End-to-end Delay Analysis of Videoconferencing over Packet Switched Networks

Mario Baldi and Yoram Ofek

February 18, 2000

Abstract

Videoconferencing is an important **global** application - it enables people around the globe to interact when they are far from one another. In order for the participants in a videoconference call to interact naturally, the end-to-end delay should be below human perception; even though an objective and unique figure cannot be set, 100 ms is widely recognized as the desired one way delay requirement for interaction. Since the global propagation delay can be about 100 ms, the actual end-to-end delay budget available to the system designer (excluding propagation delay) can be **no more than 10 ms**. We identify the components of the end-to-end delay in various configurations with the objective of understanding how it can be kept below the desired 10 ms bound.

We analyze these components going step-by-step through six system configurations obtained by combining three generic network architectures with two video encoding schemes. We study the transmission of raw video and variable bit rate (VBR) MPEG video encoding over (*i*) circuit switching, (*ii*) synchronous packet switching, and (*iii*) asynchronous packet switching. In addition, we show that constant bit rate (CBR) MPEG encoding delivers unacceptable delay – on the order of the group of pictures (GOP) time interval – while maximizing quality for static scenes.

This study aims at showing that having a **global common time reference**, together with **time-driven priority** (TDP) and VBR MPEG video encoding, provides adequate end-to-end delay, which is (*i*) below 10 ms, (*ii*) independent of the network instant load, and (*iii*) independent of the connection rate. The resulting end-to-end delay (excluding propagation delay) can be smaller than the video frame period, which is better than what can be obtained with circuit switching.

Keywords: Time-driven Priority, MPEG, Videoconference, Quality of Service End-to-end Delay, Performance Guarantees.

Early version was published in IEEE INFOCOM 1998.

1 Introduction

Interactive real-time applications over packet switched networks are challenging. Even though in some cases a poor service can be tolerated–e.g., if it is charged at a low price–the focus of this work is on a high quality service. One of the key features of such a service is to enable natural interaction that requires the end-to-end delay to be below human perception. Various studies concluded that for natural hearing this delay should be around 100 ms [1]. Even though an objective and unique figure does not exist, a 100 ms delay ensures full satisfaction to all users. While a lower end-to-end delay cannot be appreciated, delays above 100 ms will be noticed by some users and will lead them to look for a better service; eventually, all users will look for a better service. Thus, service providers that can guarantee a 100ms end-to-end delay will have a distinct market advantage.

The problem that we study in this work is videoconferencing in which voice and video should be synchronized (a.k.a. lip-sync), thus, the end-to-end delay of the video should be below 100 ms as well. Since the global propagation delay can be about 100 ms the actual end-to-end delay budget available to the system designer (excluding propagation delay) can be **no more than 10 ms**.

The video stream requires high capacity and since network resources are limited, the video pictures should be compressed. Compression can be costly in terms of end-to-end delay. In this work we assumed MPEG encoding [2], since it is one of the most popular compression techniques. Other encoding techniques such as Motion JPEG [3, 4] and H.261 [5] are possible, but they are not in the scope of this paper and are left for further research. Here we show that MPEG constant bit rate (CBR) video encoding is not advisable for high quality videoconferencing applications. As discussed in Section 3.2, efficient compression with elimination of temporal redundancy introduces unacceptable delay. Lower delay is obtained, at the expense of a higher bit rate video stream, by not eliminating temporal redundancy: the MPEG encoder works as a Motion JPEG one.

Intuitively, for small end-to-end delay, a picture (video frame) is captured, compressed, and sent once there are enough data units to send them in packets over the network. This approach can result in a short end-to-end delay. However, it raises some key questions: (*i*) *when* the compressed video data units will be ready to be sent (in general, the data units generation during compression is difficult to predict), and (*ii*) *how many* data units will be generated after compression of each picture. Both pieces of information are needed in order to reserve communication resources inside the network. In other words, the difficulty arises since the time data units are produced and the amount of data units produced may change from picture to picture. In this paper, we discuss the two problems in details and suggest some solutions which provide deterministic and predictable global quality of service (QoS) guarantees.

The heart of these solutions is the way in which packet queuing and forwarding is managed by nodes. Some solutions are based on nodes having a common time reference obtained from the GPS (global positioning system) [6] and using it to control packet forwarding; this allows queuing to be reduced in the network nodes. Other approaches are based on asynchronous packet forwarding and require special techniques, such as Packet Generalized Processor Sharing (PGPS) [7, 8], to bound the packet delay by using scheduling algorithms that approximate the fluid flow service model. Such schemes guarantee a bound on the queuing delay which is inversely proportional to the connection (or session) rate, proportional to the number of nodes traversed by that connection, and proportional to the packet size. The common time reference enables the solution with the lowest end-to-end delay bound and jitter. However, other solutions can be used when the end-to-end delay requirement is not stringent, i.e., when users can tolerate poor interactivity due to a large delay bound, or variable quality due to non deterministically controlled delay.

Designing the network for stringent delay requirements can be beneficial to various system aspects, other than user perceived quality. One of the main advantages of *small delay jitter are small buffers* inside the network and at the receiver side. Networks with high speed links require buffers with short access time, which can be expensive, and therefore, small delay jitter will save money.

In the future, when *virtual reality* applications will become real (rather than virtual) the system constraints, such as delay and loss, will be even more rigid. Therefore, in the coming years, the competition among various network vendors will be more than just the capability to provide the service: it will become primarily a competition to provide better quality of network services. Some of the techniques discussed in this manuscript will indeed provide better quality for a lower price (because of less stringent buffering requirements, for example).

2 The Model

In this section we identify the components of the end-to-end delay of a videoconferencing system for a number of relevant system configurations. In Section 3, we analyze each delay component for the various configurations. The model focuses on video rather than audio for two reasons:

1. The audio sampling rate is typically 8 kHz, which means that a voice sample is produced every 125μs, while the video sampling rate is much lower, typically 5-30 pictures per second. In general, with higher sampling rate it is possible to obtain shorter delays. Since audio samples are encoded on few bits (at most 8 bits per sample), the main limitation for audio delay is the time needed to obtain enough samples to build up a packet with reasonably low overhead. This delay can be kept short by using packets with small size header and consequently small size payload, e.g., ATM cell payload is only 48 bytes.

2. The audio bandwidth requirements are small relative to (good quality) video, and therefore, audio compression is not necessary in high speed networks, while video will have to be compressed in the foreseeable future.

2.1 High-level Delay Components

Figure 1 shows the model of a videoconferencing system. The end-to-end delay of the system is the time elapsed from when a video image is captured by the video camera at the sender side until when it is displayed on the monitor at the receiver (upper arrow in Figure 1). In order for the participants in the videoconference call to be able to interact naturally we have the following objective:

Objective 1 *The end-to-end delay (including propagation delay) should be below 100 ms.*

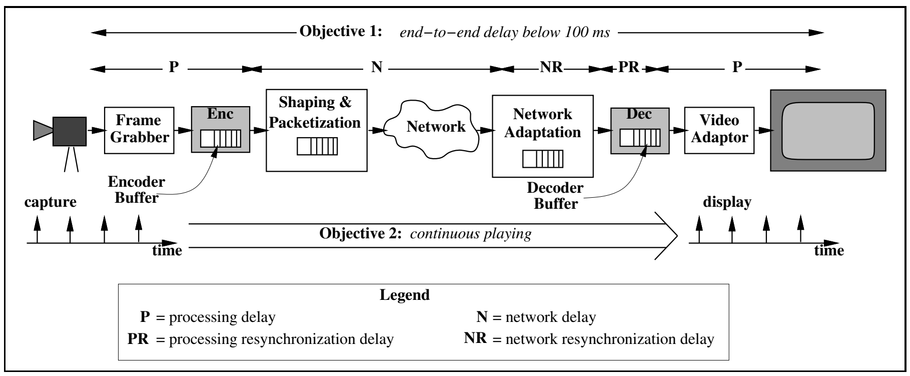

Figure 1: Model of a Videoconferencing System.

For delivering high visual quality, video frames should be displayed on the receiver's monitor at the same fixed pace they have been captured. This leads to the second objective (see Figure 1):

Objective 2 Continuous play. *The receiver displays pictures (plays audio samples) continuously and at the same rate they have been captured by the sender.*

Objective 2 means that the end-to-end delay between capture and display is constant.

The end-to-end delay is modeled with four high-level components, whose values depend on the system configuration. The segmented arrow in Figure 1 shows which function in the system introduces each delay component.

1. A *processing* delay (**P** in Figure 1) is introduced on both the sender and receiver sides. It may encompass, for example, the time spent in compressing and decompressing of pictures.

2. The *network* delay (**N** in Figure 1) is the time needed to move data units from a source to the other videoconference participant(s). The network delay includes also the protocol processing in both sender and receiver(s).

The above two delay components can vary during a videoconference call. In order to meet Objective 2 (i.e., constant end-to-end delay) these variations should be "smoothed out" before pictures are displayed at the receiver. We identify two *resynchronization* delay components, which are typically realized by some sort of *replay buffer*. All pictures when exiting this buffer have experienced the same delay from the time they were captured.

3. The *processing resynchronization* delay (**PR** in Figure 1) cancels the delay variations in generating the compressed video data units.

4. The *network resynchronization* delay (**NR** in Figure 1) cancels the variations of the delay experienced in the network (e.g., the delay jitter due to queuing in network nodes).

