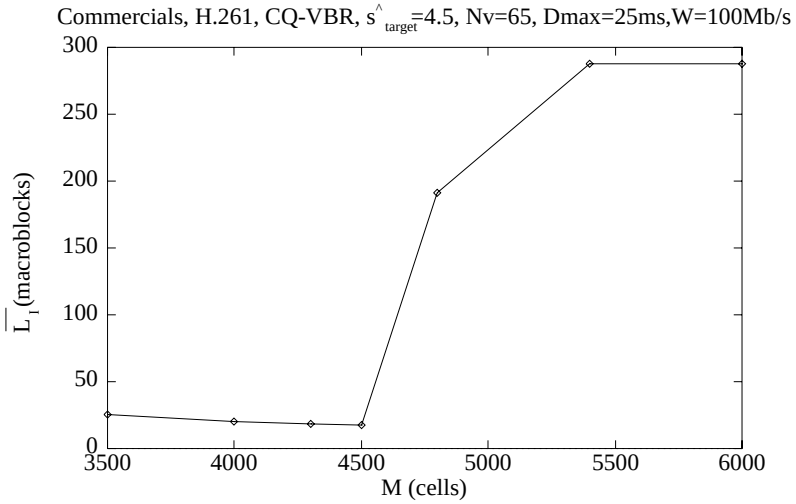
Figure 32: A glitch prolonged by packet losses in the middle (H.261, V=384 kb/s, D_{max}=100 ms, N_v = 18, 10Base-T).

Content	Enc. Sch.	Ctrl.Sch.	N_v (Net. Utl.)	D_{max} (ms)	L	g (per min.)	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$
Comm.	H.261	CQVBR	**44** **(46%)**	25	4.6×10^{-5}	0.08	7.6	90	398
Comm.	H.261	CQVBR	**51** **(54%)**	25	3.4×10^{-4}	0.92	7.4	99	448
Comm.	H.261	CQVBR	81 (85%)	**500**	3.1×10^{-3}	0.96	30.0	319	7328
Comm.	H.261	**CBR**	30 (54%)	25	2.9×10^{-4}	0.87	6.9	89	390
Comm.	H.261	**OLVBR**	33 (54%)	25	3.1×10^{-4}	0.91	6.8	93	413
Comm.	**MPEG**	CQVBR	34 (54%)	125	1.4×10^{-4}	0.97	4.7	93	321
Vconf.	H.261	CQVBR	115 (51%)	25	2.3×10^{-5}	0.09	7.8	96	527
Vconf.	H.261	CQVBR	127 (57%)	25	1.3×10^{-4}	0.94	7.7	98	531

Table 5: Glitch statistics for 100Base-T, various traffic scenarios.



(a) Glitch rate versus switch output buffer size



(b) Average number of macroblocks lost in the cells that cause a glitch versus switch output buffer size

Figure 33: Effect of switch output buffer size on glitching, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 65$, ATM, W=100 Mb/s

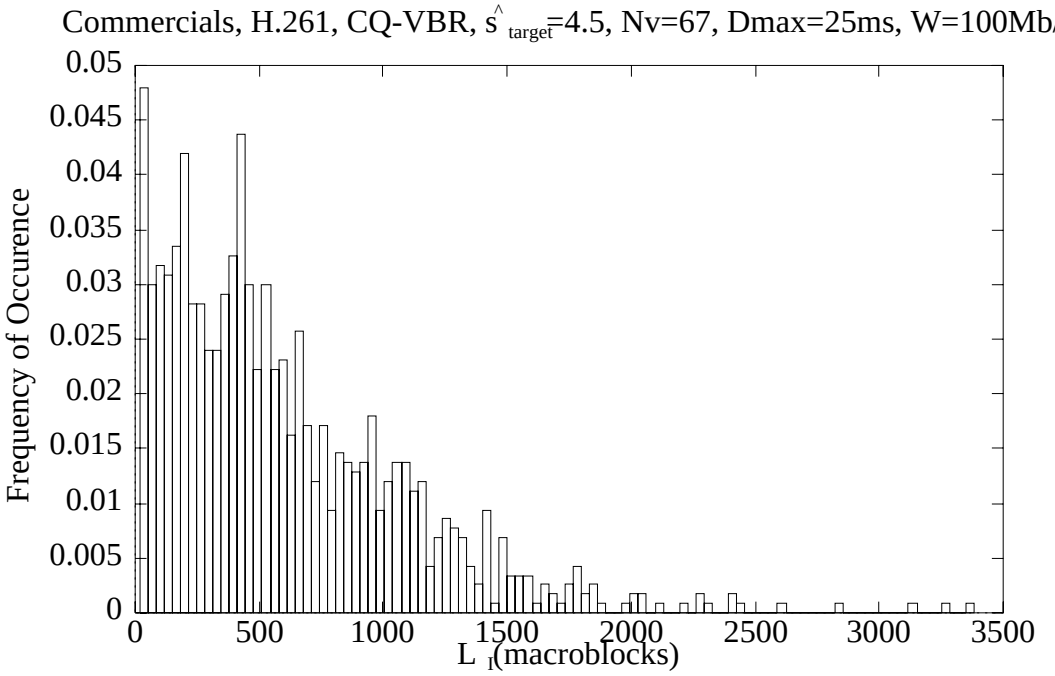


Figure 34: Histogram of L_I for H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 67$, ATM, W=100 Mb/s

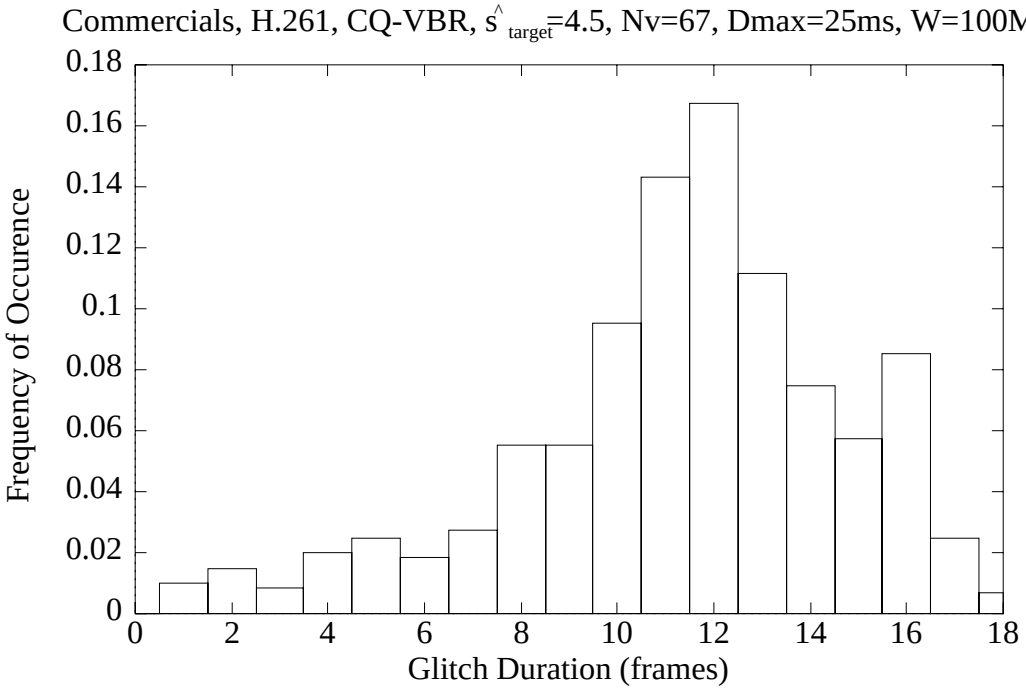


Figure 35: Histogram of glitch duration for H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 67$, ATM, W=100 Mb/s

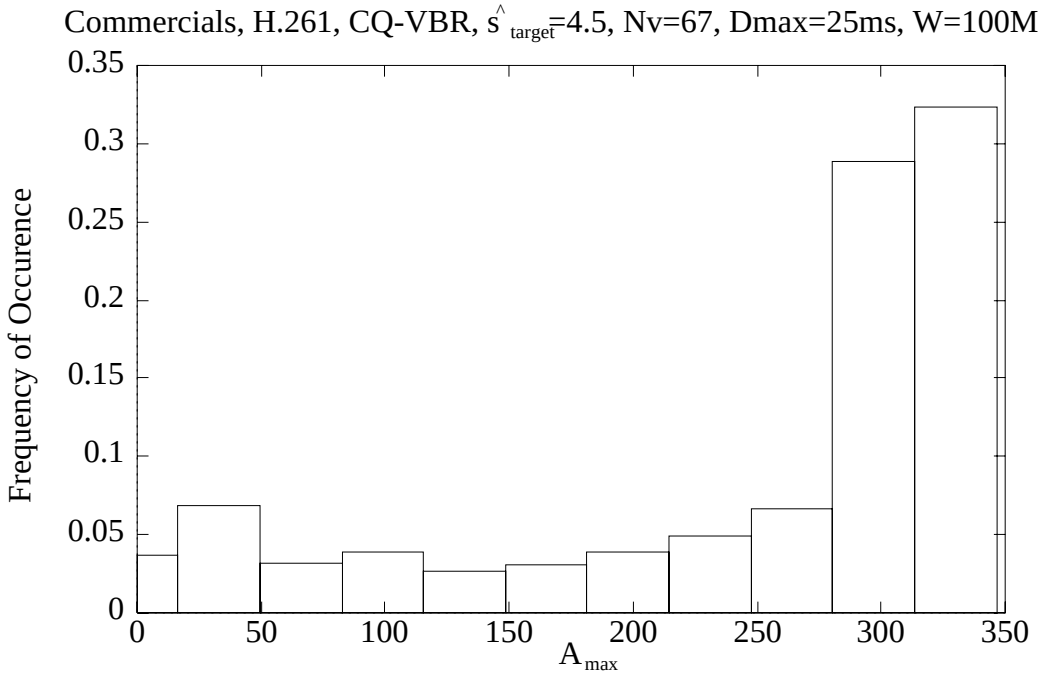


Figure 36: Histogram of A_{max} for H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 67$, ATM, W=100 Mb/s

