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

Mario Baldi*and Yoram Ofek

T. J. Watson Research Center - IBM
PO Box 218, Yorktown Heights, New York 10598

Abstract

Videoconferencing applications require from the network guaranteed bandwidth, small loss probability, and bounded delay. In order for the participants in a videoconference call to interact naturally, the end-to-end delay should be below human perception - about 100 ms. We identify the components of the end-to-end delay in various configurations with the objective of understanding how it can be controlled and reduced.

The first contribution to the end-to-end delay comes from the processing performed on pictures before sending and after receiving them; we call this processing delay. The second contribution is the network delay. Since pictures must be displayed at the same pace at which they had been captured, any variation in the processing and network delay must be compensated before pictures are displayed. This compensation is done by adding resynchronization delay, which is the third component of the end-to-end delay.

We devise these bounds, and hence the related end-to-end delay, going step-by-step through a number of configurations. We study the transmission of both raw video and MPEG video (VBR and CBR) over: (i) dedicated links, (ii) circuit switching, (iii) packet switching with time driven priority, and (iv) asynchronous packet switching. The study shows that a common time reference used with time driven priority can provide adequate results, for video conferencing with MPEG, independent of the network load.

*Vistiting student from Dipartimento di Automatica e Informatica, Politecnico di Torino, phone +39 11 564 7067, fax +39 11 564 7099, e-mail baldi@athena.polito.it

Contents

1 Introduction

The Moving Picture Expert Group (MPEG) encoding was designed for storage and one-way (playback) transmission of quality video where end-to-end delay is not a major concern. The main objective has been to reduce the communication resources requirements while maintaining high video quality.

Using MPEG for videoconferencing is not new. However, a systematic study of various quality of service (QoS) parameters has not been done. Such a study is needed in order to avoid various pitfalls, as shown in this report.

We define the following requirements for interactive ("face-to-face") real-time audio and video communications:
(i) for hearing the end-to-end (sender to receiver) delay should be below 100 ms,
(ii) the video stream should be synchronized with the audio stream (a.k.a. lip-synchronization), which implies that the video end-to-end delay should be kept below 100 ms as well, and
(iii) the media (audio and video) *replay* at the receiving side should be *continuous* (which implies that the replay buffer is never overflowed or underflowed).

The main consequence of the above requirements is that we can "close our eyes" and think that the person on the other side is next to us. If we had very high quality 3D (dimension) video, then with our eyes open we could think that the person on the other side is actually next to us - this is also known as *tele-presence or virtual reality.* So the results presented in this work can have further consequences on the design of virtual reality systems.

We identify the components of the delay and study their nature with the goal of reducing each of them thus decreasing the end-to-end delay. While reducing the end-to-end delay it is important to maintain the video quality by avoiding packet loss and by maintaining the quality of each picture. We concentrate on the minimization of the end-to-end processing delay, i.e., we consider the delay introduced by the combination of encoding/decoding and how it is influenced by the transmission and network queueing delay.

1.1 Scope of the Work

In order to have an in-depth understanding of the videoconferencing problem we divide our investigation into several steps. In each step we increase the complexity of the system we analyze. By doing so we can isolate and understand the various delay components.

We consider two variables: the network type and the video type. There are four network types:

1. Dedicated link, which allows the video stream to use the entire capacity of the link, possibly in a bursty manner. Thus, in this case the link capacity can be much higher than average video stream rate.

2. Circuit switching, which allocates a fraction of the capacity to each video stream. In this case the assumption is that the circuit capacity equals the average video stream rate.

3. Time driven priority [15], which gives the real-time traffic periodic priority over other traffic (both real-time and non real-time). Since all the switching elements are using a common time reference, time driven priority facilitates pipeline forwarding inside the network. As a result, the real-time traffic cannot be adversely affected by other (real-time or non real-time) traffic, i.e., the real-time traffic is protected by a "fire-wall", and the delay jitter can be bounded by a constant.

4. Asynchronous packet switching, which does not have common time reference and even if priority is used congestion can occur. As a result, the delay jitter can be large and packet loss can occur.

We also investigate three types of video streams in order to better understand the actual effect of the video processing on the end-to-end delay.

1. Raw video. In this case the processing in minimal: pictures are captured and sent directly to the network interface.

2. Variable bit rate (VBR) MPEG. Each picture is encoded and immediately sent to the network interface. In this case the video encoder includes small or no buffer.

3. Constant bit rate (CBR) MPEG. The video encoder includes a buffer and rate controller in order to ensure that the bit rate provided to the network interface is constant. In this case an encoded picture may be delayed for several picture periods before it is forwarded to the network interface.

The various configurations are shown in Table 1. Each configuration is characterized by both the video stream and network types. The entries of Table 1 show the structure of this work by identifying the Section in which the corresponding configuration is studied. In Section 6 we summarize the obtained results.

2 The Model

In this section we introduce the key elements of the system dealt with in this work. We focus on quality of service (QoS) issues in a digital videoconferencing system. In particular we are concern with the delay and jitter, the sender-receiver synchronization, and packet loss. Dealing with the quality of video and audio encoding is beyond the scope of this work. In fact, we model the delay as being end-to-end, such that, it includes the jitter compensation and media synchronization. In general, synchronization is needed for continuous play of audio and video streams at the receiving side.

	Dedicated Link	Circuit Switching	Time Driven Priority	Asynchronous Packet Switching
Raw Video (Section 3)	Section 3.1	Section 3.2	Section 3.3	Section 3.4
VBR MPEG (Section 4)	Section 4.2	Section 4.5	Section 4.3	Section 4.4
CBR MPEG (Section 5)	Section 5.4	Section 5.5	Section 5.6	Section 5.7

Table 1: Configurations Considered in this Work.

2.1 Requirements for Interaction

People participating in a videoconference should feel like being face-to-face in the same meeting room. A necessary requirement in order to achieve this goal is to keep the *end-to-end delay* below human perception. The human ear is sensitive to the delay response in a conversation (just think of how annoying are phone calls routed through satellite links); the assumption is that the limit below which the delay is not perceptible is *100 ms* [13].

Of course, the delay requirement is not sufficient, there are other requirements like picture quality, 3D, etc.. However, the end-to-end delay is, in our view, the most essential requirement for interactive real-time applications. The delay requirement is also the least obvious to solve, since it is subject to physical limitation, such as, the speed of light...