Thus, Objectives 1 and 2 can be summarized using the four delay components in the following way: after resynchronization the end-to-end delay, including propagation delay - Pr, is:

$$\mathbf{P} + \mathbf{N} + \mathbf{PR} + \mathbf{NR} + Pr = \mathbf{CONSTANT} \leq \mathbf{100\ ms}.$$

2.2 The Analysis Methodology

In this work we analyze the end-to-end delay by going through a number of configurations, each adding one or more components to the end-to-end delay. The system configurations differ for the network architecture and the video encoding technique exploited. We consider three network architectures and three video coding techniques; since we conclude that one of the latter is not suitable for videoconferencing, we end up with six system configurations which are studied in Section 3. The network architectures are described in Section 2.3.

The following three video encoding schemes are considered:

1. *Raw* video: the three color components of each picture are digitally sampled and the resulting data units are transmitted over the network with no delay. In order to reduce the network delay, the bits encoding each frame should be sent as soon as they exit the capture card. Since the capture card provides all the bits within few milliseconds, the traffic generated by the source may be concentrated in this short time interval, i.e., it is bursty.

2. Constant Bit Rate (CBR) MPEG: pictures are encoded according to the MPEG standard. Compression is obtained by eliminating spatial (within each picture) and temporal (between subsequent pictures) redundancy. The amount of bits needed to encode each picture (a.k.a. picture size) is not known in advance and is highly variable. Since the picture rate is constant, bits are produced at a variable rate. A buffer is used to smooth the production bit rate: bits exit this buffer (and enter the network) at a constant rate. The encoder is controlled according to the fill level of the buffer in order to prevent it from either overflowing or underflowing. The delay introduced by the buffers cannot be kept below 10 ms for CBR MPEG, as it is discussed in details in Section 3.2.

3. Variable Bit Rate (VBR) MPEG: only a small buffer is exploited at the output of the encoder for assembling data units which may exit the encoder shortly after they are produced. The rate of the resulting compressed stream is highly variable.

2.3 The Network Architectures

Three network architectures are considered. One is circuit switching which is a fully synchronous network: routing and flow control are done using time. The second is asynchronous packet switching with no notion of global common time reference. And the third one is a combination of the previous two, called time-driven priority (TDP); it has a global common time reference that is used only for flow control and not for routing.

2.3.1 Circuit Switching

A fixed amount of link capacity is assigned to each videoconference call by means of time division multiplexing. At the time a data unit (e.g., a byte) is transmitted from its source, it is possible to predict deterministically when it will exit any switch along its route. The time resolution of this advanced knowledge is *much shorter than the data unit transmission time.* As a result, network nodes introduce a small (few microseconds) delay.

2.3.2 Asynchronous Packet Switching

In packet switched networks data are gathered in packets which are sent to an ingress switch or router. Switches forward packets toward the destination while statistically multiplexing packets from different sources which are forwarded over the same link. When a packet happens to have to be forwarded on a busy link, it is delayed until the link is available. This delay is called (network) queuing delay.

Queuing delay has high variability since the time spent in an output buffer depends on the packets already in this buffer and in other buffers of the same output port. Thus, the distribution of the queuing delay experienced by packets throughout the videoconference call is determined by the resource allocation policy, the scheduling algorithms used for buffer management, and the overall network traffic characteristics.

Queuing delay often counts for a large portion of the network delay (see Section 3.1.3 for details on network delay components) and its high variability gives a major contribution to the network delay variation, a.k.a. *network jitter*. The maximum jitter is defined, in the context of this work, as the difference between maximum and minimum delay. The network resynchronization delay should be between zero (for packets having experienced maximum network delay) and the maximum jitter (for packets having experienced minimum network delay).

Actually, the replay buffer introduces an *excess (network) resynchronization* delay up to the maximum network jitter (see Section 3.1.3 for a detailed explanation). As a result, the network delay possibly contributes with its maximum value plus its maximum jitter.

2.3.3 Time-driven Priority

Time-driven priority (TDP) [10, 11] gives higher priority to real-time traffic in a *periodic fashion* in order to provide the following properties for real-time traffic: (i) bounded delay - independent of the best-effort data traffic, (*ii*) constant bound on the jitter, which is independent of the network size, and (*iii*) either *deterministic* no-loss or *probabilistic* control of the loss due to congestion inside the network. For example, for real-time service it is possible to ensure *deterministically no (loss due to) congestion inside the network.* Moreover, this can be achieved under a full link utilization and without adversely affecting the quality of service. TDP is a multiplexing scheme aimed at sharing link capacity while guaranteeing users against uncontrolled delays (or even losses) due to contention in accessing network's links.

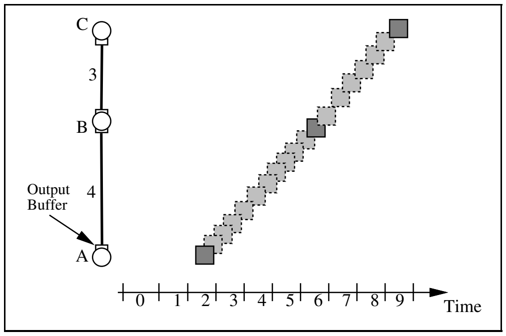

Figure 2: TDP Packet Forwarding.

The time is divided into *time frames* (TFs) of fixed duration T_f (a typical choice is $T_f = 125\,\mu s$). Given the link capacity C, in each TF a fixed amount of bits $T_f \cdot C$ can be sent on a link. Assuming small propagation delay, a real-time packet is forwarded one hop every TF. Since packets are not buffered for uncontrolled time, in the more general case with arbitrary propagation delay, each packet takes a fixed number of TFs to move from the output buffer of an intermediate node to the output buffer of the following one on the path to the destination, as shown in Figure 2. This is also called *RISC-like forwarding* because packets move through the network in the same step-lock fashion as instructions advance in the pipeline of a RISC processor.

3 End-to-end Delay Analysis

This section presents the end-to-end delay of the six system configurations following the delay model presented in Figure 1. For clarity, two of the delay elements shown in Figure 1 are not included in the analysis, since they are the same in all configurations. They are briefly described in the following.

Capture delay. The frame grabber introduces a constant delay on the order of few milliseconds.

Presentation delay. As a picture is ready to be displayed on the receiver side, it is inserted into the video frame buffer which is periodically scanned by the video adaptor to trace the image on the screen. This introduces a presentation delay which can be up to 17 ms, assuming the refreshing frequency to be 60 Hz (i.e., 1/60=16.667 ms). The presentation delay can be eliminated by synchronizing the decoder, the video controller inside the receiver, and the capture card, as shown in Figure 1. This requires synchronization between network and decoder, receiver network interface and sender network interface, network and encoder, encoder and capture card. This end-to-end synchronization, otherwise very hard to implement, can be easily obtained by using a common time reference, e.g., from the GPS [6].

3.1 Raw Video

The delay analysis of raw video is interesting since it concerns a reduced set of delay components, which are the network (**N**) and the network resynchronization (**NR**) delays. There are only two delay components since there is no compression, and therefore, the processing (**P**) and processing resynchronization (**PR**) delays are null[1].

3.1.1 Circuit Switching

In circuit switched networks it is assumed that the video transmission is continuously using the bandwidth, B^{CS}, allocated to this circuit. Figure 3 shows, for each frame, the resulting timing (and the rate) of the production of bits by the capture card and of their transmission over the network. This introduces a *network shaping* delay

$$S_{Raw}^{CS} = \frac{F_r}{B^{CS}},$$

where F_r is the picture size in bits. The end-to-end delay is constituted by a single component (the network delay **N** depicted in Figure 1) given by

$$\Delta_{Raw}^{CS} = S_{Raw}^{CS} + Pr + Sw, \tag{1}$$

where Pr is the propagation delay, Sw is the circuit switching delay (typically, a few microseconds). The minimum circuit bandwidth required for the transmission of raw video is,

$$B^{CS} \geq \frac{F_r}{T}.$$

The end-to-end delay can be reduced by allocating larger bandwidth to the circuit. In this case, the circuit is busy only for a time S_{Raw}^{CS} in each video frame period T. As a result, the remaining time $T - S_{Raw}^{CS}$ is wasted and no other connection can exploit the reserved resource left unused.

If minimum bandwidth is to be allocated for the videoconference call, the end-to-end delay is larger than one video frame period. Therefore, the lower the video frame rate, the larger the end-to-end delay. For example, the minimum bandwidth required to send raw QCIF pictures at 15 frames per second, is 4.5 Mb/s and the resulting shaping delay is 67 ms. However, if more bandwidth is allocated to decrease the network shaping delay to 30 ms, more than 50% of the allocated bandwidth is wasted because the circuit is idle for half of the video frame period.

[1] Analog-to-digital and digital-to-analog conversions of pictures requires few milliseconds which is a short time in comparison to the other delay components.

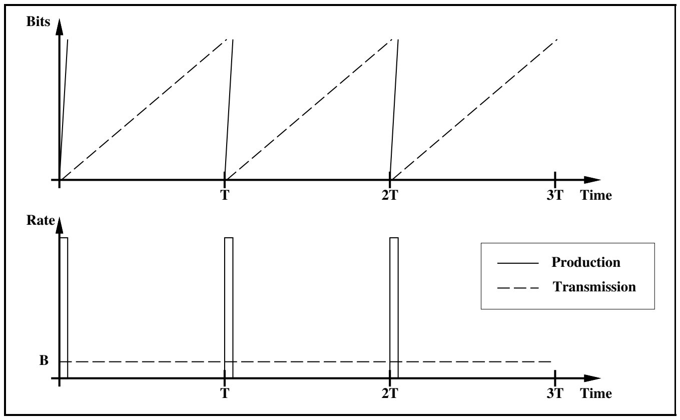

Figure 3: Raw Video Encoding and Transmission over a Circuit Switched Connection.