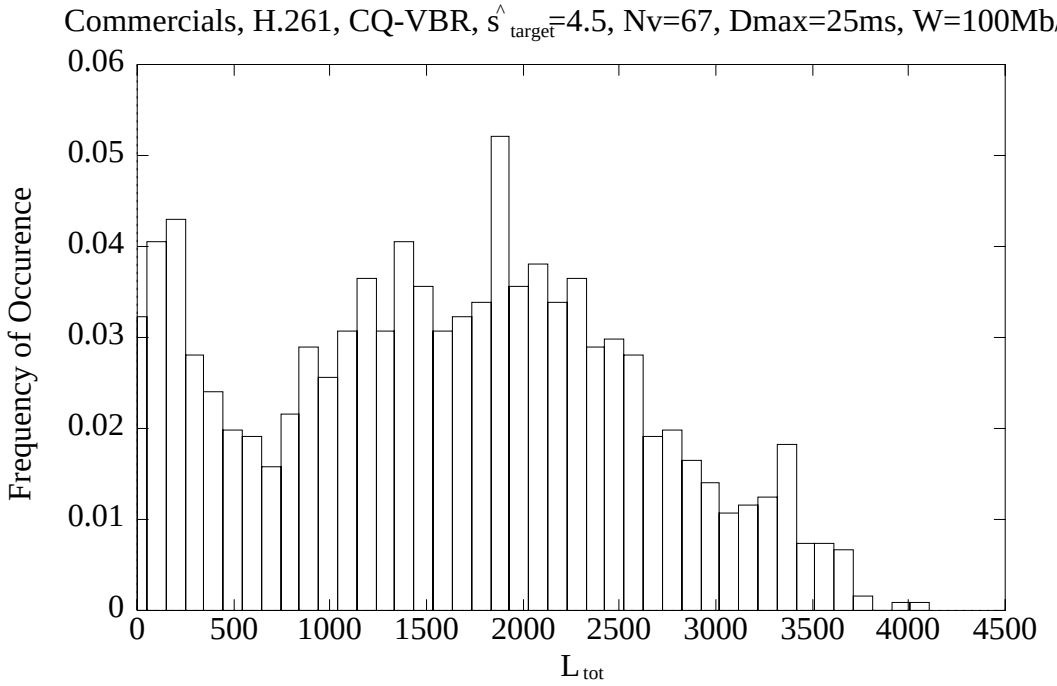


Figure 37: Histogram of L_{tot} for H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, $N_v = 67$, ATM, W=100 Mb/s

Content	Enc.Sch.	Ctrl.Sch.	N_v (Net. Utl.)	D_{max}	L	g (per min.)	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$
Comm.	H.261	CQVBR	**65 (68%)**	25ms	1.8×10^{-5}	0.17	7.8	118	556
Comm.	H.261	CQVBR	**67 (70%)**	25ms	5.5×10^{-4}	0.90	11.3	260	1644
Comm.	H.261	CQVBR	**69 (72%)**	25ms	2.0×10^{-3}	2.3	13.0	289	2087
Comm.	H.261	CQVBR	75 (79%)	**500ms**	4.0×10^{-3}	0.58	37.2	313	9176
Comm.	H.261	**OLVBR**	43 (70%)	25ms	6.1×10^{-4}	0.96	10.4	252	1463
Comm.	**MPEG**	CQVBR	48 (70%)	125ms	7.5×10^{-4}	1.3	7.8	215	1052
Vconf.	H.261	CQVBR	194 (85%)	25ms	4.0×10^{-4}	0.20	12.0	246	1793
Vconf.	H.261	CQVBR	195 (86%)	25ms	1.9×10^{-3}	0.99	16.0	294	3318

Table 6: Glitch statistics for ATM, various traffic scenarios.

	Videoconf.	Star Trek	Commercials
V=384 kb/s	38.4	192	XXXX
V=1536 kb/s	38.4	76.8	768

Table 7: B (in kb/s) at which the encoded video quality reaches a plateau.

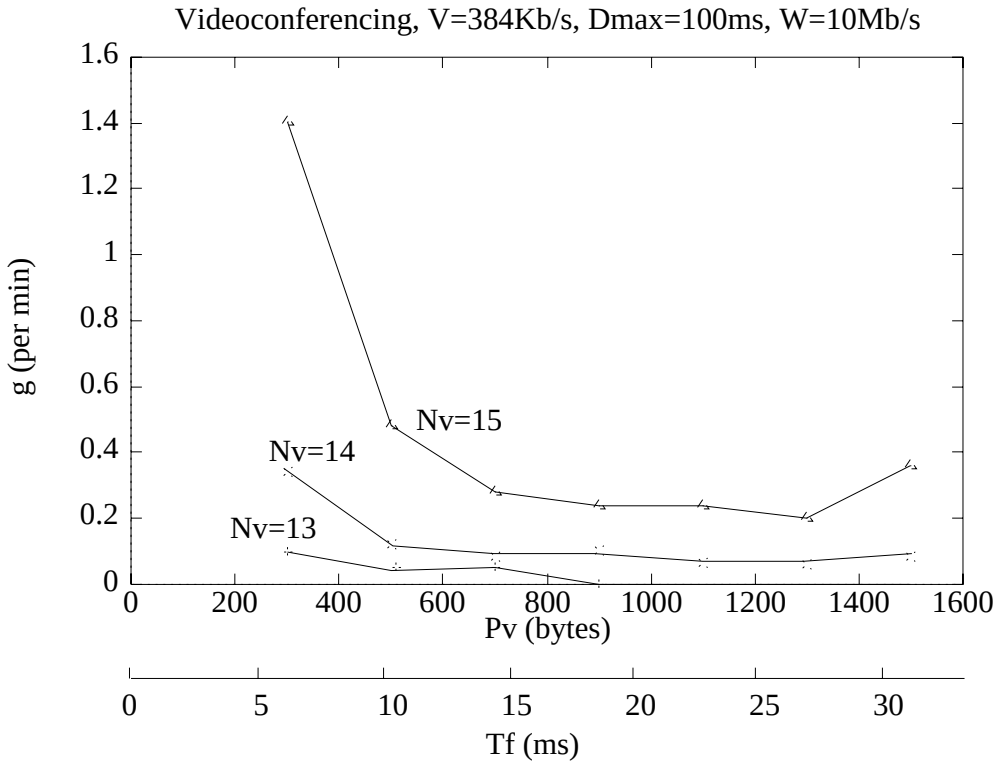


Figure 38: g versus P_v for the *Videoconferencing* sequence, CSRP, W=10 Mb/s, V=384 Kb/s, D_{max}=100ms

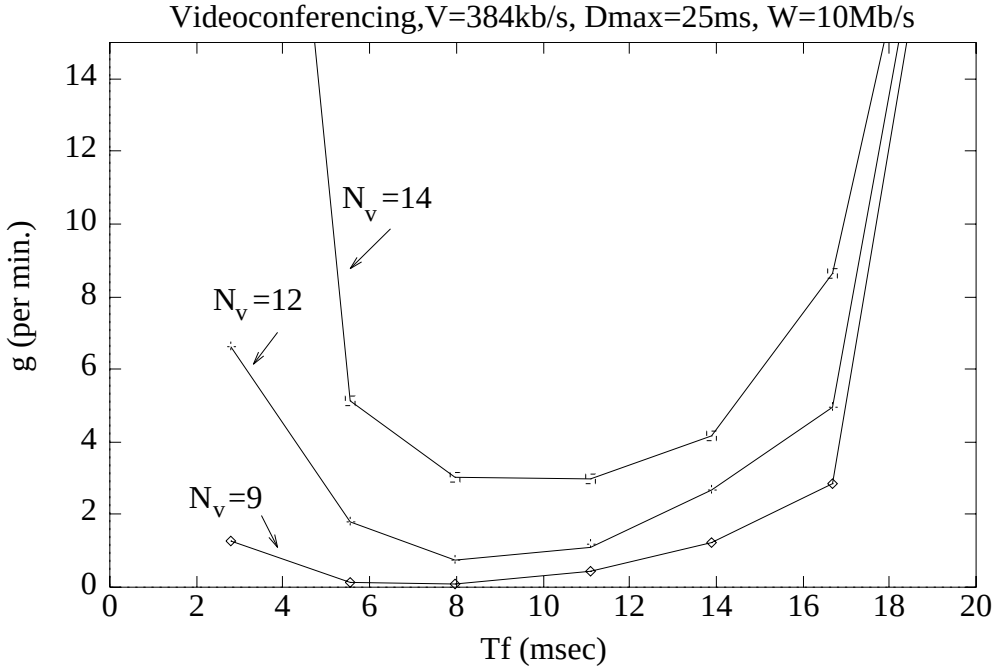


Figure 39: g versus T_f for the *Videoconferencing* sequence, VSRP, W=10 Mb/s, V=384 Kb/s, Dvmax=25ms

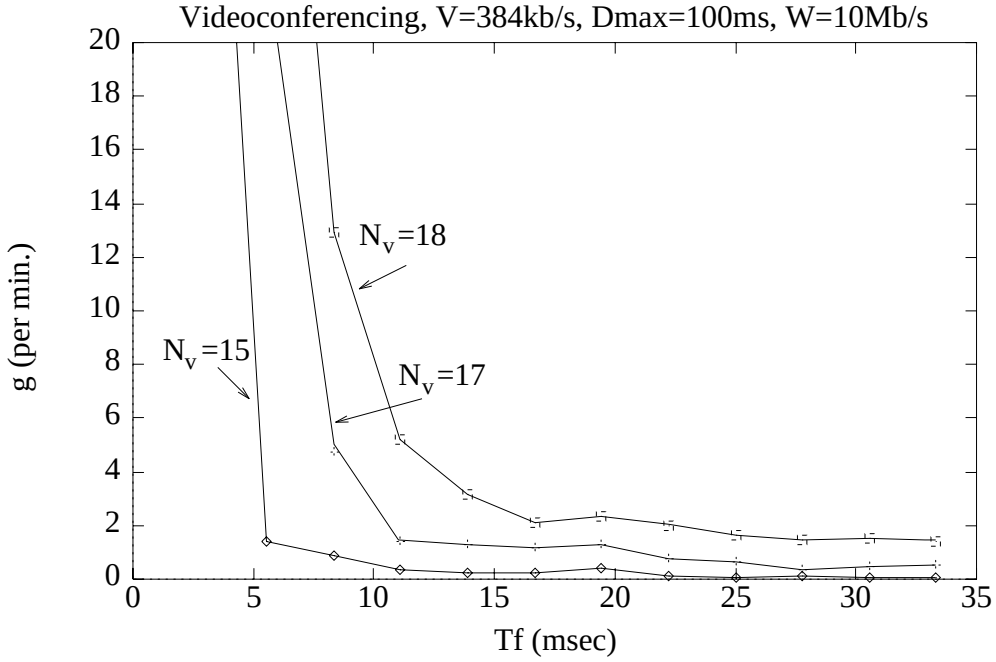


Figure 40: g versus T_f for the *Videoconferencing* sequence, VSRP, W=10 Mb/s, V=384 Kb/s, Dvmax=100ms