Requirement 1 *Video and audio streams are synchronized.*

Since the video stream should be synchronized with the audio the same end-to-end delay constraints applies to video as well. This synchronization is essential since people are used to see even before hearing (consider, for example, a spectator who is attending a concert but is far from the stage). We focus on the end-to-end delay in transmission of video because it is the harder to keep below the 100 ms target.

Requirement 2 *The end-to-end delay of the audio and video streams should be kept below 100 ms.*

The end-to-end delay has three main components that we want to study separately with the aim of reducing them and keep the end-to-end delay below the 100 ms bound:

1. *Processing*: it is needed to transform the audio and video signals in a format suitable for transmission on a digital network. Processing may include also the compression and decompression of the audio and video signals.
2. *Network*: time taken to move data containing the audio and video information from the source to the other participant(s). It also encompasses protocol processing in both sender and receiver(s).
3. *Resynchronization*: digitalization and playing of both audio and video require strict time synchronization between capture of the signal and replaying it. Being the processing and network delays variable, they introduce asynchrony that has to be compensated by adding further delay; this is described more in detail in the next Section.

2.2 Requirements for Visual Quality

Digital transmission of video requires a videocamera capture pictures at a fixed pace (*video frame rate*) and a frame grabber board digitalize them; the time between two subsequent captures is called *video frame period*. The produced bits are sent to the receiver(s) where they are converted back to pictures and displayed on a monitor at the same regular pace at which they have been captured. If the receiver does not display pictures at this same video frame rate, the reconstructed video scene results can be annoying. This is a sort of quality degradation which is not acceptable in this work, i.e.,

Requirement 3 *The receiver continuously display pictures at the average rate they have been captured.*

In order to have continuous playing of the video frames on the receiver side, the end-to-end delay of each picture must be constant, i.e., sender and receiver must be *synchronized*.

Since the network delay component is variable, the receiver has to compensate for this variability (a.k.a. *network delay jitter*) by introducing a *network resynchronization delay*. This is achieved by possibly buffering the received bits and delaying the presentation of the corresponding picture. As shown in Figure 1, the buffer used to compensate the network jitter is usually called *replay buffer*. The overall effect of the replay buffer, is that each picture takes the same time to transit from the sender (point `A` in Figure 1) to the output of the buffer (point `B` in Figure 1); the shorter the network delay experienced, the longer the time the picture spends in the replay buffer (i.e., the network resynchronization delay).

After the system (in particular the replay buffer) is dimensioned, the time a picture spends in moving from the point `A` to the point `B` in Figure 1 is fixed. If a picture experiences a network

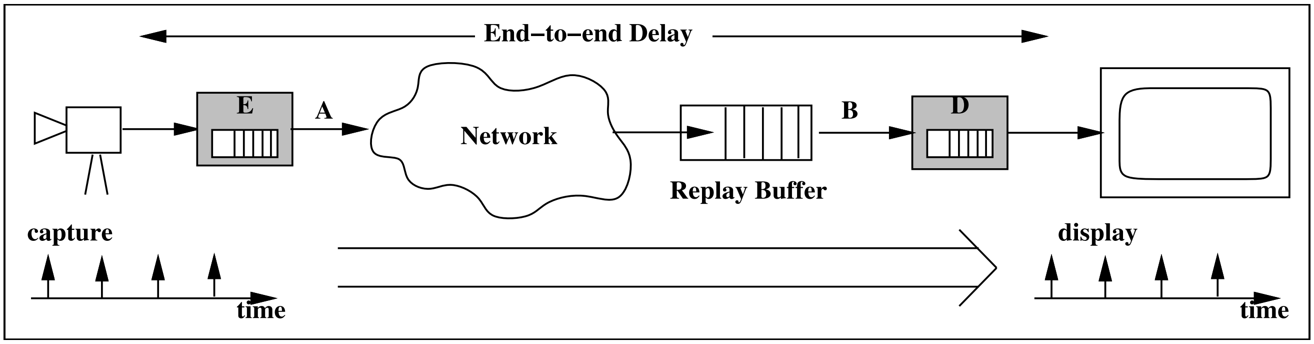

Figure 1: Synchronization in Video Transmission.

delay larger than this time, Requirement 3 is not satisfied. If only part of a picture experiences a too large delay, it cannot be used when displaying a picture. This is not acceptable, especially when the video is compressed by eliminating the temporal redundancy between pictures because the resulting image quality degradation is propagated to the following video frames. and thus it is not acceptable. Hence,

Requirement 4 *The network delay is bounded.*

If this requirement is satisfied, the replay buffer can be dimensioned large enough to compensate for the network jitter and guarantee synchronization.

Since the amount of bits obtained by digitalizing a picture is very large and their transmission very expensive, compression is performed before transmission. This takes a variable amount of time as well as their decompression; the encoder/decoder system introduces a *processing resynchronization delay* that compensates for this variability. In other words, the sum of the time each picture spends in block `E` and in block `D` depicted in Figure 1 is constant; this time is clearly larger than the maximum processing delay experienced by a picture.

Knowledge of the resynchronization delay is the basis for dimensioning the resynchronization components of the system. Since the aim of this work is keeping the end-to-end delay below a perceptible bound, in the following we do not consider resynchronization delays per se. In fact, a wise resynchronization makes the end-to-end delay be the sum of the maximum processing delay and the maximum network delay experienced by any packet. Thus, we focus on these two maximum values in various different scenarios.

2.2.1 Buffering Issues

Figure 1 depicts the replay buffer as a separate function. Actually, network delay variation can be compensated either in the network interface or in the decoder itself, i.e., the replay buffer can be merged with the decoder buffer. In the former case, the compensation can be more efficient (i.e., the network resynchronization delay be minimum) because it can exploit timing