3.1.2 Time-driven Priority

Raw video can be sent over a packet switched network with TDP by inserting each picture into one or more packets which are transmitted during a TF[2]. During the TFs between the transmission of two subsequent pictures of the same session, capacity can be reserved to other real-time sessions, as shown in Figure 4. All the unused capacity (both reserved and un-reserved) can be exploited for the transmission of best-effort traffic.

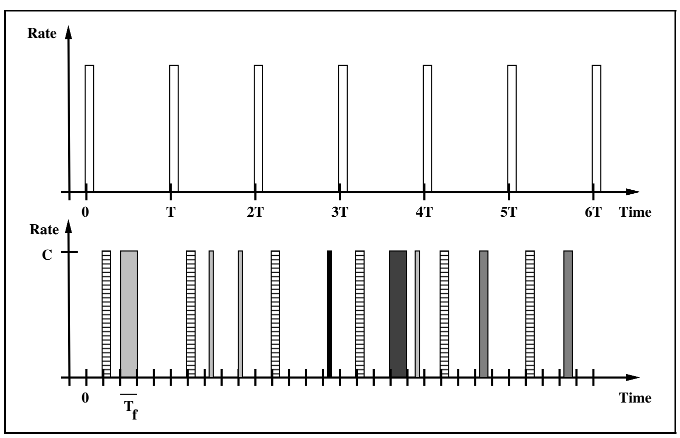

Figure 4: Generation of Raw Pictures and Transmission over a Network with Time-driven Priority (TDP).

The network delay **N** is the only component of the end-to-end delay. It can be expressed as $L \cdot T_f$ (where L is a function of the number of nodes and the processing delay inside each node) and the propagation delay - Pr. The end-to-end delay is:

$$\Delta_{Raw}^{TDP} = L \cdot T_f + Pr. \tag{2}$$

In TDP the presentation delay is zero since the frame grabber and the video display adaptor are synchronized.

[2]If the link capacity is not large enough to allow a picture to be transmitted in a single TF, it is sent over more successive time frames. This introduces a network shaping delay given by the number of TFs needed to transmit each picture.

Resource reservation is based on the definition of a *time cycle* which encompasses a predefined number of TFs: all the nodes share the same knowledge of the ordinal position of the current TF inside the time cycle[3]. Bandwidth is allocated to a sender/receiver pair, by properly reserving (a fraction of) the link capacity during a number of TFs per time cycle on each link on the path from sender to receiver[4]. In order for intermediate nodes to perform the RISC-like forwarding, the TFs on a link must be chosen according to the TFs reserved on the upstream link, and the time needed for a packet to be transmitted from the output buffer of the upstream node to the output buffer of the considered node, see more details in [11].

3.1.3 Asynchronous Packet Switching

A picture, split over one or more packets, is sent through the network from the source to a packet switching node introducing a *transmission* delay F_r/C, where F_r is the picture size and C is the link bandwidth or capacity. For example, the transmission of a QCIF image over a T3 link (45 Mb/s) introduces a delay of 6.7 ms. In the network, packets experience a fixed propagation delay, Pr, and a variable queuing delay; all the delays mentioned so far are part of the network delay component **N**.

Due to the real-time requirements of the video stream, a replay buffer is needed at the receiver to compensate for the variation in the queuing delay. The compensation is obtained by delaying the samples that have experienced a queuing delay shorter than the maximum Q_M; the delay introduced is part of the network resynchronization delay **NR**. As a result of the compensation, the sum of the queuing delay and its compensation experienced by each sample is the maximum queuing delay Q_M.

Actually, the compensation of the queuing delay introduces also the *excess resynchronization* delay $E_r \in [0, \Delta Q]$ constant over the duration of the whole conference; ΔQ is the maximum variation of the queuing delay, i.e., the difference between the maximum and the minimum queuing delay. The excess resynchronization delay is introduced because when a packet is received the actual delay it experienced in the network is not know, as explained in detail in [9].

The compensation is implemented by delaying the first packet of a flow by ΔQ and then retrieving from the replay buffer the subsequent packets at a constant rate. Figure 5 shows the resynchronization of a packet stream the first packet of which has experienced minimum queuing delay Q_m: the upper diagram shows the arrival time of pictures to the replay buffer, while the lower shows the exit time. If the network interfaces of sender and receiver are not synchronized, the latter is not able to determine the queuing delay experienced by a packet. In particular, not knowing the delay experienced by the first packet received for the videoconference call, the receiver buffers it for a time that allows resynchronization in the worst case, i.e., it is assumed that the packet has experienced the minimum queuing delay Q_m and is buffered for a time

[3]This can be easily implemented with GPS [6].

[4]The TFs during which capacity is allocated to a videoconference call are said to be reserved to the call.

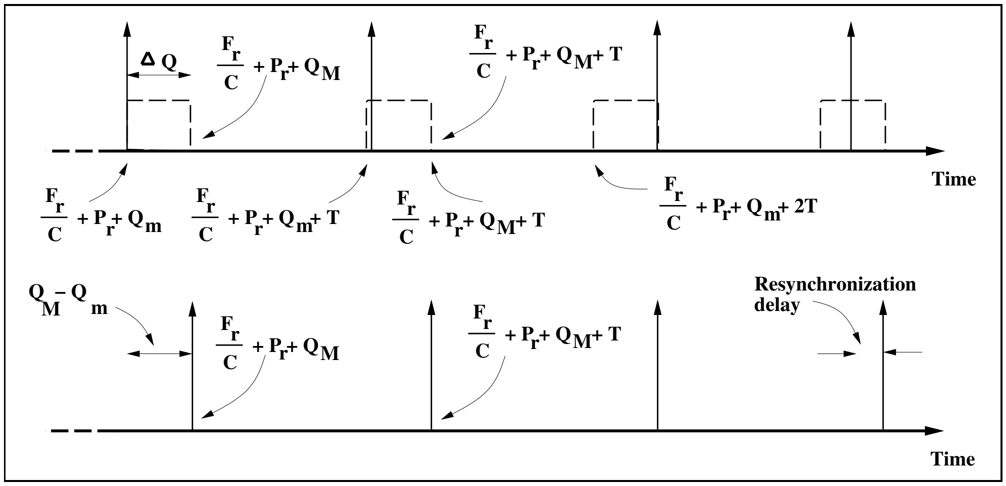

Figure 5: Network Resynchronization Delay to Compensate Queuing Delay Variation.

$$\Delta Q = Q_M - Q_m. \tag{3}$$

The following packets are resynchronized accordingly, as shown in Figure 5, because they are retrieved from the replay buffer at the constant pace at which pictures are displayed.

The upper diagram of Figure 6 shows a resynchronization example when the first packet experiences an actual queuing delay Q_M. The middle diagram shows the timing of packets exiting in case the receiver does not know the actual queuing delay experienced by the first packet, delays it by ΔQ according to (3). As a consequence, the overall delay experienced by the following packets due to queuing and resynchronization is between Q_M and $Q_M + \Delta Q$. If the receiver knows the actual delay experienced by the first packet (e.g., sender and receiver have a common time reference and the packet contains a time stamp indicating when it was sent), a packet that has already experienced maximum queuing delay in the network is not further delayed in the replay buffer and the exit times are those depicted in the lower diagram in Figure 6.

Thus, if sender and receiver do not share a common time reference, the end-to-end delay experienced by the user of the videoconferencing system is

$$\Delta_{Raw}^{Async} = \frac{F_r}{C} + Pr + Q_M + E_r.$$

The compensation delay introduced by the replay buffer and the related error constitute the network resynchronization delay component **NR** identified in Figure 1, while the other terms are part of the network delay component **N**. It can be worth pointing out that the contribution Q_M accounts partly for the **N** component (since it contains the queuing delay) and partly for the **NR** component (since it contains the compensation delay). The excess resynchronization delay E_r can be eliminated if the sender and the receiver have a common time reference.

The maximum queuing delay is usually much larger than both all other network delay components and the minimum queuing delay (that is given by the sum of the transmission delay over the traversed links).