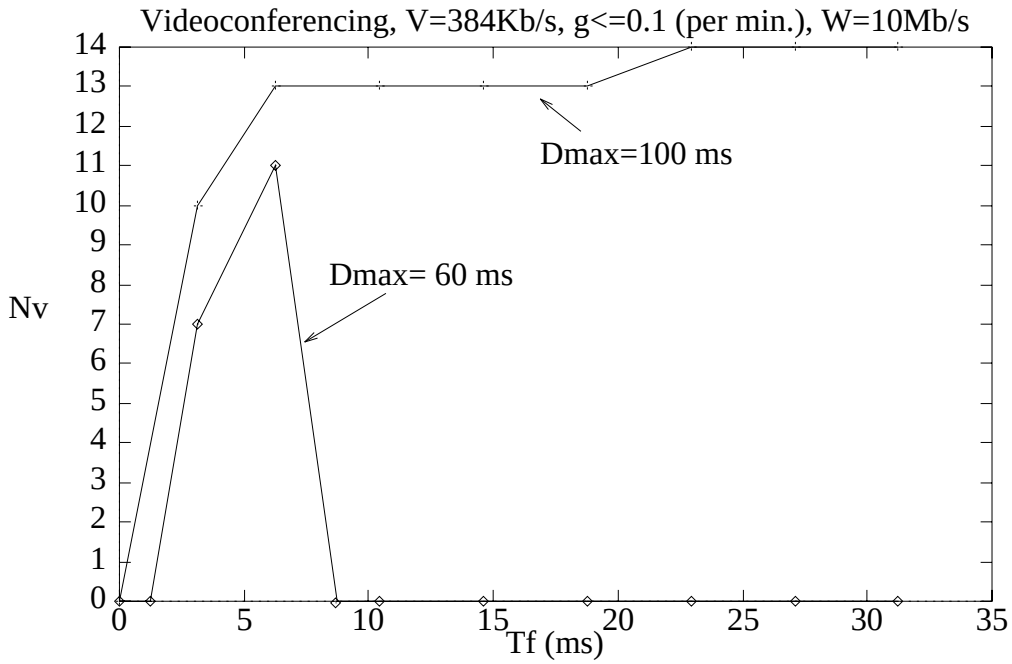


Figure 41: N_v versus T_f for CSRP, W=10 Mb/s, Videoconferencing, V=384 kb/s, g_{max}=0.1 per min.

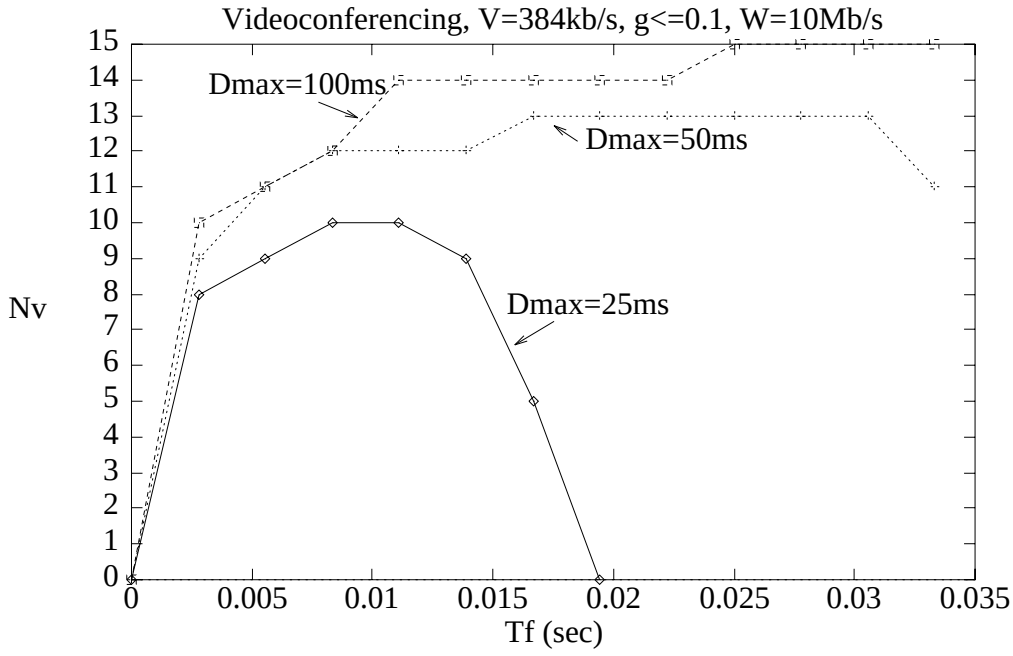


Figure 42: N_v versus T_f for VSRP, W=10 Mb/s, Videoconferencing, V=384 kb/s, g_{max}=0.1 per min.