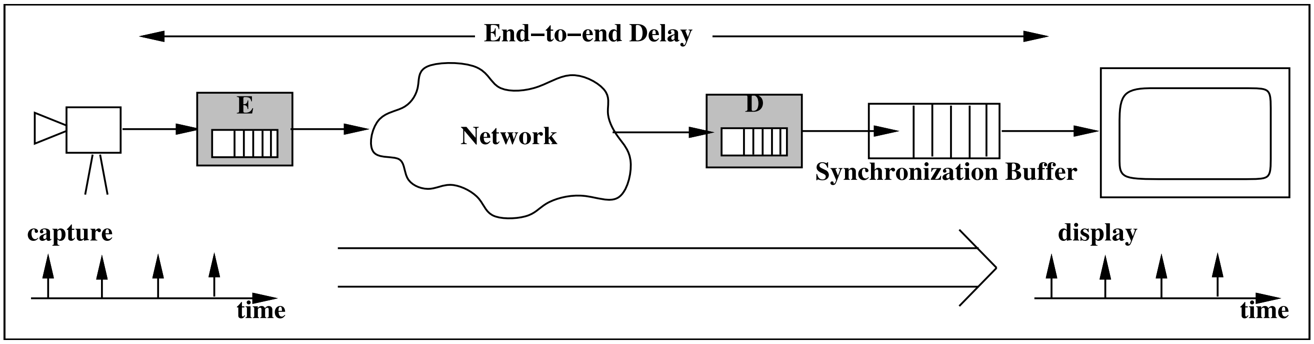

Figure 2: Synchronization Just Before Displaying.

information about the network. Instead, merging the two buffers provide more robustness against misdimensioning of the buffers because the variations of network and processing delay can sometimes compensate each other.

In principle, synchronization can be performed by a single buffer inserted just before the display, as shown in Figure 2. Buffers can be present in the network interface (e.g., if a packet switched network is used incoming bits must be assembled in packets before the header is processed) and in the decoder to store arriving data until there is enough data to be processed and while the decoder is busy. Anyway this buffers are not used for compensating variations in the processing and network delays.

Encoded pictures are decoded as soon as they are received, buffered in the format suitable for displaying, and tagged with the displaying time; when this time has come, the picture is displayed. The synchronization buffer is larger that the buffers used for synchronization before decoding the pictures.

2.3 The System

In our view, a videoconferencing system is composed of the functional blocks depicted in Figure 3. Not all of them are present in any configuration taken into consideration in this work. For example, when transmission of raw video is considered, both the encoder and decoder functionalities are not included in the system.

In this work each of these functional blocks is considered from the viewpoint of the delay it introduces. In this Section only capture card and display are discussed; in the following Sections the other blocks are described as they are first introduced.

A capture card, or frame grabber, takes around 3 ms to digitalize each picture captured by the video camera; this time can be neglected with respect to the objective 100 ms end-to-end delay bound.

When a picture is available on the receiver side in a format suitable for displaying, it is inserted into the video frame buffer. The video adaptor periodically scans the video frame

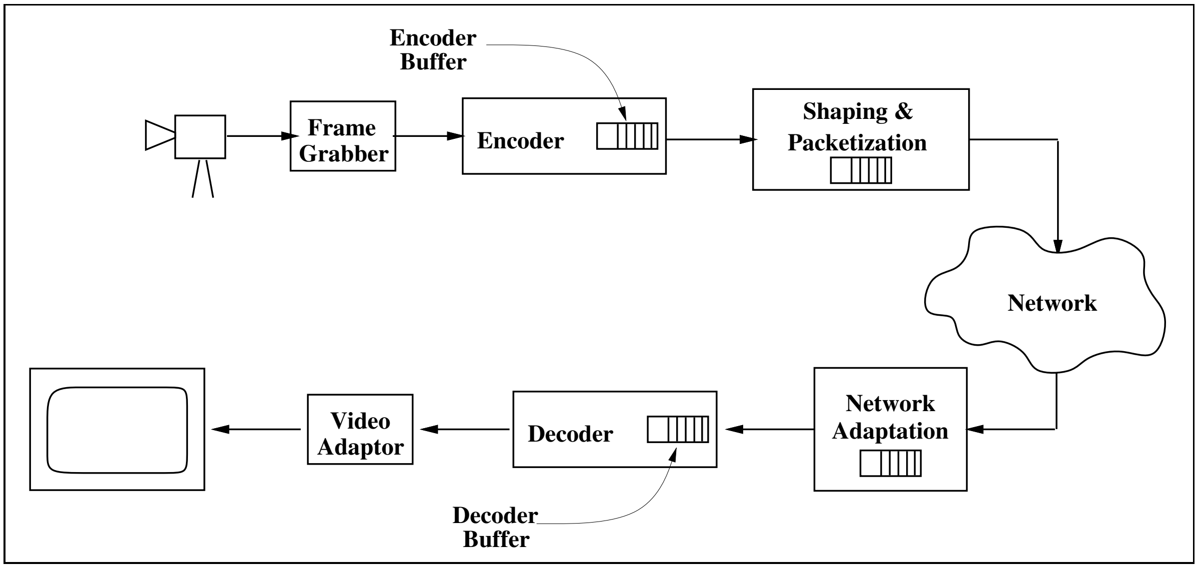

Figure 3: High Level Model of a Videoconferencing System.

buffer and traces the image on the screen according to the information stored for each pixel. The delay introduced by the display depends on the video refreshing frequency. Current monitors work at refreshing frequencies R_f between 60 Hz and 100 Hz, thus introducing a *presentation delay*

$$P_d \in \left[0, \frac{1}{R_f}\right]$$

The bound on the presentation delay is 17 ms for $R_f = 60$ Hz. $P_d = 0$ is obtained by synchronizing the video adaptor with other components of the receiver: the component to be synchronized will be identified for each of the system configurations presented throughout this work. When no synchronization is provided, the value of P_d varies in the given interval according to the instantaneous time relationship between when a picture is inserted in the video frame buffer and when the buffer is scanned for refreshing the display.

3 Transmission of Raw Video

In order to better analyze each component of the end-to-end delay and show in which component of the system it is introduced, an incremental approach is used. We start from the configuration of the videoconferencing system which requires the smallest number of functional blocks and has the smallest number of components in the end-to-end delay. Then, in Sections 4 and 5 we add functional blocks to the configuration and show how the end-to-end delay is affected.

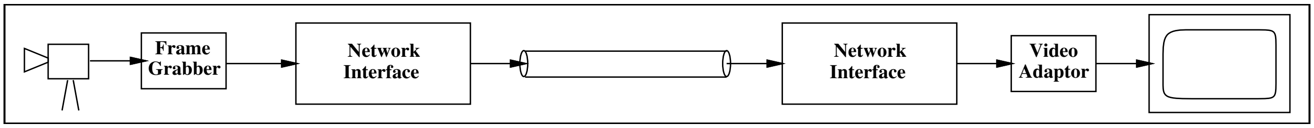

Figure 4: Raw Video over a Dedicated Link

In this first approach we consider the transmission of raw video over a number of different network configurations. Although video is not transmitted in a raw format due to its high bandwidth requirements, it is instructive to study this case. Since in this case we can study in isolation the network delay and the presentation delay[1].