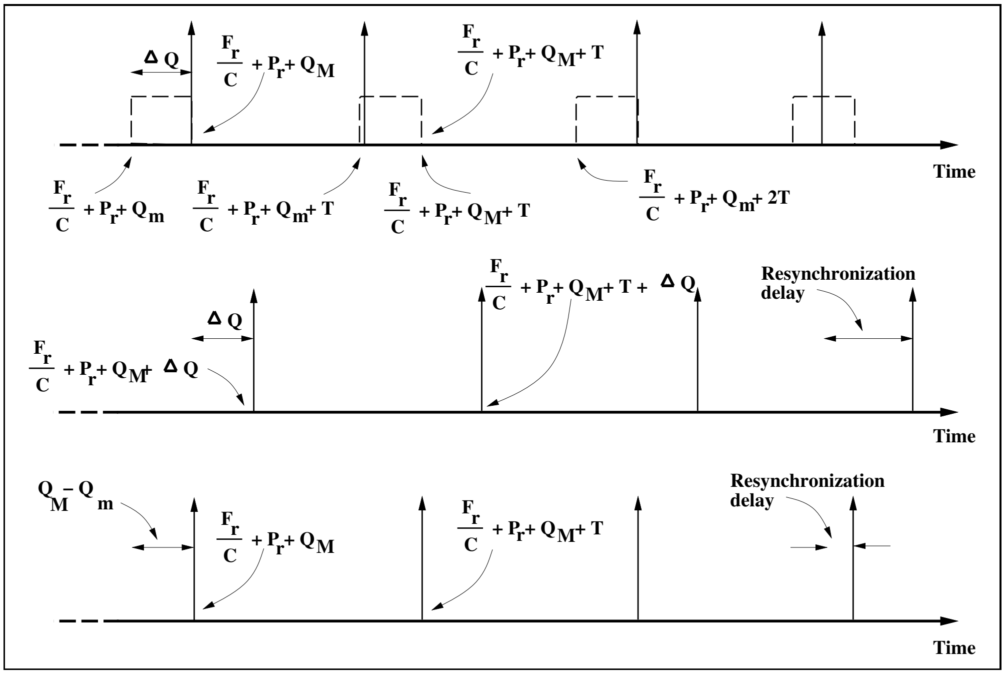

Figure 6: Network Resynchronization Delay to Compensate Queuing Delay Variation.

Consequently, $\Delta Q \simeq Q_M$ and, due to jitter compensation, *the maximum queuing delay can contribute twice* (namely, by itself and as excess resynchronization delay) to the end-to-end delay [9]. Schemes, like PGPS [8], are being proposed for preventing loss and bounding the queuing delay. Such schemes provide a bound that is inversely proportional to the bandwidth allocated to the session, and proportional to the packet size and the number of hops. This is stated, for example, by the general result on *PGPS Networks* in Section X (page 146) of [8]. In particular, Equations 37, 38 and 39 on page 148, which have the following general structure:
$PGPS - Delay - bound - connection_i \leq \frac{2(K-1)L_i}{\rho_i}$, where L_i is the packet size, K is the number of hops and ρ_i is the rate of connection i. [8] are based on several timing assumptions, such as, that the delay between nodes is zero or constant. Realizing such an assumption will require the synchronization of the local clocks of all the nodes, which is equivalent to the global common time reference used for TDP.

The contribution to the end-to-end delay due to the queuing delay and its compensation can be reduced by under-dimensioning the replay buffer, thus having the queuing delay and the excess resynchronization delay contributing with a value $\hat{Q}_M < Q_M$. As a consequence, all the (parts of) pictures experiencing a network delay larger than $F_r/C + Pr + \hat{Q}_M + E_r$ are discarded at the receiver at the expenses of the visual quality of the received video stream. $\hat{Q}_M$ is some percentile of the queuing delay chosen as to guarantee that the percentage of discarded packets does not affect visual quality. Some videoconferencing applications explicitly designed for operation over asynchronous packet switched networks adapt the resynchronization delay introduced by the replay buffer to the instantaneous distribution of the network delay experienced by packets [12]. This results in variable visual quality and user perceived delays, and does not comply with

Objective 2.

Losses in network nodes due to congestion and buffer overflow also decrease the perceived quality of a videoconference call. In order to reduce buffer overflows and control the distribution of queuing delay over the duration of videoconference calls, resources are reserved in the nodes along the connection path and access to the network is controlled. This limits the amount of guaranteed traffic routable over the same link. Since transmission of large amounts of data at link speed (*bursts*) makes queues grow suddenly, bursty sources require a large amount of resources to be allocated and significantly reduce the overall amount of real-time traffic the network can support. Source burstiness can be reduced through *traffic shaping* at the network boundaries. Traffic shaping mechanisms like, for example, the *leaky bucket* [13], guarantee an average bandwidth B to the source while keeping the burstiness below a predefined value[5]. This introduces a shaping delay

$$S_{Raw}^{Async} = \frac{F_r - A}{B}, \tag{4}$$

where A is the largest burst size, i.e., the maximum number of bits which can be sent at the full link speed. On one hand, the traffic shaping at the boundary of the network reduces the buffer requirements in the nodes, the queuing delay in the network and its variability (i.e., both the **N** and **NR** components of the end-to-end delay model proposed in Figure 1); on the other hand it introduces a variable shaping delay that is compensated on the receiver side (i.e., contributes to both the **N** and **NR** components of the end-to-end delay model proposed in Figure 1). In summary, the end-to-end delay can be expressed as

$$\Delta_{Raw}^{Async-TS} = S_{Raw}^{Async} + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r,$$

where P_s is the size of packets sent into the network.

3.2 Why VBR MPEG?

Compression, although reduces the transmission delay, may have large processing (**P**) and processing resynchronization (**PR**) delays. Moreover, if the naturally variable rate of a highly compressed stream is to be converted into a constant one, a significant contribution to the processing resynchronization delay must be further introduced, thus adversely affecting the end-to-end delay. The objective of the following discussion is to present and justify the rationale for recommending the use of VBR MPEG, rather than CBR MPEG, for videoconferencing applications. The motivation is that VBR encoding can provide high compression while keeping very short the contribution to the end-to-end delay due to the processing resynchronization (**PR**) component[6].

[5] If a leaky bucket is exploited, B is its token generation rate and A is the token pool size.

[6] A detailed discussion can be found in [9].

MPEG Overview. The Motion Picture Expert Group (MPEG) [2, 4, 14] video encoding standard was designed for digital storage of quality video for later re-playing. The encoder receives a sequence of digitalized pictures and encodes each of them in one of two different ways[7].

Intra-frame Coding eliminates spatial redundancy inside pictures. The picture (the luminance and the two chrominance components) is divided in 8x8 pixel blocks. On each block a Discrete Cosine Transform (DCT) is performed and an 8x8 matrix of coefficients is devised. Each coefficient is *quantized* by integer dividing it by an integer called *quantization step-size*. The result of the quantization is run-length encoded to gain further compression thanks to the many zero valued coefficients. Finally, the run-length encoded symbol sequence is Huffman encoded. The resulting encoded picture is called an *I-frame*.

Predictive Coding eliminates temporal redundancy between a picture and the previous one. The picture is divided into *Macro Blocks* (MBs), each composed of a 16x16 pixel matrix of luminance information and the two corresponding 8x8 pixel blocks of the two chrominance components. *Motion estimation* is performed for each MB, i.e., the previous picture is searched for a "similar" MB. If this MB is found, the pixel by pixel difference between the actual MB and the reference one is calculated and coded by performing DCT, quantization, run-length encoding, and Huffman encoding. If a "similar" MB is not found, each block of the MB is encoded like a block in an I-frame.

The obtained encoded picture is called a *P-frame* and it is typically from 2 to 4 times smaller than an I-frame. The more similar two subsequent pictures are, the higher the probability of finding in the reference picture a MB "similar" to the one being coded. Subsequent pictures are similar if the scene is slow moving, thus not changing much from a video frame period to the other. In summary, predictive coding delivers more compression on slow scenes.

A video sequence may be compressed by encoding one picture out of N as an I-frame, and the remaining $N-1$ pictures as P-frames; the sequence of N pictures is called a *Group Of Pictures* (GOP). The larger N, the smaller the amount of bits needed to encode each video sequence that contains a good deal of temporal redundancy. If the network introduces an error in an encoded I-frame, the error may propagate into the entire GOP. The next I-frame is the first picture not to be affected by such error. Thus, the larger N the more damage an error can have.

Controlling the Stream Rate. Due to the difference between I-frames and P-frames, the rate of the bit stream produced by the encoder has high variability. Figure 7 shows the amount of bits produced by the software MPEG encoder `dvdenc` [15] during the encoding of the "Cheerleaders" video sequence. It

[7] Actually, a third type of encoding, called bidirectional predictive coding, exists. Before a picture can be coded, a reference subsequent picture must be captured and coded. This introduces a delay of some frame periods that we consider not acceptable given the 100 ms end-to-end delay bound. Thus, this type of compression is not considered here.

shows a group of cheerleaders in a stadium, being therefore a scene with a lot of motion on a background with many details.

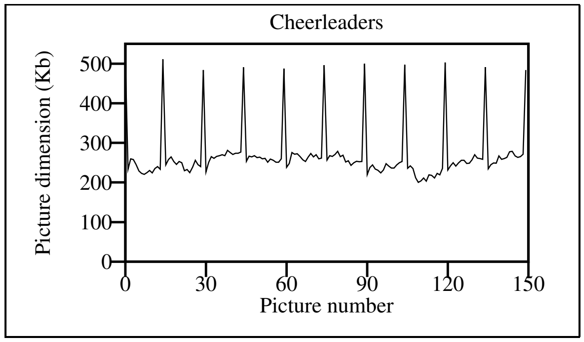

Figure 7: Natural Dimension of MPEG Encoded Pictures from the "Cheerleaders" Sequence.

A CBR MPEG stream is obtained by filling a buffer with the output of the basic encoding process and retrieving bits at a constant *target rate* B, as shown in Figure 8. This buffering process introduces a sensible variability of the processing delay component $\mathbf{P}$ of each picture which must be compensated in the decoder thus introducing a large processing resynchronization delay component $\mathbf{PR}$. A *rate control* function (in principle) monitors the fullness of the encoder buffer and adjusts the quantization step-size to change the natural bit rate in order to prevent it from underflowing or overflowing. The rate control function tunes the bit production of the encoder to grant the stream compliance according to a model of a system target decoder buffer (see Figure 8) whose dimension is included in the MPEG stream [2]. The bit production can be modified acting on the step-size of a quantizer used at the last stage of the encoding process[8].