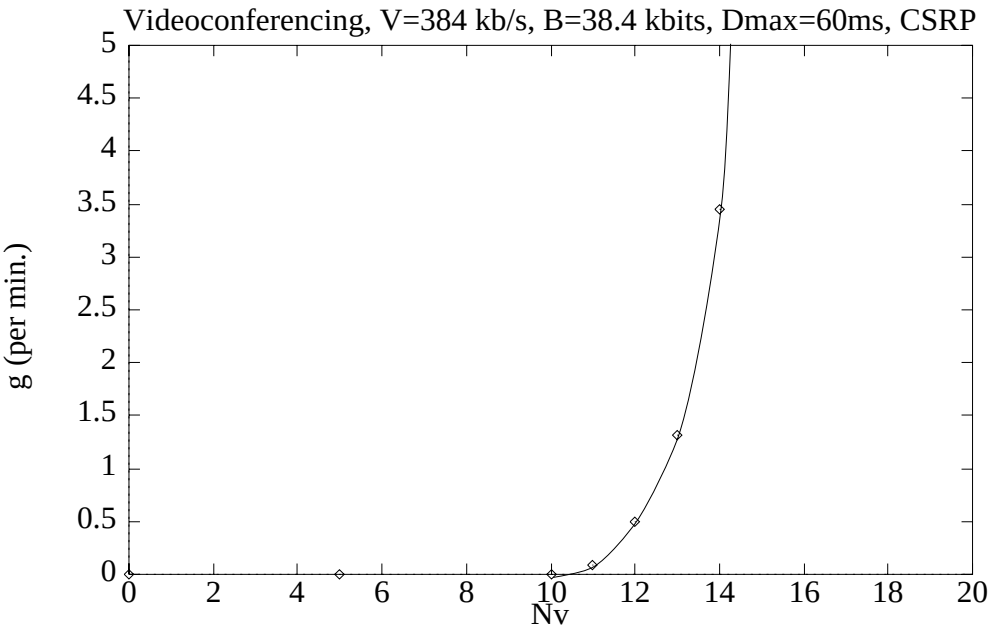


Figure 43: g versus N_v for CSRP, V=384 kb/s, D_{max}=60 ms, T_f=3 ms, 10Base-T

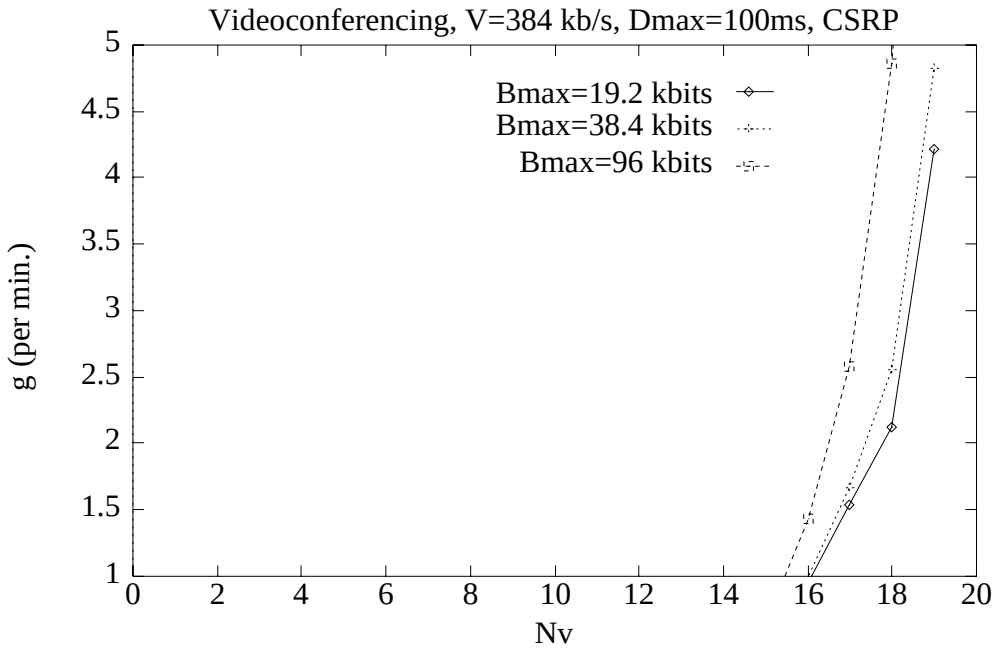


Figure 44: g versus N_v for CSRP, V=384 kb/s, D_{max}=100 ms, 10Base-T

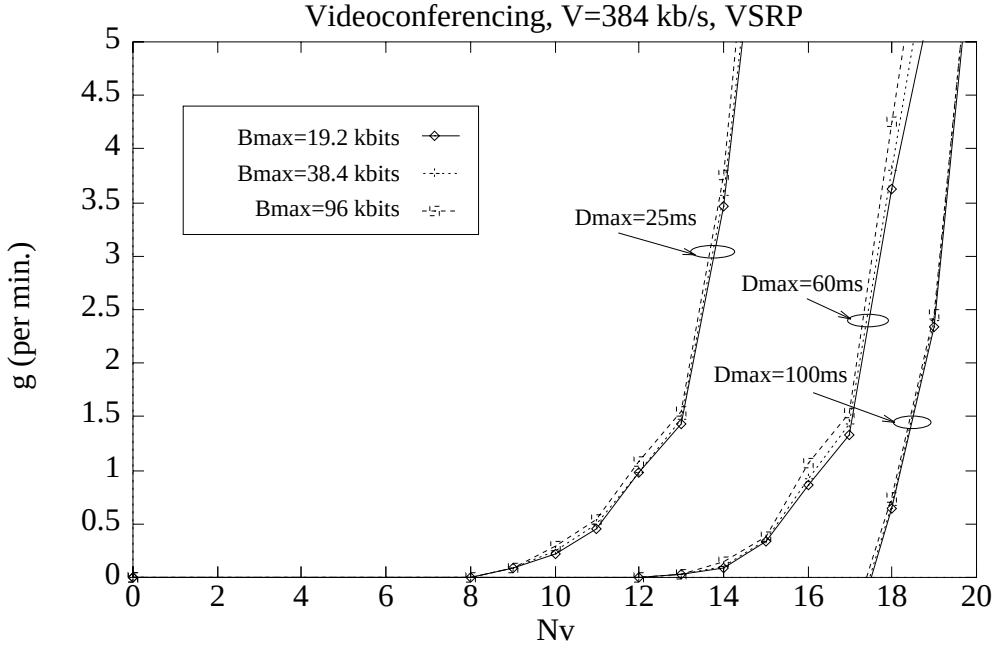


Figure 45: g versus N_v for VSRP, V=384 kb/s, D_{max}={25,60,100} ms, 10Base-T

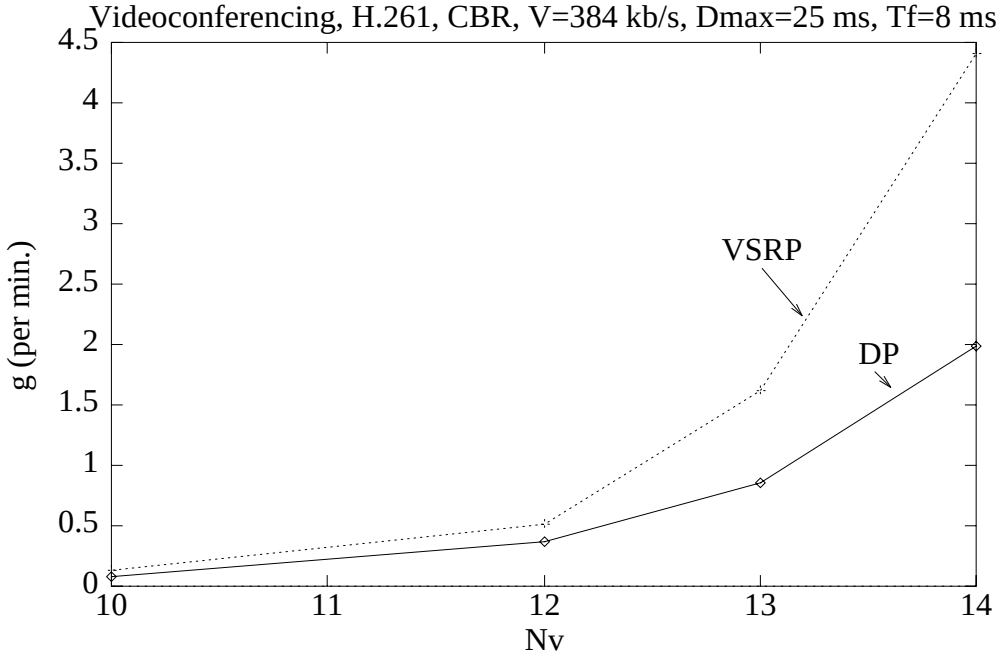


Figure 46: Glitch rate versus N_v for DP and VSRP, *videoconferencing*, H.261, CBR, V=384 kb/s, D_{max}=25 ms.

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)
(ms)	(per min.)	384 kb/s	1536 kb/s	384 kb/s	1536 kb/s	1536 kb/s
25	0.1	0	0	0	0	0
	1	0	0	0	0	0
60	0.1	11 (42%)	4 (61%)	0	4 (61%)	2 (30%)
	1	12 (50%)	5 (77%)	0	5 (77%)	3 (46%)
100	0.1	14 (54%)	5 (77%)	0	5 (77%)	4 (61%)
	1	16 (61%)	5 (77%)	0	5 (77%)	5 (77%)
250	0.1	18 (69%)	5 (77%)	0	5 (77%)	5 (77%)
	1	19 (73%)	5 (77%)	0	5 (77%)	5 (77%)
500	0.1	20 (77%)	5 (77%)	0	5 (77%)	5 (77%)
	1	21 (80%)	5 (77%)	0	5 (77%)	5 (77%)
600	0.1	21 (80%)	5 (77%)	20 (77%)	5 (77%)	5 (77%)
	1	22 (83%)	5 (77%)	20 (77%)	5 (77%)	5 (77%)
Bandwidth limit		26	6	26	6	6

Table 8: N_{max} and the corresponding network utilization for CBR, CSRP, 10Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)
(ms)	(per min.)	384 kb/s	1536 kb/s	384 kb/s	1536 kb/s	1536 kb/s
25	0.1	9 (36%)	3 (46%)	6 (23%)	3 (46%)	2 (30%)
	1	12 (46%)	3 (46%)	10 (38%)	4 (61%)	2 (30%)
60	0.1	14 (54%)	4 (61%)	10 (38%)	4 (61%)	3 (45%)
	1	16 (61%)	5 (77%)	13 (50%)	5 (77%)	3 (45%)
100	0.1	15 (58%)	5 (77%)	12 (46%)	5 (77%)	3 (45%)
	1	17 (65%)	5 (77%)	16 (61%)	5 (77%)	4 (61%)
250	0.1	19 (73%)	5 (77%)	18 (69%)	5 (77%)	4 (61%)
	1	20 (77%)	5 (77%)	19 (73%)	5 (77%)	4 (61%)
500	0.1	21 (80%)	5 (77%)	21 (80%)	5 (77%)	5 (77%)
	1	22 (83%)	5 (77%)	22 (83%)	5 (77%)	5 (77%)
Bandwidth limit		26	6	26	6	6

Table 9: N_{max} and the corresponding network utilization for CBR, VSRP, 10Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)
(ms)	(per min.)	384 kb/s	1536 kb/s	384 kb/s	1536 kb/s	1536 kb/s
25	0.1	0	0	0	0	0
	1	0	0	0	0	0
60	0.1	115 (44%)	39 (60%)	0	39 (60%)	21 (32%)
	1	126 (48%)	41 (63%)	0	41 (63%)	24 (37%)
100	0.1	210 (81%)	53 (81%)	0	52 (80%)	36 (56%)
	1	215 (83%)	55 (85%)	0	54 (83%)	40 (62%)
250	0.1	221 (85%)	56 (86%)	0	56 (86%)	56 (86%)
	1	225 (86%)	56 (86%)	0	56 (86%)	56 (86%)
500	0.1	221 (85%)	56 (86%)	0	56 (86%)	56 (86%)
	1	225 (86%)	56 (86%)	0	56 (86%)	56 (86%)
600	0.1	221 (85%)	56 (86%)	221 (85%)	56 (86%)	56 (86%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	56 (86%)
Bandwidth limit		260	65	260	65	65

Table 10: N_{max} and the corresponding network utilization for CBR, CSRP, 100Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)
(ms)	(per min.)	384 kb/s	1536 kb/s	384 kb/s	1536 kb/s	1536 kb/s
25	0.1	118 (45%)	31 (47%)	115 (44%)	31 (47%)	31 (47%)
	1	135 (52%)	35 (54%)	131 (50%)	35 (54%)	35 (54%)
60	0.1	211 (81%)	53 (81%)	210 (81%)	53 (81%)	53 (81%)
	1	220 (85%)	55 (85%)	220 (85%)	55 (85%)	55 (85%)
100	0.1	221 (85%)	56 (86%)	220 (85%)	56 (86%)	55 (85%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	55 (85%)
250	0.1	221 (85%)	56 (86%)	220 (85%)	56 (86%)	55 (85%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	55 (85%)
600	0.1	221 (85%)	56 (86%)	220 (85%)	56 (86%)	55 (85%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	55 (85%)
Bandwidth limit		260	65	260	65	65

Table 11: N_{max} and the corresponding network utilization for CBR, VSRP, 100Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)
(ms)	(per min.)	384 kb/s	1536 kb/s	384 kb/s	1536 kb/s	1536 kb/s
25	—	0	0	0	0	0
60	0.1	112 (43%)	28 (43%)	111 (43%)	27 (41%)	27 (41%)
	1	128 (49%)	32 (49%)	124 (48%)	32 (49%)	32 (49%)
100	0.1	221 (85%)	56 (86%)	220 (85%)	56 (86%)	55 (85%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	55 (85%)
500	0.1	221 (85%)	56 (86%)	220 (85%)	56 (86%)	55 (85%)
	1	225 (86%)	56 (86%)	225 (86%)	56 (86%)	55 (85%)
Bandwidth limit		260	65	260	65	65

Table 12: N_{max} and the corresponding network utilization for frame-by-frame operation, CBR, VSRP, 100Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)	
(ms)	(per min.)	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5
25	0.1	11 (30%)	6 (27%)	8 (29%)	4 (25%)	4 (24%)	2 (21%)
	1	13 (35%)	8 (36%)	10 (36%)	5 (32%)	4 (24%)	2 (21%)
60	0.1	13 (35%)	7 (32%)	11 (39%)	4 (25%)	5 (30%)	2 (21%)
	1	16 (43%)	9 (41%)	13 (47%)	6 (38%)	6 (36%)	3 (32%)
100	0.1	15 (40%)	8 (36%)	11 (39%)	5 (32%)	6 (36%)	2 (21%)
	1	19 (51%)	11 (50%)	14 (50%)	7 (44%)	7 (42%)	3 (32%)
250	0.1	18 (49%)	11 (50%)	13 (47%)	6 (38%)	7 (42%)	4 (42%)
	1	22 (59%)	13 (59%)	16 (57%)	8 (50%)	9 (54%)	5 (53%)
500	0.1	24 (65%)	14 (64%)	17 (63%)	8 (50%)	8 (49%)	5 (53%)
	1	26 (70%)	15 (68%)	18 (66%)	9 (57%)	10 (60%)	5 (53%)
Bandwidth limit		37	22	27	15	16	9

Table 13: N_{max} and the corresponding network utilization for CQ-VBR, 10Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)	
(ms)	(per min.)	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5
25	0.1	165 (45%)	115 (51%)	119 (43%)	69 (44%)	75 (45%)	44 (46%)
	1	196 (53%)	127 (57%)	136 (49%)	80 (50%)	89 (54%)	51 (54%)
60	0.1	298 (81%)	184 (83%)	221 (80%)	123 (80%)	134 (81%)	75 (80%)
	1	315 (85%)	190 (85%)	235 (85%)	132 (84%)	140 (85%)	81 (85%)
500	0.1	315 (85%)	188 (85%)	236 (85%)	135 (85%)	140 (85%)	81 (85%)
	1	315 (85%)	190 (85%)	236 (85%)	135 (85%)	140 (85%)	81 (85%)
Bandwidth limit		370	222	277	158	166	95