3.1 Dedicated Link Between Sender and Receiver

Figure 4 shows the block diagram of a videoconferencing system in which sender and receiver are directly connected by a link having capacity C. The F_r bits of a picture are made available to the network interface, from the capture card, in a few milliseconds. The picture is then sent without any additional processing at the link speed, as shown in Figure 5. We call *picture dimension* the number of bits encoding a picture.

A buffer is needed on the sender side to match the rate at which bits are produced by the frame grabber, to the link speed C. A buffer is also needed on the receiver side to gather the bits while they are being received from the network; as soon as the last bit of a picture is received, the picture can be moved into the video frame buffer for display. Thus, each bit experiences in the sender and receiver buffers an overall delay which is equal to the *transmission delay* of F_r bits (dimension of a raw picture) over a link of capacity C, i.e., F_r/C: the first bit spends the time only in the receiver buffer, while the last one only in the sender buffer; intermediate bits spend time partly in the sender buffer and partly in the receiver one.

If the time needed for digitizing pictures is neglected, the end-to-end delay is given by

$$\Delta_{Raw}^{Ded} = \frac{F_r}{C} + P + P_d,$$

where F_r is the dimension of a raw picture, P is the *propagation delay*, and P_d is the presentation delay which depends on the refreshing frequency of the monitor and the synchronization relationship between capture card and video adaptor. P_d is null if the video adaptor on the receiver is synchronized with the capture card of the sender.

The propagation delay exists in all the configurations presented in this work, even though in some of them it is concealed inside some other component of the end-to-end delay. The propagation delay is a physical lower bound on the end-to-end delay of the system; if sender and receiver(s) are so far away such that $P > 100$ ms[2], the bound on the end-to-end delay

[1]The time required for digitalizing (capture) a picture is considered relatively small.

[2]Propagation delay of 100 ms in fiber is about 20,000 kilometers or half of the Earth circumference.

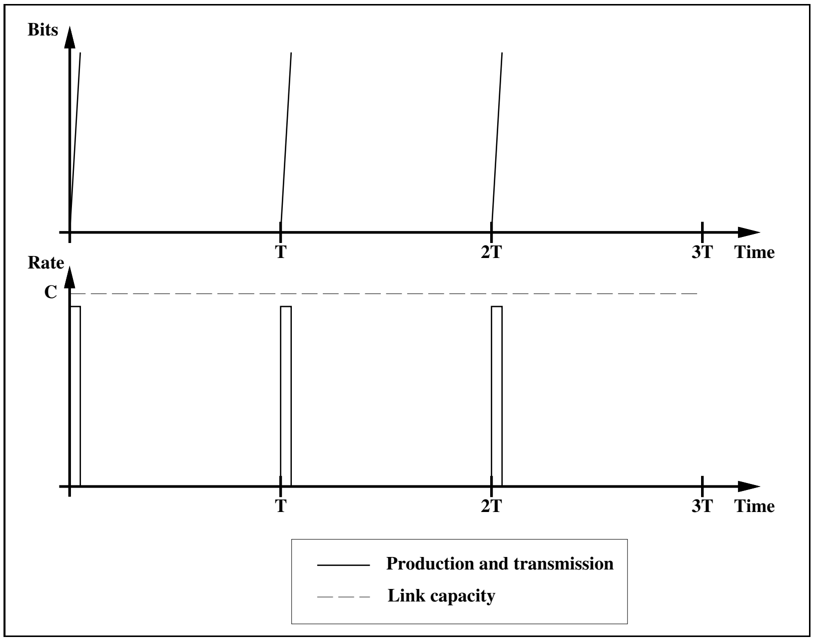

Figure 5: Generation and Transmission of Raw Pictures over a Dedicated Link.

cannot be satisfied (i.e., face-to-face-like interaction is not possible), independently of the configuration used for the videoconferencing system.

F_r depends on the resolution of the pictures (i.e., the number of pixel per picture) and the number of bits used to encode each pixel. For example, a QCIF (Quarter Common Image Format) picture has a definition of 176 x 144 pixels; assuming that each pixel is encoded on 8 bits, $F_r = 198$ kb. Instead a High Definition TeleVision (HDTV) video frame is digitalized using 1920 x 1080 pixels, i.e., 16200 kb. Thus, for example, the transmission delay of a QCIF picture on a 100 Mb/s link is 1.98 ms: even considering the worst case for the presentation delay, 80 ms can be spent in propagation delay over the link, while keeping the end-to-end delay under the 100 ms bound. Instead, an HDTV picture takes 162 ms to be transmitted over a 100 Mb/s link; this is above the 100 ms bound required for interaction and moreover it does not allow for real-time video, as shown in the following.

Real-time video transmission requires that

Requirement 5 *Each picture is transmitted within the video frame period T,*

i.e., the transmission delay must be smaller than T; thus, given the resolution of pictures and the video frame rate, the capacity of the link must be large enough to satisfy

$$C \geq \frac{F_r}{T}. \tag{1}$$

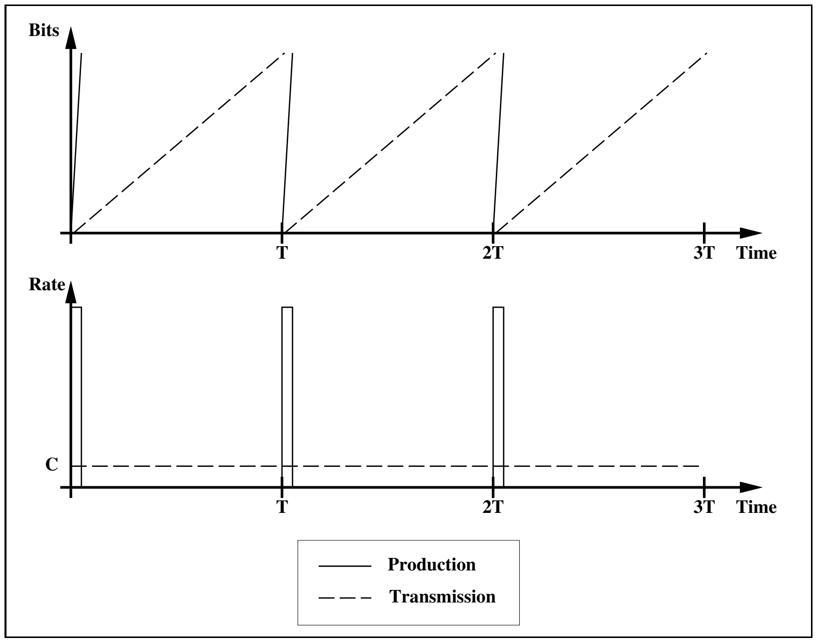

Figure 6: Generation and Transmission of Raw Pictures over a Slow Dedicated Link.