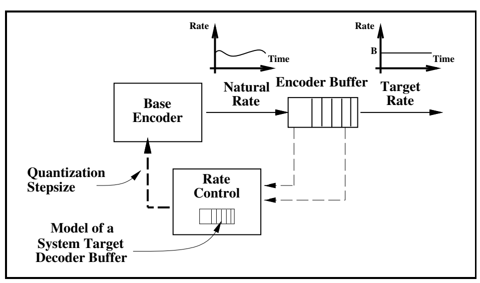

Figure 8: High Level Model of a CBR MPEG Encoder.

Delay and Picture Dimension. The CBR encoder contributes to the processing delay with F/B, being F the size of an encoded picture. Since picture size is not constant, this contribution is variable and must be compensated at the receiver. The resulting overall contribution (after resynchronization) to the end-to-end delay is called *coding shaping delay* S_c:

$$S_c \geq \frac{\max_{seq} F}{B},$$

[8]See [14] for further information on the MPEG encoding process.

where $\max_{seq} F$ is the maximum picture size over the whole sequence. In order for the video stream to be continuous (actually having a constant rate), $S_c \geq T$; otherwise there would be a time interval between two subsequent frames during which no bits exit the encoder. As a consequence, *CBR MPEG encoding always introduces a delay larger than the video frame period.* With reference to the model depicted in Figure 1, S_c contributes to the processing (**P**) and processing resynchronization (**PR**) delay components.

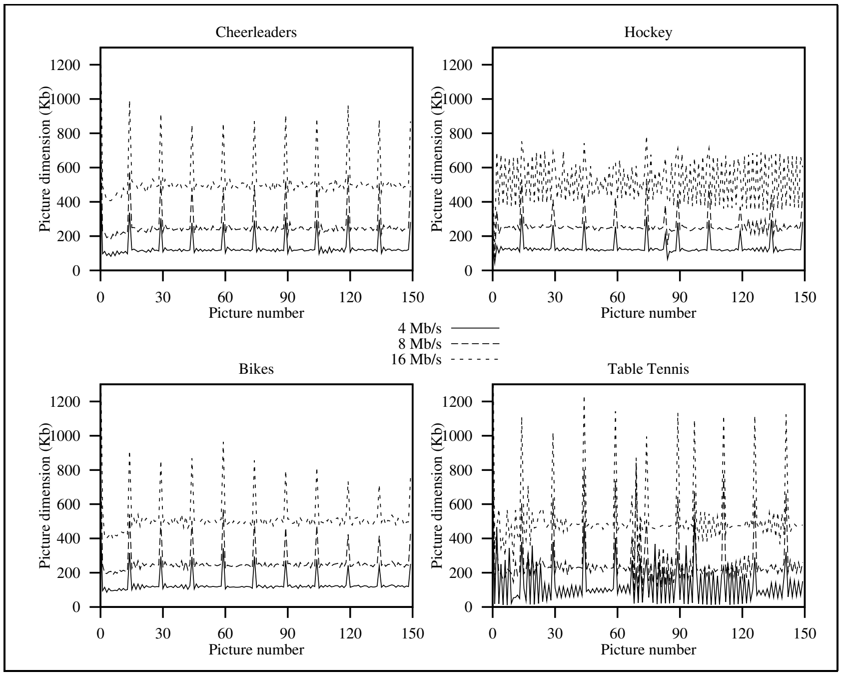

Figure 9: Dimension of Pictures in CBR MPEG Encoding with $N = 15$.

Figure 9 shows the picture size obtained when encoding four video sequences with the `dvdenc` [15] CBR MPEG encoder. Each sequence is encoded at three different target rates. The maximum picture size in each sequence determines the lower bound on the coding shaping delay. In order to keep constant the rate of the video stream (i.e., avoid the encoder buffer to underflow), the encoder must either limit the compression and keep P-frames from becoming too small, or increase the size of I-frames as the P-frames become smaller.

P-frame dimension is determined by the chosen encoding parameters and by the motion in the scene: the slower the scene, the larger the amount of temporal redundancy, the smaller the size of P-frames if such redundancy is eliminated. Videoconferencing scenes are usually quite static: P-frames are expected to be small and I-frames consequently large. In order to obtain a confirmation from experimental data, we encoded two completely static scenes built by replicating 120 times the same picture of the "Cheerleaders" and "Hockey" sequences, respectively. Figure 10 shows the resulting picture size for the same target rates used in the previous experiment, deploying the encoding parameters of `dvdenc`. As expected, using these encoding parameters, the maximum picture size, i.e., the lower bound on the coding shaping delay, has significantly increased with respect to Figure 9, especially at low bit rates.

The maximum picture size throughout a sequence is not known in advance. Nevertheless, when dealing

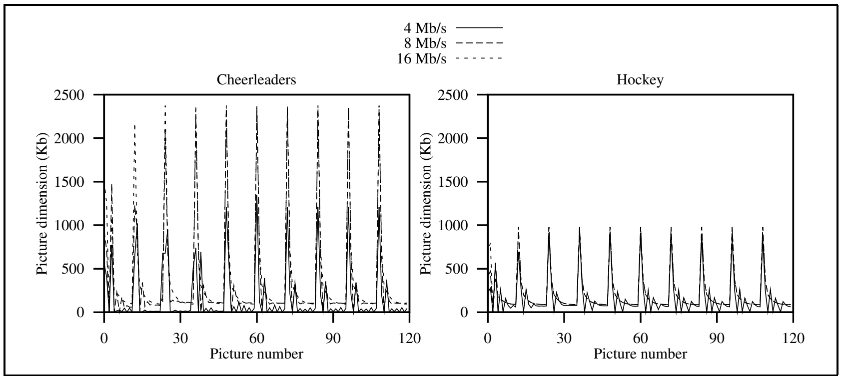

Figure 10: Dimension of Pictures in CBR MPEG Encoding of a Static Scene

with real-time video (like in a videoconferencing scenario) the encoding delay bound must be known before starting the videoconference call in order to meet Objective 2 (continuous playing) by introducing the proper resynchronization delay.

The following of this section is devoted to identifying the maximum picture size given the characteristics of the CBR MPEG encoder. We show that the corresponding coding shaping delay when applying CBR MPEG to videoconferencing is unacceptable for meeting Objective 1 (end-to-end delay less than 100 ms).

A Delay on the Order of the GOP Duration. Since the rate control function aims at avoiding the encoder buffer to overflow, picture size is upper bounded by the buffer dimension. It follows that the coding shaping delay can be reduced by exploiting a small encoder buffer. Hence, the issue becomes how to reasonably dimension the encoder buffer. Whenever a picture smaller than $B \cdot T$ is produced, the buffer must contain a backlog large enough to guarantee the continuity of the stream. Thus, the smaller the buffer, the less picture size can vary.

In principle, in order to deliver maximum quality the buffer should be large enough to allow P-frames to be encoded with no bits when the image is completely static. This is particularly important when dealing with videoconferencing because the camera can be pointed, for example, over a blackboard thus capturing a completely static scene. The fewer bits used to encode P-frames, the more can be dedicated to I-frame (thus creating the backlog in the buffer) and the higher the quality of the resulting image. Quality of static images is particularly critical since human eye is more sensitive to errors on static images, than on moving scenes.

Null size P-frames can be produced if an I-frame can be as large as the whole amount of bits sent during a GOP. I.e., in order to deliver maximum quality of static images, the encoder buffer size must be at least the GOP size; this results in a *coding shaping delay on the order of the GOP duration.* Such a coding shaping delay is not acceptable when aiming at Objective 1: for example, if operating at 30 frames per second with a GOP size of 15 pictures, the coding shaping delay is 500 ms.

An encoder designed to keep the introduced delay bounded may avoid eliminating temporal redundancy in P-frames, by encoding many macro blocks as in I-frames. In this way the size of I-frames is smaller and the coding shaping delay limited consequently. However, the quality of a static scene, like the one resulting from pointing the camera over a blackboard, is not maximized.

In summary, CBR MPEG encoding may not be the most advisable scheme for high quality videoconferencing; VBR MPEG is evaluated in the following and it seems to provide an appealing solution.

3.3 Transmission of VBR MPEG

While the CBR encoder introduces an unacceptable delay in the encoder, a VBR video stream may impact the network performance leading to either high delay, or high loss, or the need for an over-allocation of communication resources in some of the configurations. The rest of this section examines the resources required for the transmission of VBR MPEG video over the three network architectures, and the end-to-end delay of the resulting videoconferencing system.