Table 14: N_{max} and the corresponding network utilization for CQ-VBR, 100Base-T

	Commercials	Star Trek	Videoconf.
CQ-VBR $\hat{s}_{min}$	3.8	3.8	3.8
CQ-VBR Avg. Rate (kb/s)	600	360	270
CBR Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	1200	540	300

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

	Commercials	Star Trek	Videoconf.
CQ-VBR $\hat{s}_{min}$	4.3	4.3	4.3
CQ-VBR Avg. Rate (kb/s)	1050	630	450
CBR Rate (kb/s) for the same $\hat{s}_{min}$ (± 0.1)	1800	660	480

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

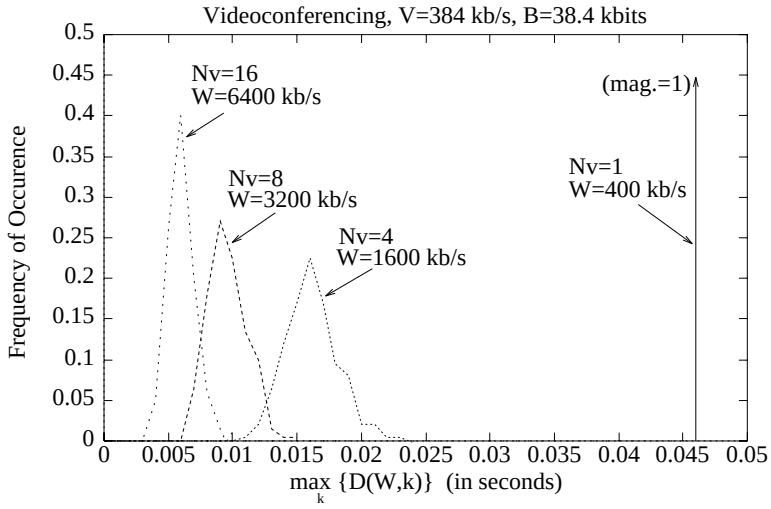
Table 15: Comparison of average bit rates of CQ-VBR and CBR encoded sequences for the same $\hat{s}_{min}$.

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)	
(ms)	(per min.)	300 kb/s	480 kb/s	540 kb/s	660 kb/s	1200 kb/s	1800 kb/s
25	0.1	12 (36%)	8 (38%)	6 (32%)	6 (39%)	3 (36%)	2 (36%)
	1	15 (45%)	9 (43%)	8 (43%)	7 (46%)	3 (36%)	2 (36%)
60	0.1	18 (54%)	11 (53%)	9 (49%)	8 (53%)	4 (48%)	2 (36%)
	1	19 (57%)	13 (62%)	11 (59%)	9 (59%)	5 (60%)	3 (54%)
100	0.1	19 (57%)	12 (58%)	10 (54%)	9 (59%)	5 (48%)	2 (36%)
	1	21 (63%)	14 (67%)	12 (65%)	10 (66%)	5 (48%)	3 (54%)
250	0.1	24 (72%)	15 (72%)	13 (70%)	10 (66%)	5 (48%)	3 (54%)
	1	25 (75%)	16 (77%)	14 (76%)	11 (73%)	6 (72%)	3 (54%)
500	0.1	26 (80%)	16 (81%)	14 (76%)	12 (80%)	6 (72%)	4 (72%)
	1	27 (82%)	16 (81%)	15 (83%)	12 (80%)	6 (72%)	4 (72%)
Bandwidth limit		33	20	18	15	8	5

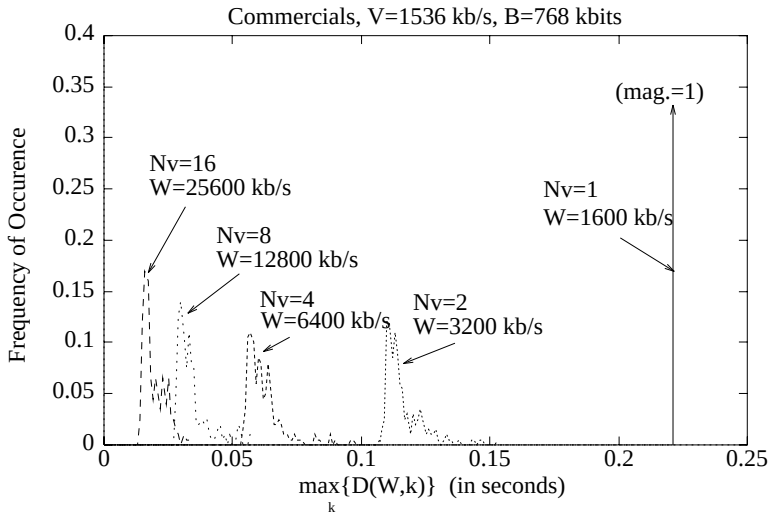
Table 16: N_{max} and the corresponding network utilization for CBR at the same minimum quality as CQ-VBR, 10Base-T

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)	
(ms)	(per min.)	300 kb/s	480 kb/s	540 kb/s	660 kb/s	1200 kb/s	1800 kb/s
25	0.1	153 (46%)	108 (52%)	81 (44%)	70 (46%)	38 (46%)	26 (47%)
	1	176 (53%)	121 (58%)	98 (53%)	81 (53%)	44 (53%)	30 (54%)
60	0.1	268 (81%)	166 (80%)	147 (80%)	122 (81%)	66 (80%)	43 (78%)
	1	280 (84%)	176 (85%)	156 (85%)	128 (85%)	70 (85%)	46 (83%)
500	0.1	282 (85%)	176 (85%)	156 (85%)	128 (85%)	70 (85%)	46 (83%)
	1	284 (85%)	176 (85%)	156 (85%)	129 (85%)	70 (85%)	47 (85%)
Bandwidth limit		333	208	185	151	83	55

Table 17: N_{max} and the corresponding network utilization for CBR at the same minimum quality as CQ-VBR, 100Base-T



(a) Videoconferencing, V=384 kb/s, B=38.4 kbits



(b) Commercials, V=1536 kb/s, B=768 kbits

Figure 47: Histogram of $\max_k\{D(W,k)\}$ for Videoconferencing and Commercials, various values of W and N_v such that $VN_v = 0.96W$.

D_{max}	g_{max}	N_{max} (Videoconf.)		N_{max} (Star Trek)		N_{max} (Comm.)	
(ms)	(per min.)	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5	$\hat{s}_{target}$=4.0	$\hat{s}_{target}$=4.5
25	0.1	326 (88%)	193 (85%)	222 (80%)	123 (77%)	116 (70%)	64 (67%)
	1	332 (90%)	195 (86%)	228 (82%)	126 (79%)	122 (73%)	67 (70%)
50	0.1	330 (89%)	195 (86%)	233 (84%)	129 (81%)	124 (74%)	68 (71%)
	1	333 (90%)	197 (87%)	240 (86%)	133 (84%)	129 (77%)	72 (76%)
100	0.1	332 (90%)	199 (88%)	236 (85%)	133 (84%)	126 (76%)	69 (72%)
	1	334 (90%)	201 (88%)	242 (87%)	136 (86%)	132 (79%)	73 (77%)
250	0.1	334 (90%)	202 (89%)	240 (86%)	135 (85%)	133 (80%)	70 (74%)
	1	334 (90%)	204 (90%)	245 (88%)	140 (88%)	140 (84%)	74 (78%)
500	0.1	334 (90%)	204 (90%)	242 (87%)	139 (88%)	142 (85%)	72 (76%)
	1	334 (90%)	204 (90%)	247 (89%)	142 (89%)	148 (89%)	76 (80%)
CBR Equiv. N_{max}		301	188	167	137	75	50

Table 18: Comparison of N_{max} for CQ-VBR and CBR on an ATM multiplexer, W=100 Mb/s

Videoconferencing						Commercials						
No. of Vconf. streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	No. of Comm. streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	Network Util.
194	4.0×10^{-4}	0.20	12.0	246	1793	0	—	—	—	—	—	86Mb/s
119	3.9×10^{-4}	0.29	12.1	269	1812	25	4.5×10^{-4}	0.58	10.3	235	1321	80Mb/s
26	2.4×10^{-4}	0.23	11.0	259	1773	57	5.2×10^{-4}	0.67	9.8	221	1218	72Mb/s
0	—	—	—	—	—	67	5.5×10^{-4}	0.90	11.3	260	1644	70Mb/s

Table 19: Mixing of different video content (Commercials and Videoconferencing), H.261, CQ-VBR, $\hat{s}_{target}$=4.5.

MPEG						H.261						
No. of MPEG streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	No. of H.261 streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	Network Util.
48	7.5×10^{-4}	1.3	7.8	215	1052	0	—	—	—	—	—	70Mb/s
30	6.4×10^{-4}	1.2	7.6	221	1047	25	5.3×10^{-4}	0.97	10.9	271	1689	70Mb/s
10	6.6×10^{-4}	1.2	7.9	219	1084	52	5.6×10^{-4}	0.96	11.1	268	1702	70Mb/s
0	—	—	—	—	—	67	5.5×10^{-4}	0.90	11.3	260	1644	70Mb/s

Table 20: Mixing of MPEG and H.261 video streams, Commercials, CQ-VBR, $\hat{s}_{target}$=4.5.

CBR						CQ-VBR						
No. of CBR streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	No. of CQ-VBR streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	Network Util.
50	0	0	—	—	—	0	—	—	—	—	—	90Mb/s
30	3.2×10^{-4}	0.44	10.1	236	1474	26	3.9×10^{-4}	0.57	11.2	275	1513	81Mb/s
7	3.8×10^{-4}	0.38	10.2	229	1421	58	4.2×10^{-4}	0.46	10.6	238	1452	74Mb/s
0	—	—	—	—	—	67	5.5×10^{-4}	0.90	11.3	260	1644	70Mb/s

Table 21: Mixing of CBR and CQ-VBR streams, H.261, Commercials, $\hat{s}_{min}$=4.3.

D_{max}=25 ms						D_{max}=500 ms						
No. of Streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	No. of Streams	L	g	$\overline{d}$	$\overline{A_{max}}$	$\overline{L_{tot}}$	Network Util.
67	1.5×10^{-3}	0.92	21.3	321	4644	0	—	—	—	—	—	70Mb/s
30	1.5×10^{-3}	0.91	21.3	320	4640	37	0	0	—	—	—	70Mb/s
0	—	—	—	—	—	75	4.0×10^{-3}	0.58	37.2	313	9176	79Mb/s

Table 22: Mixing of streams with different D_{max} values for H.261, CQ-VBR, Commercials, $\hat{s}_{target}$=4.5.