For example, the minimum link capacity needed to transmit video sequences at HDTV definition is 486 Mb/s since the HDTV frame rate is 30 pictures per second. When a minimum capacity link is exploited, the generation and transmission of pictures shows the behavior depicted in Figure 6.

Thus, in a videoconferencing system that transmits raw video over a dedicated link, the main parameter in determining the end-to-end delay is the link speed. The higher the picture definition and the video frame rate, the faster the links. If the video frame rate is low or the propagation delay is large, the lower bound on link speed is not given by (1), but by the bound on the end-to-end delay. This leads to high cost for the videoconferencing system, especially when high definition is required and sender and receiver are far (that is the situation in which videoconference is usually needed).

High speed links are now available also on long distances, but they are very expensive. Thus, dedicating the whole capacity of a link to a videoconference call is not cost effective, particularly over long distances; nevertheless, given the speed of links, the configuration described in this Section provides a lower bound in the end-to-end delay for transmission of raw video. According to other transmission schemes, links are shared among various calls or types of traffic; in the following other schemes are studied with respect to the results shown in this Section.

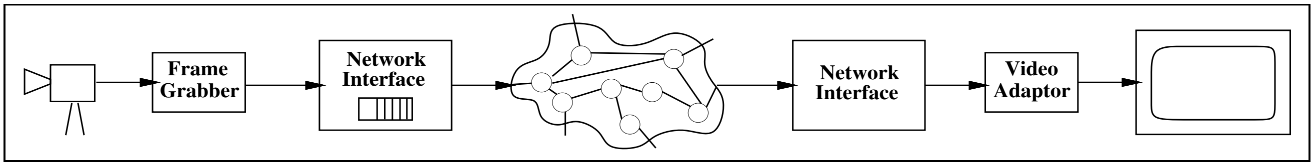

Figure 7: Model of a Videoconferencing System Exploiting a Circuit Switched Network.

3.2 Circuit Switching

In circuit switching link capacity is shared among different calls. A fixed amount of capacity is assigned to each connection using time division multiplexing. The architecture of the videoconferencing system is depicted in Figure 7; sender and receiver communicate as they were connected by a dedicated link, even though its bandwidth is usually smaller than the physical speed of the links on which data actually flow.

A videoconference call is allocated a circuit with bandwidth B by reserving the corresponding fraction of the capacity of each link on the path between sender and receiver. The bits encoding each picture are sent through the circuit at rate B as shown in Figure 8. This introduces a network shaping delay given by:

$$S_n = \frac{F_r}{B} \tag{2}$$

The end-to-end delay is given by

$$\Delta^{CS}_{Raw} = S_n + P + Sw + P_d, \tag{3}$$

where P is the propagation delay and Sw is the *switching delay*. This end-to-end delay is (almost) constant because B, P, and Sw are almost constant; thus, network resynchronization is not needed in the receiver.

According to Requirement 5 for videoconferencing application $S_n \leq T$, i.e.,

$$B \geq \frac{F_r}{T}$$

.

On one hand, the larger B, the smaller the end to end delay; on the other hand, the larger B, the larger the amount of bandwidth is wasted because the circuit is busy only for a time S_n in each video frame period T. During the remaining time $T - S_n$, no other connection or class of traffic can exploit this unused fraction of links capacity.

If minimum bandwidth is allocated to the videoconference call, the end-to-end delay is larger than the video frame period; the lower the video frame rate, the larger the end-to-end delay. For example, the minimum bandwidth required to send QCIF pictures at 15 fps, is 3 Mb/s, but the resulting shaping delay is 67 ms. Nevertheless, if more bandwidth is allocated

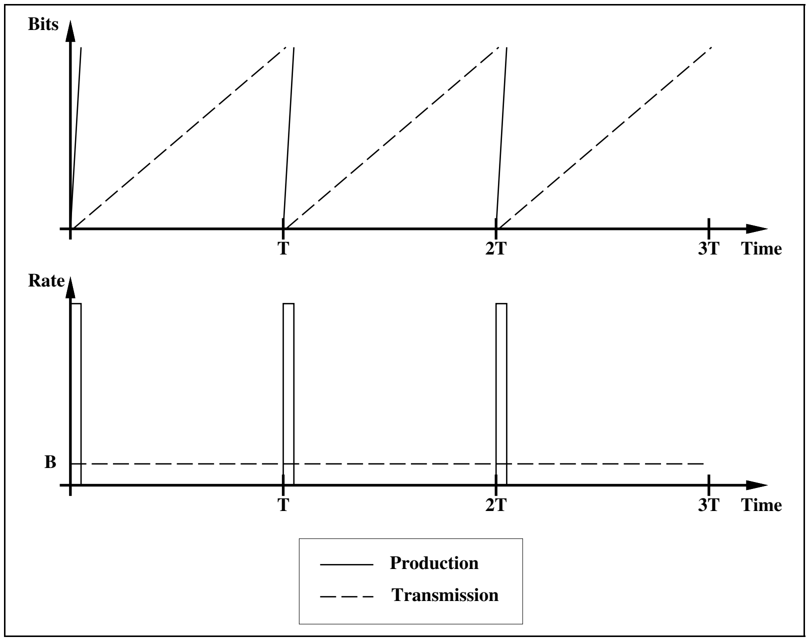

Figure 8: Amount of Bits and Rate in Raw Video Encoding and Transmission over a Circuit Switched Connection.

to decrease the network shaping delay to 30 ms, 50% of the allocated bandwidth is wasted because the circuit is idle for half of the video frame period. On of the motivations for packet switching is to share link capaity in a flexible and efficient manner. However, some packet switching schemes may require to maintain low bandwidth utilization when trying to satisfy real-time traffic requirements.