3.3.1 Circuit Switching

The MPEG encoder produces bits with a timing similar to that shown in Figure 11 as it encodes I-frames and P-frames. The time required to encode a picture and the amount of bits produced are not constant. In principle, if the videoconference call is allocated a circuit with bandwidth larger than the maximum instantaneous rate of the encoder, bits are transmitted as soon as they are produced and bits get to the decoder at the same rate they had been produced after having experienced the constant propagation (Pr) and switching (Sw) delays. The time needed to encode a picture is not constant; the decoder introduces a

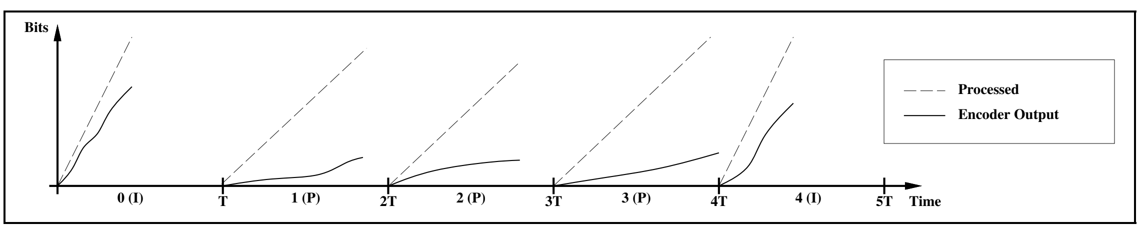

Figure 11: Bit Production of a Non-controlled VBR MPEG Encoder.

processing resynchronization delay to keep constant the time between decoded pictures. As a result, each picture experiences an overall coding-decoding-resynchronization delay CD_M which is the maximum time required to encode and decode a picture. The end-to-end delay is given by

$$\Delta_{VBR}^{CS} = CD_M + Sw + Pr,$$

where the first term is the overall coding-decoding-resynchronization delay and contributes to the processing delay[9] $\mathbf{P}$, while the other two terms contribute to the network delay component $\mathbf{N}$.

Since we are dealing with real-time video, a picture should be encoded (decoded) within the video frame period, i.e., any encoder will feature $CD_M \leq 2 \cdot T$. Thus, if a scene is captured at 30 frames per second

[9]Actually, CD_M contributes also to the processing resynchronization delay component $\mathbf{PR}$, but since the variability of the $\mathbf{P}$ component is relatively small, the contribution of $\mathbf{PR}$ can be neglected. See [9] for details.

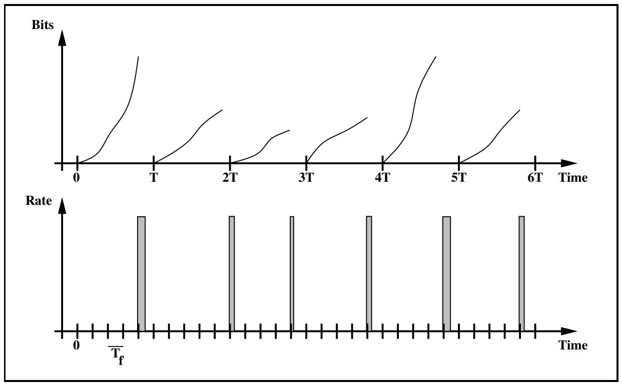

Figure 12: Transmission of VBR MPEG Video with Time-driven Priority.

and the end-to-end delay objective is 100 ms, the delay budget available for propagation is 40 ms which corresponds to a span of 8,000 Km (like a transoceanic call). A faster encoder features a lower CD_M and consequently enables a larger span.

The system configuration discussed in this section provides a lower bound on the end-to-end delay in a videoconferencing system exploiting MPEG compression. Nevertheless, it is not practical since the bandwidth of the circuit allocated to the videoconference call is only partially used and the unused fraction is wasted. Moreover, since the bandwidth of the circuit is equal to the peak rate of the encoder, encoding P-frames at a lower bit rate does not provide any advantage. Since motion estimation is the most time consuming function of the encoding process, CD_M is significantly reduced by exploiting only intra-frame coding. This corresponds to deploying a Motion JPEG encoder.

Reducing bandwidth occupation introduces a larger delay since transmission over a circuit with a bandwidth smaller than the peak rate of the encoder introduces a network shaping delay S^{CS}_{VBR}.

3.3.2 Time-driven Priority

As soon as all the bits for encoding a picture are produced by the encoder, they are inserted into a packet and sent at the full speed of the ingress link, as depicted in Figure 12. The end-to-end delay of the system is given by

$$\Delta^{TDP}_{VBR} = CD_M + L \cdot T_f + Pr, \tag{5}$$

where L is the number of TFs a packet takes to travel from sender to receiver. With reference to the model of the videoconferencing system depicted in Figure 1, the first term contributes to the processing delay component **P** (and in a negligible way to **PR**), while the second term contributes to the network delay component **N**.

Equation (5) is the actual end-to-end delay only if the nodes on the path from sender to receiver perform RISC-like forwarding of packets. To guarantee the fixed $L \cdot T_f$ network delay and loss free delivery, resources must be allocated in the network and video frames sent during reserved TFs. To reserve resources in packet switched networks with TDP, the amount of data to be sent and their timing must be known at reservation time so that the proper fraction of link capacity can be reserved during the proper TFs. The amount of bits reserved should be larger than (and as close as possible to) the dimension of the picture being sent. As was shown in Figure 7, picture dimension is not known in advance and resource reservation may not be accurate.

If during a TF a user sends more bits than the reserved amount, the network does not provide any guarantee on the delivery of the excess data units. If, on the other hand, the videoconferencing application uses only a fraction of the reserved capacity, the leftover bandwidth can be used by best-effort traffic and is not wasted (unlike circuit switching). Even though this is acceptable from the network point of view, the solution is not optimal for the user that is possibly paying for the allocated bandwidth and would like to use it all by himself.

The rest of the section concerns:

1. How to determine the amount of bits to be reserved for both types of pictures given the bandwidth to be allocated to a videoconference call.

2. The tuning of the encoding process in order to control picture dimension so that the videoconferencing system never uses more bandwidth than the allocated amount and exploits as much of it as possible.

3. How scheduling (i.e., the choice of the TFs to be reserved) impacts the end-to-end delay.

Even though some configurations can deliver unacceptable end-to-end delay, if the system is adequately equipped and operated, its performance is actually given by Equation (5).

1. **Choosing a Bound on Picture Dimension.** As explained in Section 3.2 on MPEG, the slower the motion in the scene being encoded, the larger the dimension of I-frames with respect to P-frames. Since videoconferences are expected to be slow moving scenes, we propose to reserve different amounts of bits for transmission of I-frames and P-frames. These amounts determine the bandwidth reserved to the videoconference call as

$$B^{TDP} = \frac{F^I + (N-1) \cdot F^P}{N \cdot T}, \tag{6}$$

   where F^I and F^P are the amount of bits reserved for I-frames and P-frames, respectively, N is the number of pictures per GOP, and $N \cdot T$ is the GOP duration.

   As discussed, the relative dimension of I-frames and P-frames yielded by a non-controlled MPEG encoder depends on the amount of motion in the scene. The *picture ratio*

$$\alpha = \frac{F^I}{F^P} \tag{7}$$

must then be chosen wisely depending to the amount of motion expected in the scene to be encoded and transmitted. This is, in general, a difficult task, but in the particular case of videoconferencing, scenes are likely to be slow and α consequently large.

Combining Equations (6) and (7), the amount of bits to be reserved to each frame can be expressed as a function of the bandwidth to be allocated as

$$\begin{cases} F^P = \frac{B^{TDP} \cdot N \cdot T}{N+\alpha-1} \\ F^I = \alpha \cdot F^P \end{cases} \tag{8}$$

2. **Controlling Picture Dimension.** The reserved bandwidth is used more efficiently if the encoding process is controlled and picture size is kept below (and as close as possible to) the amount of bits reserved. This amount is an upper bound on picture dimension and can be used to devise a target dimension for each type of video frame. There exist proposals to predict the amount of bits an encoding process is going to produce based on the raw scene [16] or on the portion of stream already produced [17]. However, to get service guarantees from the TDP network, the upper bound must never be exceeded and an adaptive approach based on feedback from the output of the encoder should be exploited. The encoder is extended with a *rate control function*, as shown in Figure 13. It tunes the parameters of the basic MPEG coding process so that the dimension of each picture is best fitted to the target size associated with its type. Among the parameters of the MPEG encoding process the quantization step size is the most suited to this purpose.

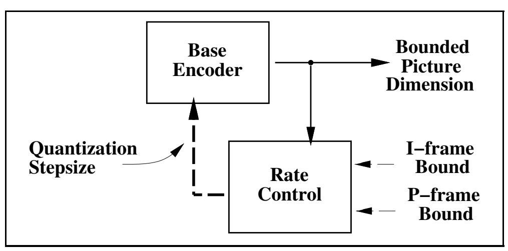

Figure 13: MPEG Encoder for Controlling Dimension of Frames.

Varying the granularity of quantization throughout the picture delivers non-uniform visual quality. Many approaches have been proposed in the literature for uniformly choosing the quantization parameter [18] possibly taking into account the characteristics of the human visual system [19]. Other authors propose iterative approaches [20, 21] to determine a suitable value of the quantization step-size to be used throughout a whole picture. This can lead to coding times not acceptable for real-time encoding as required in video-conferencing applications. [22] proposes to exploit a rate-quantization model to choose a quantization step-size for a whole picture. The model is tuned according to the characteristics of the encoded stream already produced. The rate control algorithm also proposes how to re-quantize the picture if the yielded dimension is not compliant with the target.

The above mentioned approaches have been proposed and analyzed in scenarios different from ours.

Thus, we have performed some experiments to prove that picture dimension can be controlled as the proposed network technology requires. The software encoder `dvdenc` has been augmented with a rate control function which calculates the quantization step-size on a MB by MB basis given a target picture dimension F_t and a tolerance on it (in terms of maximum and minimum acceptable dimensions). The quantization step-size is determined as a function of the amount of bits F produced so far and the number of bits expected according to the target F_t. The prototype rate control function has proven to be able to always keep picture size below the given bound F^I or F^P. As a consequence, bandwidth can be efficiently allocated by reserving fixed amount of bits for the transmission of pictures of the same type.