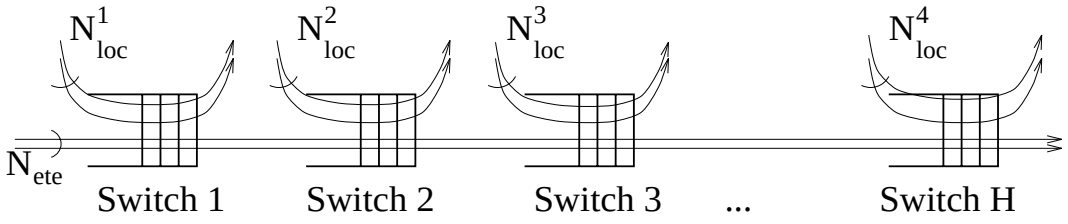


Figure 48: A worst-case scenario for ATM Multihop traffic.

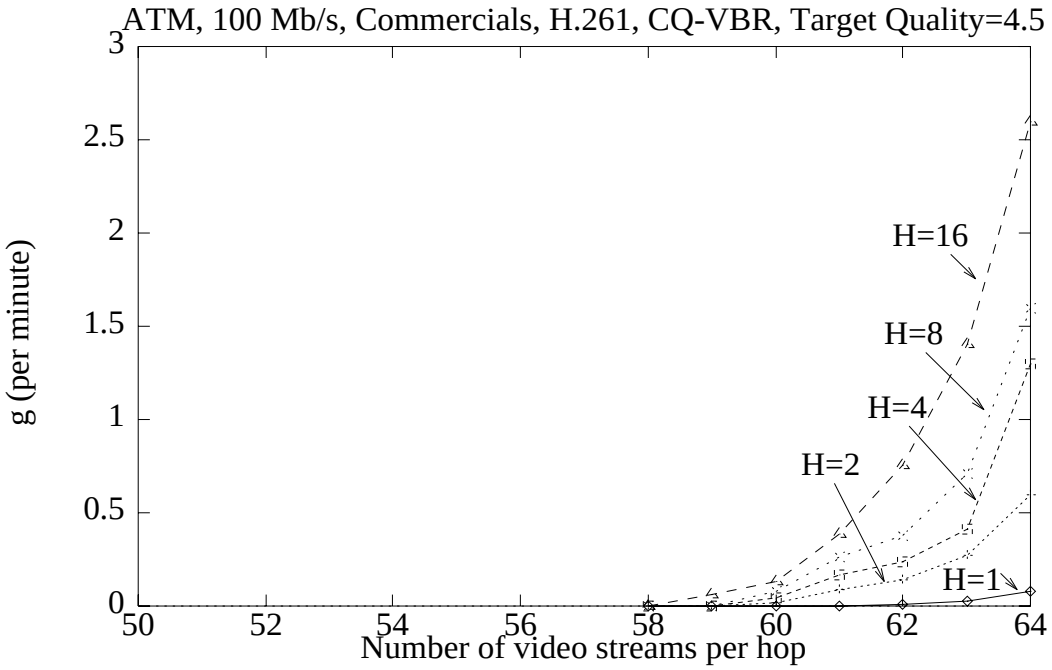


Figure 49: Glitch rate versus the number of video streams per hop for H={1,2,4,8,16}, W=100 Mb/s, *commercials*, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, N_{ete}=1.

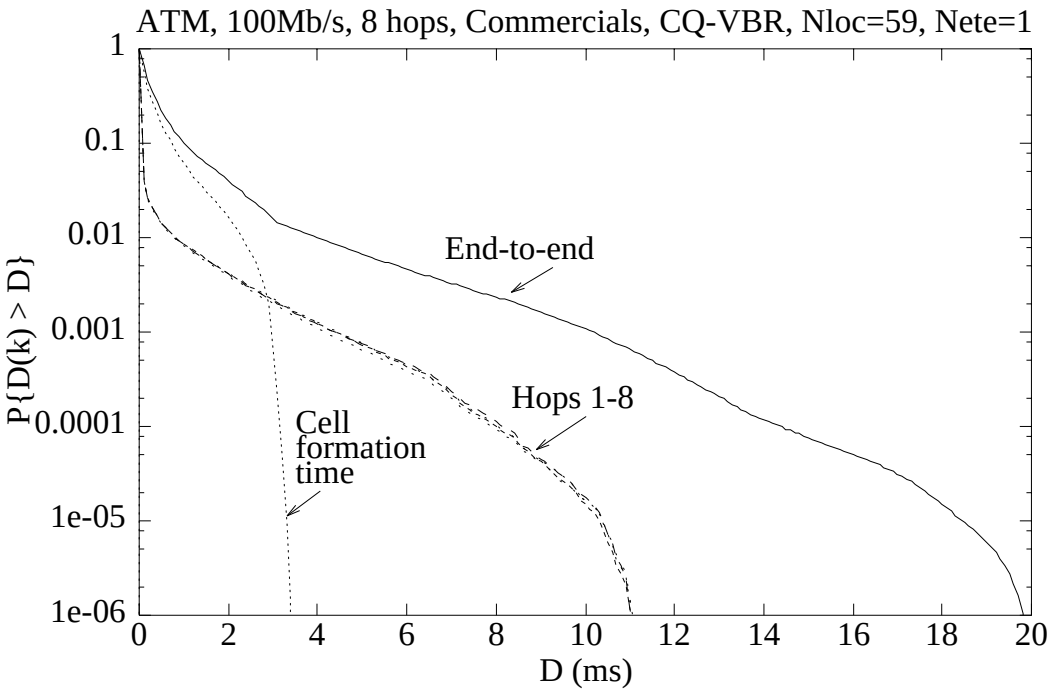


Figure 50: Complement of the cumulative distribution function of cell delay in individual hops and end-to-end for H=8, W=100 Mb/s, *commercials*, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, N_{loc}=59, N_{ete}=1.

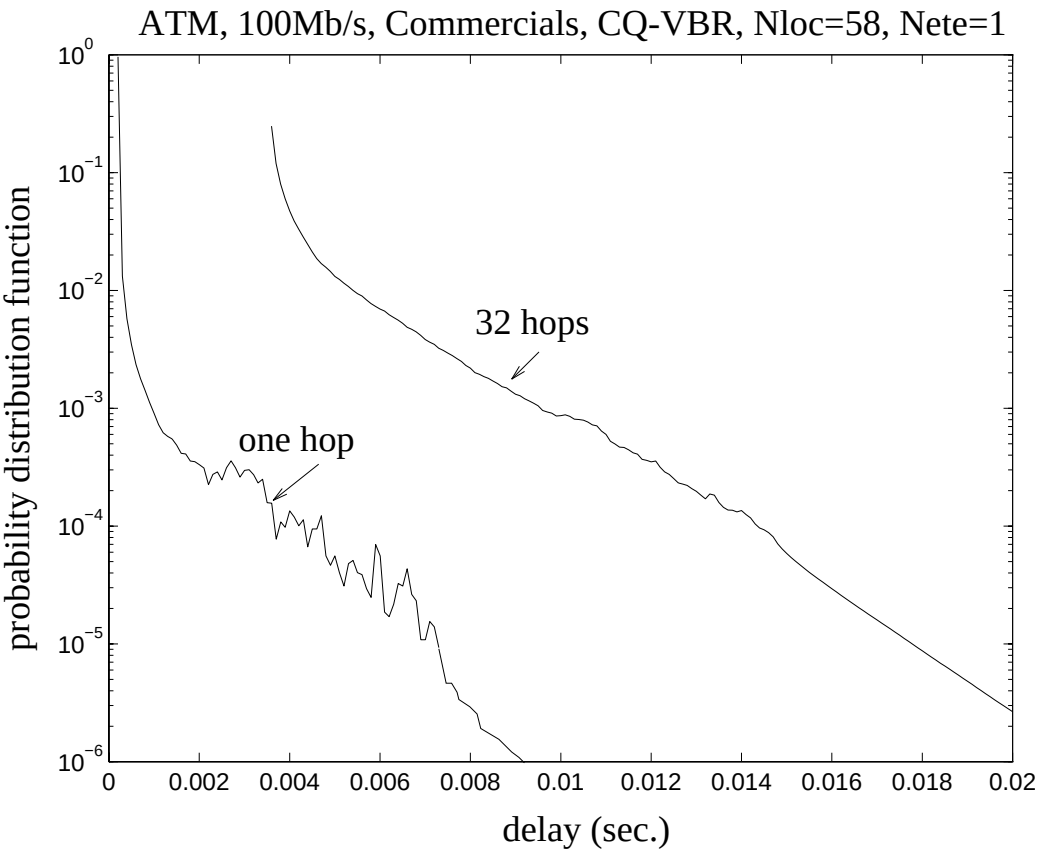


Figure 51: Probability distribution function of end-to-end delay for H=32, W=100 Mb/s, *commercials*, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, N_{loc}=58, N_{ete}=1. (Obtained by convolution.)

D_{max}	H=1	H=2	H=4	H=8	H=16
25 ms	64	61	60	60	59
100 ms	69	68	68	68	68

Table 23: Maximum number of streams supportable per hop (N_{max}) for W=100 Mb/s, g_{max}=0.1 per minute, *commercials*, CQ-VBR, $\hat{s}_{target}$=4.5, N_{ete}=1.

D_{max}	H=1	H=2	H=4	H=8
25 ms	64	62	61	61
100 ms	69	69	68	68

Table 24: Maximum number of streams supportable per hop (N_{max}) for W=100 Mb/s, g_{max}=0.1 per minute, *commercials*, CQ-VBR, $\hat{s}_{target}$=4.5, N_{ete}=50.

D_{max}	H=1	H=2	H=4	H=8	H=16
25 ms	192	191	190	190	189

Table 25: Maximum number of streams supportable per hop (N_{max}) for W=100 Mb/s, g_{max}=0.1 per minute, *videoconferencing*, CQ-VBR, $\hat{s}_{target}$=4.5, N_{ete}=1.