3.3 Packet Switching with Time Driven Priority

Figure 5 shows that the transmission of a video frame can require use of the link for a short time: the higher the link speed, the longer the idle time. In this Section we show how the link can be used for transmission of other classes of traffic (either requiring guarantees on service quality or "best effort") during its idle periods.

3.3.1 Basic Principles of Time Driven Priority

Time driven priority [15] is a multiplexing scheme aimed at sharing link capacity while guaranteeing sources from uncontrolled delays (or even losses) due to contention in accessing links. The time is divided in *time frames* (TFs) of fixed duration T_f (a typical choice is $T_f = 125\,\mu s$). In each TF a fixed amount of bits $T_f \cdot C$ can be sent on a link: data are grouped in packets and a packet is transmitted during a single TF.

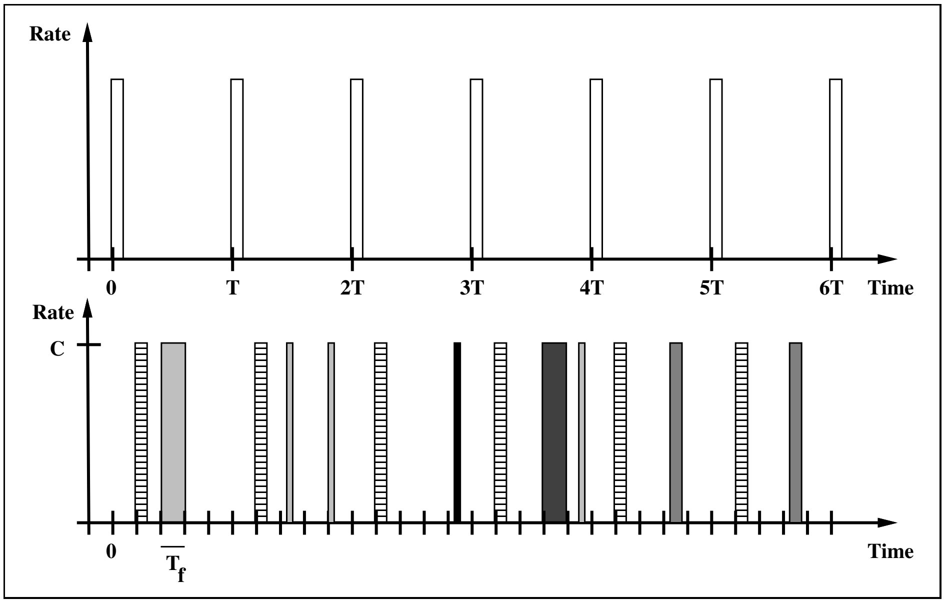

Figure 9: Generation of Raw Pictures and Multiplexing with Time Driven Priority.

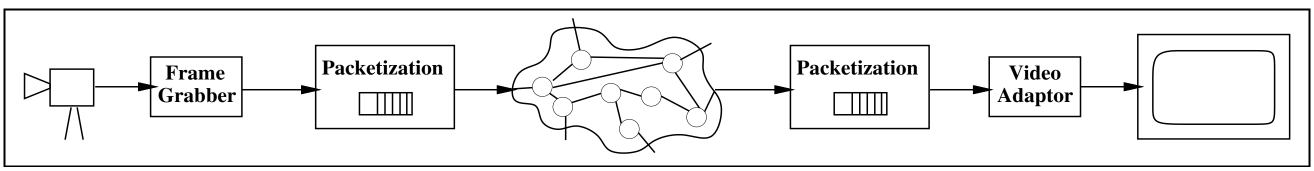

Figure 10: Model of a Videoconferencing System Exploiting a Packet Switched Network.

A picture is divided into one or more parts which are inserted into packets. Each packet is sent in a single TF. For the sake of simplicity, in this work we do not consider the processing required for packetization in either the sender or the receiver and the transmission overhead introduced by the packet headers. In the TFs between the transmission of two subsequent pictures of the same session (and in the remaining time in the same TF), other data can be transmitted (see Figure 9), i.e., traffic multiplexing is performed on the links.

Sender and receiver are not directly connected through a dedicated link; instead, they are connected to intermediate systems, which in turn are interconnected in a general topology, as shown in Figure 10. The sender sends its packets to the intermediate node to which it is directly connected; the packet header identifies the receiver so that the intermediate nodes can route it to the destination.

The network shows the best performance when a packet is forwarded during the TF immediately following the one in which it was received[3]. To guarantee that it is always possible,

[3]A node can happen to require a number of TFs to process the packet; the best TF for the forwarding is

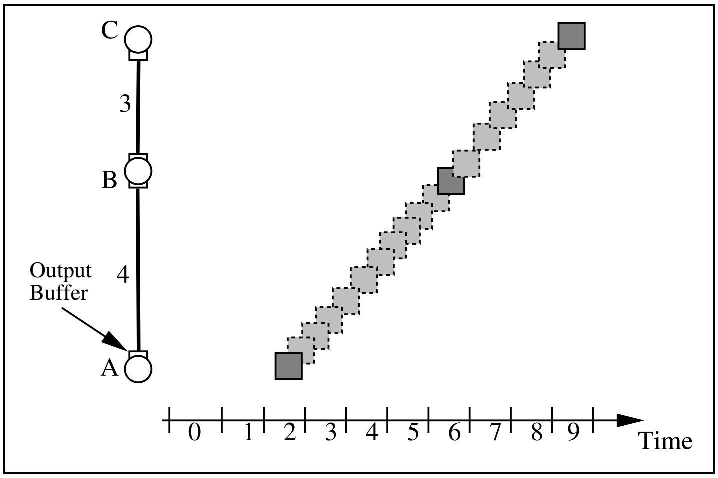

Figure 11: RISC-like Packet Forwarding.

TFs are reserved on the links on the path between source and destination and the source transmits data during the TFs reserved on its outgoing link. A TF is reserved for a videoconference call when a fraction of the $C \cdot T_f$ bits which can be used during it is kept available for forwarding a packet (belonging to the videoconference call) received in the previous TF. The other TFs, and even the non-reserved bits in the reserved TFs, can be used to send data not belonging to the videoconference call. Moreover, if a packet belonging to the videoconference call is not to be forwarded, the allocated bits can be used anyway to send other data.