3. **Complex Scheduling.** Scheduling, i.e., the choice of the TFs to be reserved to a video-conference call, is simplified by considering the nature of the application generating traffic. Different amounts of bits should be reserved in the TFs intended for sending I-frames and those for P-frames. The time cycle must be set to an integer multiple M of the GOP period and $N \cdot M$ TFs must be reserved within the time cycle. The choice of the TFs to be reserved on each link on the path between sender and receiver is called *complex scheduling*[10]. The choice of the TFs impacts both network performance (in terms of maximum number of real-time connections concurrently supported) and the end-to-end delay of the video-conference call.

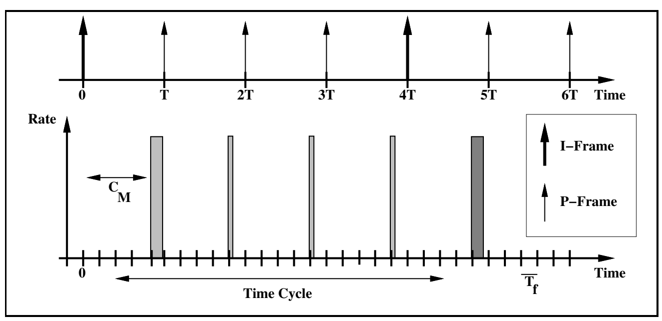

Figure 14: Time Driven Priority and Complex Scheduling.

Figure 14 shows a sample reservation with $M = 1$ and $N = 4$. The upper diagram depicts the frame grabbing time and the lower one shows the amount of bits reserved in the TFs: F^I in one TF and F^P in the following $N-1$ TFs. The capacity for sending the encoded video frames is reserved in the first TF beginning after C_M (the maximum coding delay) from the capture of each picture. This is the optimal schedule which leads to minimum delay as given by Equation (5). In order for the encoded pictures to be ready before the reserved TF, the capture card (as well as the encoder) must be synchronized with the network interface. Lack of synchronization would introduce a variable delay whose maximum value would be the GOP period (namely, $N \cdot T$). Also, if the optimal schedule is not feasible, a *scheduling shaping delay* S_{Sched} is introduced which contributes, together with $L \cdot T_f$, to the network delay component **N** identified in the model depicted in Figure 1.

[10]This scheduling is said complex because not the same capacity is allocated during all the reserved TFs.

Assuming synchronization between capture card and network interface, the general equation for the end-to-end delay is

$$\Delta_{VBR}^{TDP-CxSc} = CD_M + S_{Sched} + L \cdot T_f, \tag{9}$$

where $S_n^{Sched} \in [0, N \cdot T]$ is determined when performing the complex scheduling (i.e., when the videoconference call is placed); it can be small if a wise scheduling is performed.

Reducing Decoding Time. Decoding time can be reduced if the decoder does not wait for the whole picture to be received before starting processing it. This requires encoded pictures to be inserted in smaller packets that are sent as soon as they have been assembled. This does not affect any component of the end-to-end delay given by Equation (9) other than CD_M.

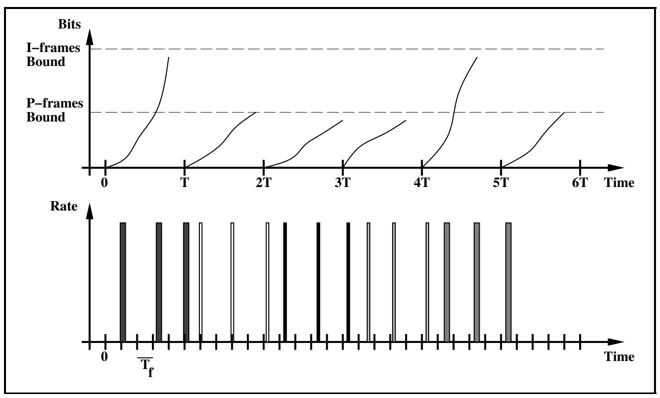

Figure 15: Time-driven Priority and Complex Scheduling.

It is thus better to allocate more than one TF per picture and send a packet in each of them (Figure 15), than to allocate a single TF during which a whole encoded picture is sent (Figure 14). Choosing this TFs encompasses two non-trivial issues: determining their position inside the time cycle and their number so that the end-to-end delay be minimized. In the following these problems are described and the trade-offs between different choices are outlined, but a solution is not proposed because it is out of the scope of this work.

When a TF reserved on the sender link begins, enough bits must have been produced by the encoder, otherwise the amount of transmitted data is smaller than the one reserved. This is called *encoder underflow* and it is critical not just because the allocated bandwidth is underutilized (it can be exploited by best-effort traffic). If the total amount of bits actually produced to encode the picture is very close to the global amount of bits reserved for the picture (over a number of TFs), the remaining reserved TFs will not have enough capacity to carry the other bits that should be sent in the present TF. On the other hand, if the TFs reserved for a picture are chosen later (with respect to the beginning of the video frame period) in order to minimize the risk of encoder underflow, the benefit of using more TFs to transmit a picture is reduced, i.e., the end-to-end delay is increased.

3.3.3 Asynchronous Packet Switching

As soon as the encoder produces enough bits to assemble a packet of dimension P_s, the packet is assumed to be sent into the network where it experiences a variable queuing delay. The receiver must exploit a replay buffer to compensate the queuing delay variation, thus introducing the network resynchronization delay component **NR**. The end-to-end delay is thus given by

$$\Delta_{VBR}^{Async} = CD_M + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r,$$

where CD_M is the maximum time required to encode and decode a picture (contributing to **P** and to a small **PR**), $\hat{Q}_M$ is some percentile of the maximum queuing delay, and $E_r \in [0, \Delta Q]$ is the excess resynchronization delay introduced by the replay buffer (contributing to **NR**). Using small packets and a decoder which starts decoding video frames as soon as data are received, can reduce CD_M. Anyway, The end-to-end delay is dominated by $\hat{Q}_M$ that, with reference to the videoconferencing system model depicted in Figure 1, contributes to the network delay component **N** and to the network resynchronization delay component **NR**.

Traffic Shaping at Network Boundaries. Resources are reserved in the network in order to bound the queuing delay (i.e., to reduce **N** and **NR**). As discussed in Section 3.1.3, resource reservation is more efficient if traffic shaping is performed at the boundaries of the network, even though it introduces a network shaping delay (i.e., it increases **N** and **NR**). Resources are reserved in the network based on the traffic description, given in terms of burstiness and average rate, corresponding to the shaped traffic. The network guarantees the quality of the service (i.e., the deterministic bound Q_M on the queuing delay or the statistical one $\hat{Q}_M$), only if the actual traffic is compliant with the description given at resource reservation time.

The delay globally experienced by a picture due to the traffic shaper depends on the natural bit generation rate of the encoder, the implementation of the traffic shaper, and the characteristics of the shaped traffic. The receiver has to compensate it by means of the network resynchronization delay introduced by the replay buffer. Thus, each packet experiences a network shaping delay S_{VBR}^{Async} partly in the traffic shaper and partly in the replay buffer. Equation (4) gives the delay introduced when dealing with raw video. Since devising an analogous equation for VBR MPEG encoded video is a harder task and it is not the goal of this work, we do not analyze S_{VBR}^{Async} in more detail. The end-to-end delay of the videoconferencing system is given by

$$\Delta_{VBR}^{Async-TS} = CD_M + S_{VBR}^{Async} + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r.$$

and contributes to the **P**, **N**, and **NR** components[11] of the end-to-end delay model presented in Figure 1.

[11] A relatively small contribution to the **PR** component is also present.

Adapting the Encoded Video Stream to the Network. Since the traffic pattern generated by a natural VBR MPEG encoder is not known in advance, it can be incompatible with the shaped traffic description. The non-compliant packets can be discarded by either the traffic shaper itself, or a traffic policing function inside the network [23]. For example, if a leaky bucket is exploited to shape the traffic, the token generation rate B and token pool size A determine the average rate and the burstiness of the shaped traffic. If the characteristics of the encoded video are not compatible with the values chosen for B and A the excess traffic must be either discarded or sent in the network as best-effort traffic. Even though a buffer is inserted before the leaky bucket to adapt the video stream to the traffic description, it can overflow if the two are too different.

The loss of packets is not acceptable in the transmission of MPEG encoded video, especially when the GOP is large. Even though techniques have been proposed to limit the effect of loss [24], it should be better for the videoconferencing system to avoid loss in order to deliver the highest possible quality.

The MPEG encoding process can be controlled to avoid that the traffic shaper discards or sends as best-effort traffic packets that cannot be adapted to the traffic description. A rate control function tunes the parameters of the basic MPEG encoder according to the traffic description used to drive the traffic shaper [25, 26]. Due to the unpredictable output of MPEG encoders, this approach does not guarantee against packets not compliant with the traffic description.

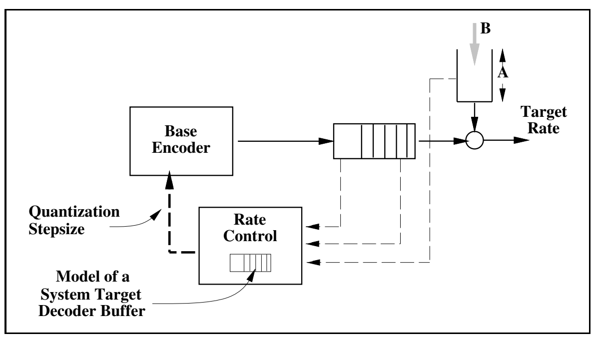

Figure 16: MPEG Encoder Controlled using Feedback from a Traffic Shaper.