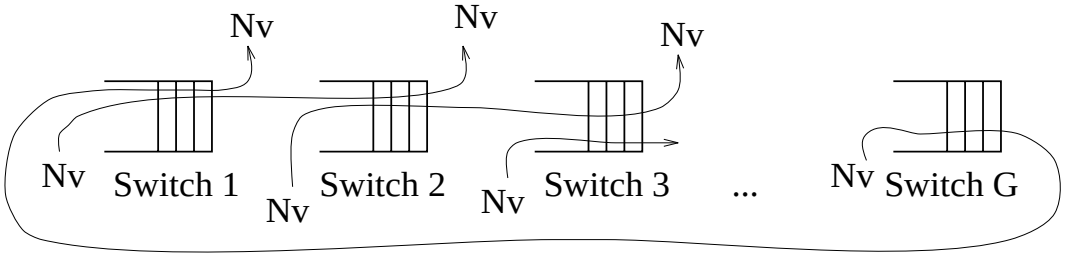


Figure 52: ATM multihop network and traffic scenario 2

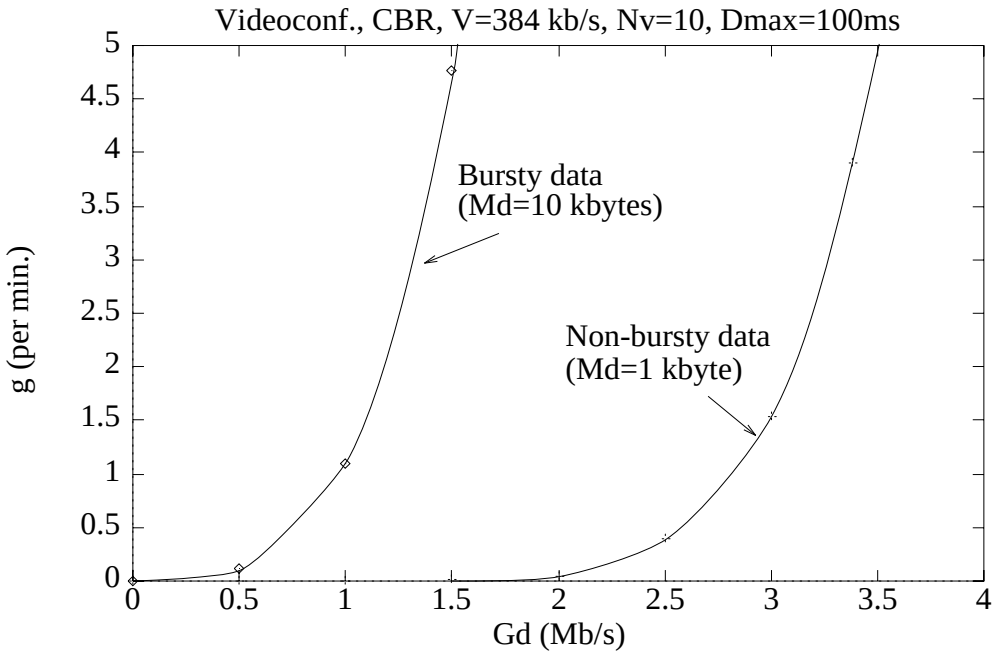
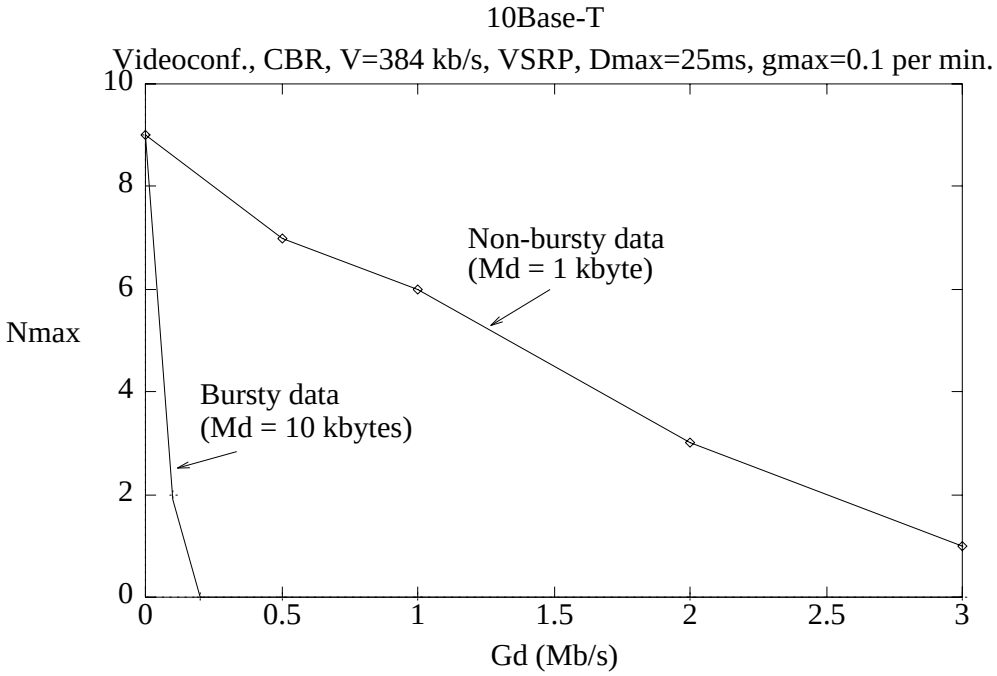
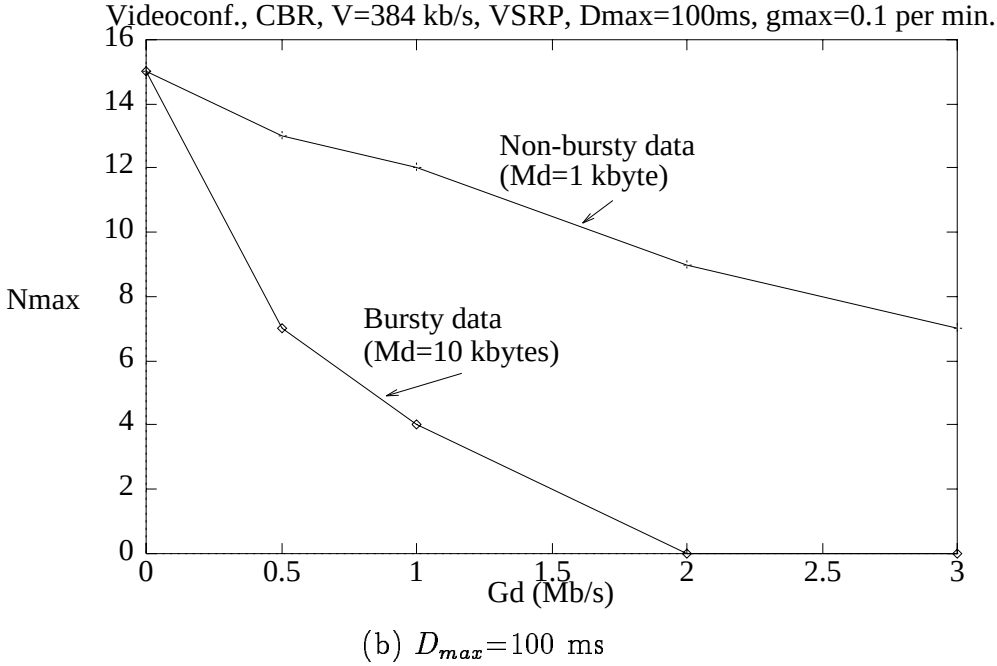


Figure 53: g vs. G_d for VSRP, D_{max}=100 ms, N_v=10, W=10 Mb/s.