A reservation of TFs allows nodes to perform *RISC like forwarding* [14]: packets are buffered in intermediate nodes for a fixed time and 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 11. As a consequence the time needed to move a packet from source to destination depends only on the physical topology of the path and has a delay variation or jitter of $2 \cdot T_f$ [15]. This network jitter is fixed and does not depend on traffic load or the path length from source to destination. As a consequence, a packet switched network with time driven priority and resource reservation compiles with Requirement 4 because network delay has a predefined bound. The buffering time in each node is fixed at reservation time: keeping it small enough to avoid overflowing of buffers, the network is not subject to congestion and the related loss.

3.3.2 Transmission of Video and Scheduling

Given the path through the network, a packet takes a time $L \cdot T_f$ to travel from source to destination: the end-to-end delay in the videoconferencing system is

the first after this processing time.

$$\Delta_{Raw}^{TDP} = L \cdot T_f + P_d$$

where $P_d \in [0, 1/R_f]$ is the presentation delay and R_f is the refreshing rate of the display. P_d can be eliminated by synchronizing the video adaptor and the receiver network interface so that the display is refreshed just after the TF in which the picture is received.

The variation of the network delay is small enough to be tolerated, notwithstanding the continuous playing Requirement 3. Hence, there is no need for introducing a network resynchronization delay, i.e., no replay buffer is needed on the receiver side. L depends on the number of hops on the path (it is at least equal to the number of hops), the processing delay inside each node, and the propagation delay between each pair of intermediate systems.

Resource reservation is based on the definition of a *time cycle* which encompasses a fixed number of TFs: all the nodes share the same knowledge of the ordinal position of the current TF inside the time cycle. Bandwidth is allocated to a sender/receiver pair, by reserving a proper number of TFs per time cycle on each link on the path between sender and receiver. In order to have intermediate nodes performing 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 transit from the output buffer of the upstream node to the output buffer of the considered node.

Since raw video has a natural pacing driven by the picture rate, the time cycle duration must be an integer multiple of the video frame period. For the sake of simplicity we consider the time cycle being equal to the video frame period; the obtained results can be extended to the general case by taking into the considerations made in Section 3.3.3. A TF in the time cycle is reserved to the videoconference call, as shown in the lower part of Figure 12. The choice of the TF to be reserved within the time cycle on each involved link is called *scheduling* and it must impacts the maximum number of real-time connections concurrently supported by the network [14]. In the following the impact of scheduling on the end-to-end delay of the videoconference call is presented; anyway, as scheduling is out of the scope of this work and it will not be addressed in details in this work.

The lower part of Figure 12 depicts the TFs allocated on the link connecting the sender to the network and the upper part shows the timing of the frame grabber digitalizing pictures. If the frame grabber is not synchronized to the network interface, the bits produced by the former are buffered for a time $S_n^{AS} \in [0, T]$ while waiting to be transmitted; this is the *application synchronization* component of the *network shaping delay* since it is used to adapt the timing characteristics of the application to the ones of the network. If the clock that drives the pace at which pictures are captured is synchronized to the one used by the network interface to delineate TFs, S_n^{AS} is constant during a videoconference call. If there is a drift between the two, S_n^{AS} slowly increases or decreases; when S_n^{AS} reaches the boundary of its variation interval and wraps around, a TF can be left unused (when S_n^{AS} changes from 0 to T) or a picture overwritten by a new one (when S_n^{AS} changes from T to 0), and hence it is not transmitted.

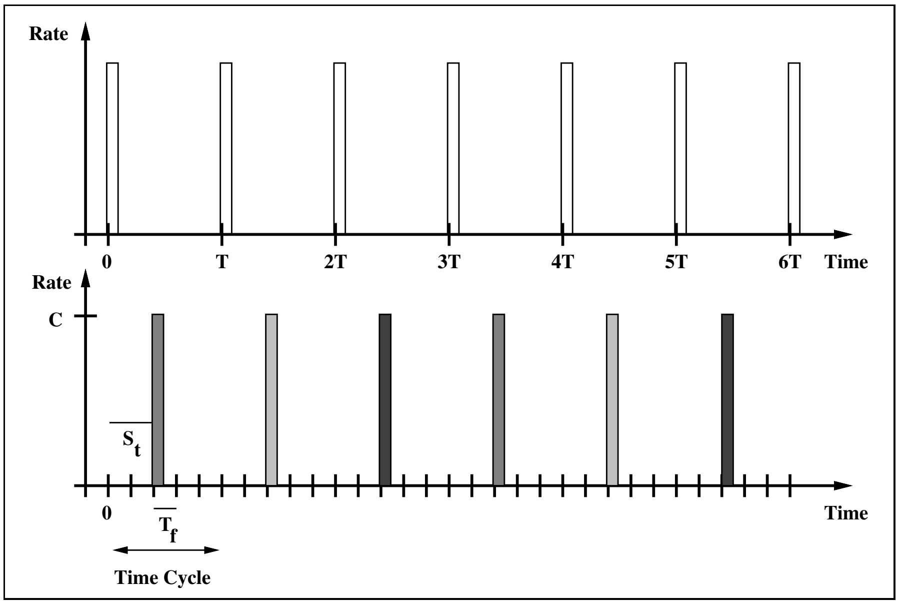

Figure 12: Transmission with Time Driven Priority.

The end to end delay is given by:

$$\Delta_{Raw}^{TDP} = S_n^{AS} + L \cdot T_f + P_d \tag{4}$$

The given expression holds when a picture can be included in a single packet, i.e., $F_r \leq C \cdot T_f$. This can be satisfied when the links are fast enough so that

$$C \geq \frac{F_r}{T_f}$$

Even with QCIF format pictures, capacity of links must be larger than 1.5 Gb/s in order to send a whole picture during a single TF (HDTV definition requires link speed larger than 130 Gb/s. Where such links are not available, a picture can be sent during N_s TFs, where

$$N_s = \left\lceil \frac{F_r}{T_f \cdot C} \right\rceil$$

(e.g., in Figure 13 $N_s = 2$). Thus, the end-to-end delay in a videoconferencing system sending raw video frames using packet switching with time driven priority is given by:

$$\Delta_{Raw}^{TDP} = S_n^{AS} + (N_r - 1) \cdot T_f + L \cdot T_f + P_d$$

where $N_r \geq N_s$ is the number of TFs between the first and the last TF reserved for the transmission of a picture. In fact, it can happen that N_s consecutive available TFs cannot be