Alternatively, a rate control function can tune the MPEG encoder parameters based on feedback information received from the traffic shaper (e.g., the fullness of the buffer preceding a leaky bucket) [27, 28], as shown in Figure 16. If this will significantly degrade the visual quality of pictures, the resource allocation can be renegotiated according to a rate-quantization model that is tuned as the encoding progresses.

4 Discussion

In this work we analyze the end-to-end delay of videoconferencing in six system configurations obtained by combining three network technologies with two encoding schemes. The results are summarized in Table 1. We study the transmission of raw video and variable bit rate MPEG video over (*i*) circuit switching, (*ii*) time-driven priority packet switching, and (*iii*) asynchronous packet switching. In addition, we show that in some encoding configurations constant bit rate (CBR) MPEG encoding delivers long delay, which is on the order of the group of pictures (GOP) duration. Thus, if the sampling rate is of 30 frames per second and a GOP of 15 pictures is used (i.e., each I-frame is followed by 14 P-frames), the resulting delay is on the order of 500 ms.

	Circuit Switching	Time-driven Priority	Asynchronous Packet Switching
Raw Video	$\Delta^{CS}_{Raw} = S^{CS}_{Raw} + Sw + Pr$ **N**	$\Delta^{TDP}_{Raw} = L \cdot T_f + Pr$ **N**	$\Delta^{Async}_{Raw} = \frac{F_r}{C} + Pr + \hat{Q}_M + E_r, E_r \in [0, \Delta Q]$ $\Delta^{Async-TS}_{Raw} = S^{Async}_{Raw} + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r$ **N + NR**
VBR MPEG	$\Delta^{CS}_{VBR} = CD_M + Sw + Pr$ **P + N**	$\Delta^{TDP-CxSc}_{VBR} = CD_M + S_{Sched} + L \cdot T_f + Pr$ $S_{Sched} \in [0, N \cdot T]$ **P + N**	$\Delta^{Async}_{VBR} = CD_M + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r$ $\Delta^{Async-TS}_{VBR} = CD_M + S^{Async}_{VBR} + \frac{P_s}{C} + Pr + \hat{Q}_M + E_r$ **P + N + NR**

Table 1: End-to-end Delay for the System Configurations Considered in this Work.

Given the long distance and high bit rate requirements of videoconferencing, video compression should be used. Since CBR encoding has unacceptable delay, variable bit rate (VBR) encoding should be used. In the case of VBR encoding, circuit switching is not practical since the network utilization is very low. *Thus, packet switching should be used.* However, relying on asynchronous packet switching with statistical multiplexing and first come first serve queuing discipline can result in high loss and delay jitter under high load conditions. Other queuing disciplines, such as weighted fair queuing, can only guarantee deterministic no loss to CBR traffic, which as mentioned would result in an unacceptably large delay bound.

This study shows that having a global common time reference can be used for implementing time-driven priority with complex periodicity scheduling for transporting variable bit rate MPEG encoding. This will provide adequate deterministic delay bounds which are *independent of the network load and the connection rate.* In such system configuration, the end-to-end delay (excluding propagation delay) can be smaller than the video frame period. Furthermore, time-driven priority with complex periodicity scheduling can *deterministically ensure no loss (due to congestion) of VBR traffic.* These unique results cannot be obtained with circuit switching or any other known alternative schemes.

Acknowledgments

We thank Prasoon Tiwari for providing us with the software MPEG encoder `dvdenc` and Peter Westerink for his kind and useful help in understanding, modifying, and operating the encoder.

References

[1] G. Karlsson, "Asynchronous transfer of video," *IEEE Communications Magazine*, pp. 118 – 126, Aug. 1996.

[2] ISO/IEC, *Information technology - Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s*, International Organization for Standardization, 1993.

[3] B. Fufht, "A survey of multimedia compression techniques and standards. Part I: JPEG standard," *Real Time Imaging*, , no. 1, pp. 49 – 67, 1995.

[4] R. Steinmetz and K. Nahrstedt, *Multimedia: computing, communications & applications*, Prentice Hall, Upper Saddle River, NJ 07458, 1995.

[5] ITU-T, *Recommendation H.261*, Video codec for audiovisual services at p x 64 kbit/s, Mar. 1993.

[6] P. H. Dana, "Global positioning system overview," http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html.

[7] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The sigle-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[8] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 137–150, Apr. 1994.

[9] M. Baldi and Y. Ofek, "End-to-end delay of videoconferencing over packet switched networks," Research Report RC 20669 (91480), IBM, Dec. 1996.

[10] C-S Li, Y. Ofek, and M. Yung, ""Time-driven priority" flow control for real-time heterogeneous internetworking," in *IEEE INFOCOM'96*, 1996.

[11] C-S. Li, Y. Ofek, A. Segall, and K. Sohraby, "Pseudo-isochronous cell switching in ATM networks," in *IEEE INFOCOM'94*, 1994, pp. 428–437.

[12] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 340 – 351, June 1996.

[13] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the 'leaky bucket' policing mechanism in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 355 – 342, Apr. 1991.

[14] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 47 – 58, Apr. 1991.

[15] E. Linzer, "A robust MPEG-2 rate control algorithm," Unpublished technical report, IBM – T. J. Watson Research Center, department 924A.

[16] R. M. Rodriguez-Dagnino, M. R. K. Khansari, and A. Leno-Garcia, "Prediction of bit rate sequances of encoded video signals," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 305 – 313, Apr. 1991.

[17] S. Jung and J. S. Meditch, "Adaptive prediction and smoothing of MPEG video in ATM networks," in *IEEE International Conference on Communications*, 1995, pp. 832 – 836.

[18] C. Horne and A. Puri, "Video coding with adaptive quantization and rate control," in *Visual Communications and Image Processing*, 1992, vol. 1818, pp. 798 – 806.

[19] A. Sultan and H. A. Latchman, "Adaptive quantization scheme for MPEG video coders based on HVS (Human Visual System," in *Visual Communications and Image Processing*, 1996, vol. 2668, pp. 181 – 188.

[20] L. Wang, "Rate control for MPEG video coding," in *Visual Communications and Image Processing*, 1995, vol. 2501, pp. 53 – 63.

[21] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "A gradient based rate control algorithm with applications to MPEG video," 1995, pp. 392 – 395.

[22] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 12 – 20, Feb. 1996.

[23] I. E. G. Richardson and M. J. Riley, "Usage parameter control cell loss effects MPEG video," in *IEEE International Conference on Communications*, June 1995, pp. 970 – 974.

[24] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Cell loss concealment method for MPEG video in ATM networks," in *GLOBECOM '95*, 1995, pp. 1920–1924.

[25] S. Singh and S-S. Chan, "A multi-level approach to the transport of MPEG-coded video over ATM and some experiments," in *IEEE GLOBECOM '95*, 1995, pp. 1920–1924.

[26] M. Simon, P. Villegas, J. Caballero, and M. Roser, "A general approach to output rate control in video coding," in *Visual Communications and Image Processing*, 1993, vol. 1903, pp. 246 – 254.

[27] D. Reininger, G. Ramamurthy, and D. Raychaudhuri, "VBR MPEG video coding with dynamic bandwidth renegotiation," in *IEEE International Conference on Communications*, June 1995, pp. 1773 – 1777.

[28] W. Ding and B. Liu, "Joint encoder and channel rate control of VBR video over ATM networks," in *Visual Communications and Image Processing*, 1996, vol. 2668, pp. 392 – 407.

A List of Acronyms and Symbols

A dimension of the token pool in a leaky bucket shaper.

B target rate for an MPEG CBR encoder; also average rate of a traffic shaper.

B^{CS} rate of a circuit switched connection used to send raw video.

B^{TDP} bandwidth allocated to a videoconference call using Time Driven Priority.

C capacity of physical links.

C_M maximum coding delay.

CD_M maximum time required to encode and decode a picture (using variable bit rate MPEG encoding).

CBR Constant Bit Rate.

DCT Discrete Cosine Transform.

E_r excess resynchronization delay

F number of bits encoding a frame;

- F_r dimension of a raw picture,
- F^I dimension of an I-frame,
- F^P dimension of a P-frame.
- F_t target picture dimension for the rate control function.

GOP Group Of Pictures.

L number of time frames needed to transfer a packet from sender to receiver with time driven priority.

M number of GOP periods in the time cycle.

MB MacroBlock.

MPEG Motion Picture Expert Group.

N number of video frames in a group of pictures.

Pr propagation delay.

P_s size of a packet.

Q_m minimum queueing delay.

Q_M maximum queueing delay.

$\hat{Q}_M$ statistical bound on queueing delay.

ΔQ maximum queuing delay variation.

S_c coding shaping delay.

S_{Raw}^{CS} network shaping delay in the transmission of raw video over a circuit switched network.

S_{VBR}^{CS} network shaping delay in the transmission of variable bit rate video over a circuit switched network.

S_{Raw}^{Async} network shaping delay due to traffic shaping at the boundaries of an asynchronous packet switched network when transmitting raw video.

S_{VBR}^{Async} network shaping delay due to traffic shaping at the boundaries of an asynchronous packet switched network when transmitting variable bit rate video.

S_{VBR} Scheduling shaping delay.

Sw switching delay.

T video frame period.

TDP Time Driven Priority.

TF Time Frame.

T_f duration of a time frame.

VBR Variable Bit Rate.