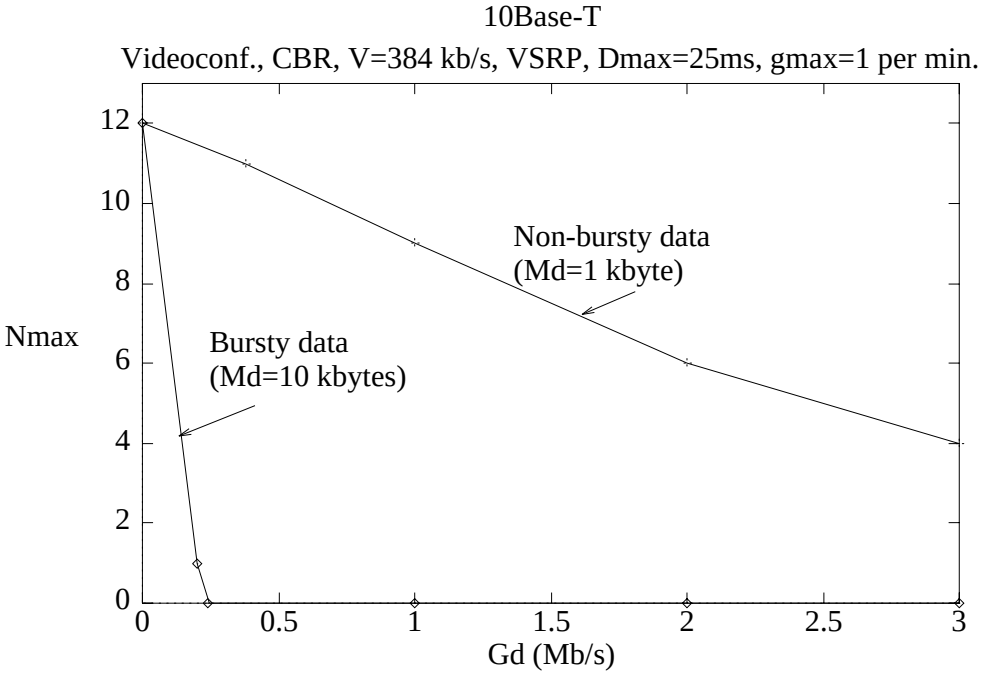


(a) D_{max}=25 ms

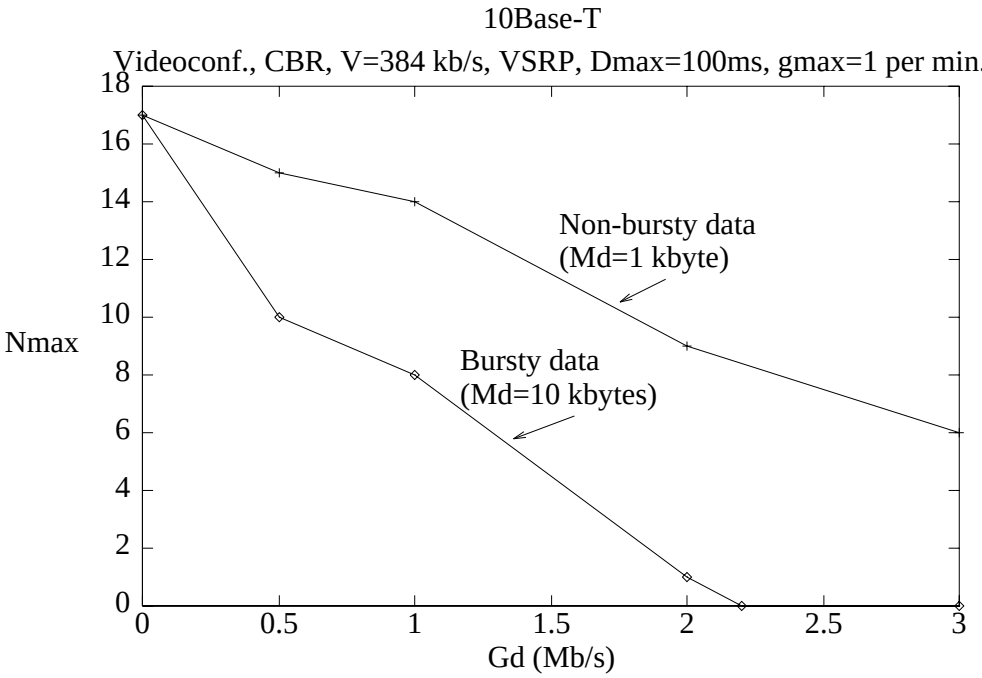


(b) D_{max}=100 ms

Figure 54: N_{max} vs. G_d for VSRP, g_{max}=0.1 per min., W=10Mb/s

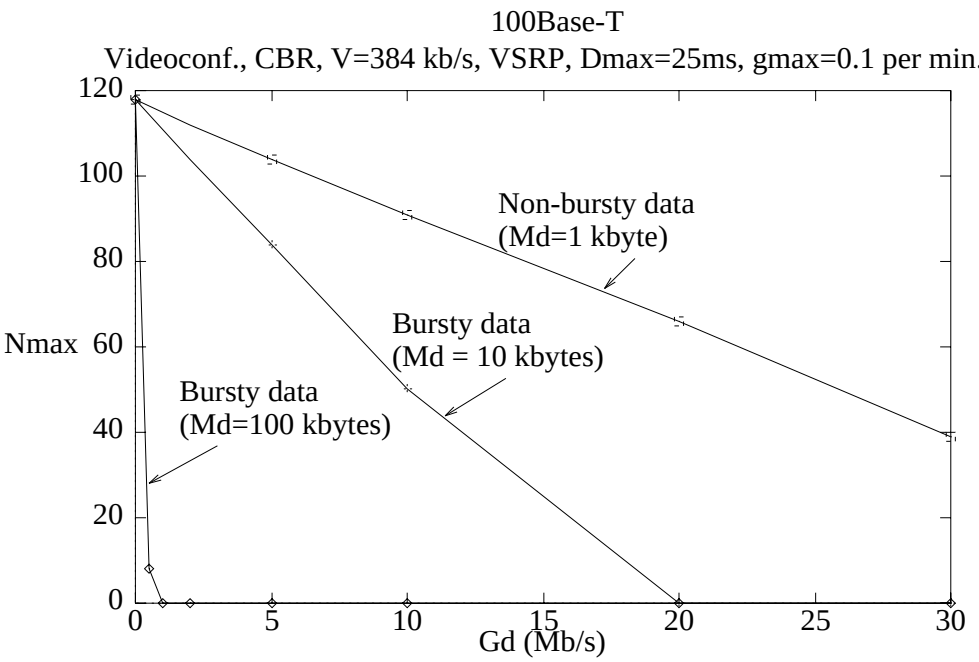


(a) D_{max}=25 ms

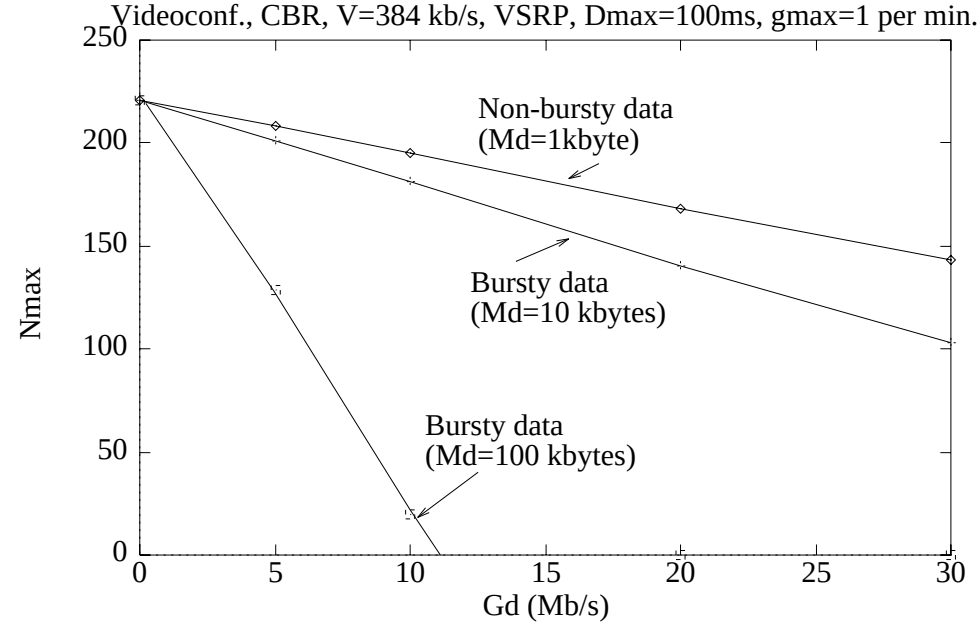


(b) D_{max}=100 ms

Figure 55: N_{max} vs. G_d for VSRP, g_{max}=1 per min, W=10Mb/s



(a) D_{max}=25 ms, g_{max}=0.1 per min.



(b) D_{max}=100 ms, g_{max}=1 per min.

Figure 56: N_{max} vs. G_d for VSRP, W=100 Mb/s

	M_d=1 kbyte	M_d=10 kbytes
Average Data Delay (ms)	3	9
Standard Deviation of Data Delay (ms)	2	8
Data Packet Loss Rate	0	0

Table 26: Data delay and packet loss statistics when there ar e N_{max} stations present on the network for *Videoconferencing*, CBR, VSRP, D_{max}=100 ms, g_{max}=1 per min,W=10 Mb/s.

	M_d=1 kbyte	M_d=10 kbytes	M_d=100 kbytes
Average Data Delay (ms)	0.7	2.8	6.0
Standard Deviation of Data Delay (ms)	0.6	2.4	4.2
Data Packet Loss Rate	6.4×10^{-4}	4.6×10^{-4}	5.2×10^{-5}

Table 27: Data delay and packet loss statistics when there a re N_{max} stations present on the network for *Videoconferencing*, CBR, VSRP, D_{max}=100 ms, g_{max}=1 per min, W=100 Mb/s.

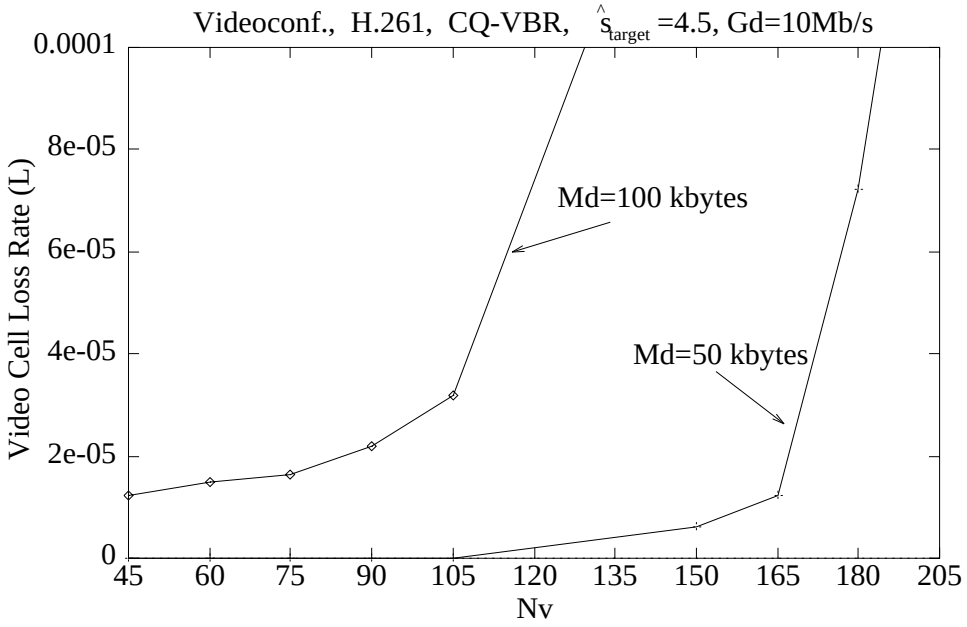


Figure 57: Video cell loss rate versus N_v for ATM, Videoconferencing, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, G_d=10 Mb/s.

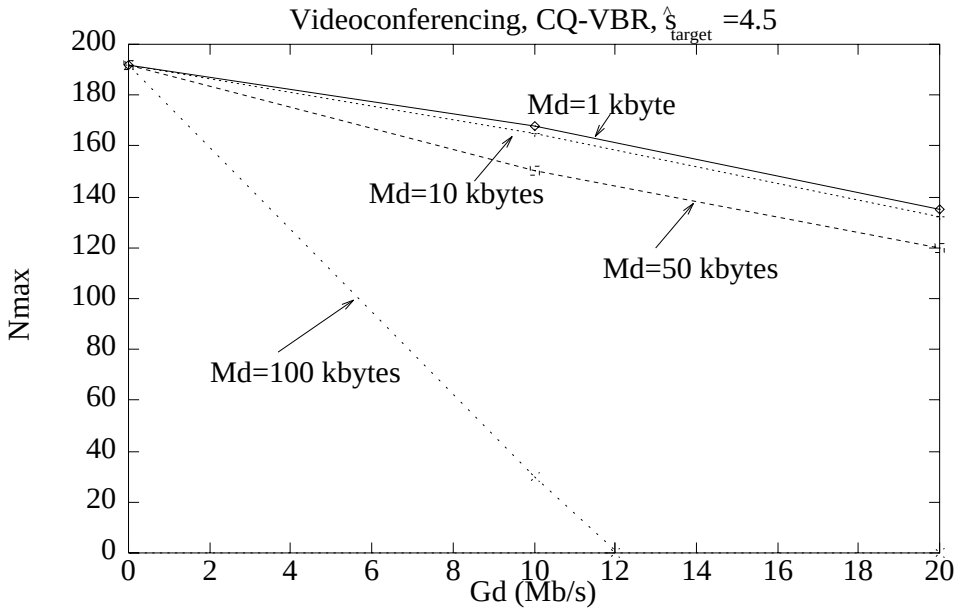


Figure 58: N_{max} versus G_d for ATM, Videoconferencing, H.261, CQ-VBR, $\hat{s}_{target}$=4.5, D_{max}=25 ms, g_{max}=1 per min.