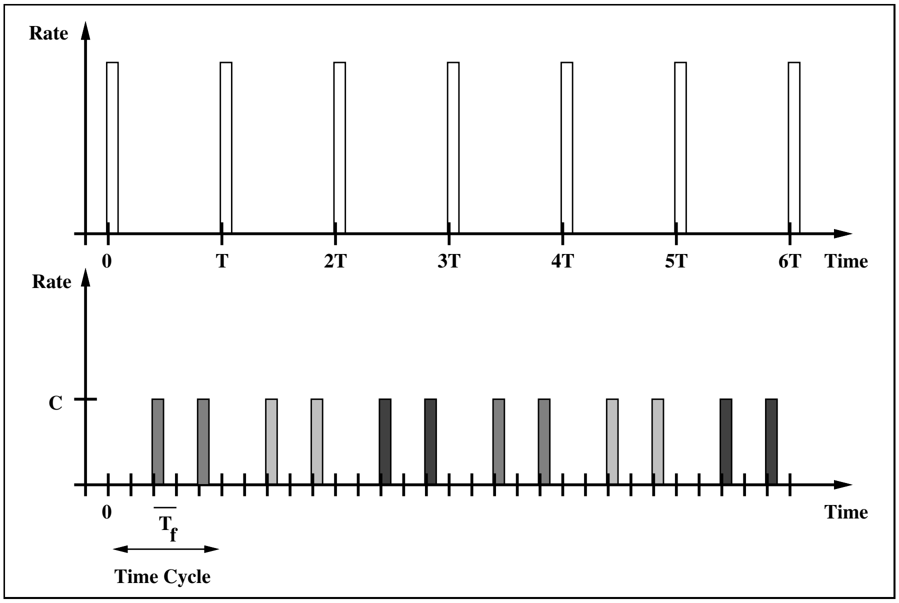

Figure 13: Transmission with Time Driven Priority.

found on each link in the proper position inside the time cycle; thus, non consecutive TFs are reserved and the end-to-end delay is increased.

The end-to-end delay can then be written as[4]

$$\Delta_{Raw}^{TDP} = S_n + L \cdot T_f + P_d \tag{5}$$

The component $L \cdot T_f$ depends on the topological and physical characteristics of the system. We call *network shaping delay*

$$S_n = S_n^{AS} + (N_r - 1) \cdot T_f$$

the delay introduced in order to adapt the rate at which the bits encoding each picture are generated, to network and transmission scheme exploited to transmit them. S_n depends on the link speed, the definition of pictures, and the scheduling.

Using QCIF format at 15 pictures per second, $N_s = 16$ and S_n can vary from 1.8 ms (when capture card and network interface are synchronized and an optimal schedule is found) to 68.8 ms. Considering a scenario in which the path from sender to receiver includes 3 hops (with negligible propagation and processing delay, i.e., $L = 3$), $S_n^{AS} = 0$ and $P_d = 0$, the end-to-end delay is 2.175 ms. In the configuration with direct link in similar conditions ($C = 100$ Mb/s and P is negligible) the end to end delay is 1.98 ms, as was shown in

[4]If the time cycle does not coincide with the video frame period and an optimal schedule is not found, the problems presented in Section 3.3.3 arise.

Section 3.1. Thus, the end-to-end delay is comparable in the two cases, but 97% of the dedicated link capacity is left unused.

As equipment for videoconferencing applications will evolve, synchronization between capture card and network interface will be more likely and the S_n^{AS} component will be null because a picture is sent immediately after it has been digitalized (provided that an optimal schedule does exist). The lack of this synchronization can have a worse impact on the end-to-end delay the lower is the video frame rate. Even though capture card and network interface are synchronized, a large network shaping delay can be due to the impossibility of finding an optimal schedule. Thus, if the route from sender to receiver is characterized by a small value of L, a sub-optimal schedule can be accepted provided that it compiles with the 100 ms end-to-end delay bound. If L is large, an optimal schedule is required to the network when the conference call is submitted; this increases the probability for the call to be blocked.

3.3.3 Multiple Videoconference Calls on the Same Network

In Section 3.3.2 is shown how bandwidth is allocated to a videoconference call by reserving on TF in a time cycle whose duration equals the video frame period T. If more than one videoconferencing system is exploiting the same network and they do not use the same video frame rate, the time cycle is chosen as the minimum common multiplier of all the video frame periods in use.

From the point of view of each of the videoconference calls, the time cycle is an integer multiple M of its own video frame period T. Assuming that each picture is sent within a single TF, bandwidth is allocated by reserving M TFs within the time cycle. If the TFs are evenly distributed within the time cycle, i.e., the time distance between two subsequent TF is T, everything has been stated about the case $M = 1$, still holds.

If the reserved TFs are not evenly distributed, their relative time distance is not T any more and the receiver does not receive pictures at a constant pace. This is equivalent to a variation of the network delay; the less evenly distributed are the TFs, the larger the resulting variation of the picture interarrival time. If this variation is too large according to the continuous playing Requirement 3, the receiver compensates it through a replay buffer and the end-to-end delay is accordingly increased with respect to the one given in (4). If more than one TF is used to send each picture, the choice of the TFs to be reserved impacts the end-to-end delay; a detailed analysis of the yielded end-to-end delay is outside the scope of this work.

In general, scheduling is important to both network performance and end-to-end delay of videoconferencing applications. The two sometimes impose conflicting optimality criteria for scheduling and thus a trade-off must be found among them [1]. A videoconference application requiring a bandwidth reservation to the network, can ask for a schedule that minimizes the end-to-end delay and does not require a resynchronization capable receiver, at the risk of getting the request refused. Alternatively, if the receiver is equipped with a replay buffer to compensate the variation of the network delay introduced by scheduling, it can accept a schedule that increases the end-to-end delay.

The larger M, the harder is for the scheduling to satisfy the optimality criteria. In order to keep M small, videoconferencing application should choose video frame rate that are integer multiples one of the other. The largest video frame period can be chosen as time cycle in the network.

3.4 Asynchronous Packet Switching

The videoconferencing system can be built on a traditional asynchronous packet switching network which statistically multiplex data on the links. The F_r bits encoding a picture are inserted in packets of (maximum) dimension P_s which are sent into the network at the full link speed. In the network packets experience fixed delay due to transmission (F_r/C) and propagation (P), and variable delay due to queueing.

Queueing introduces a network jitter that must be compensated by a replay buffer as described in Section 2.2. According to Requirement 4, the network delay must be bound, i.e.,

Assumption 1 *A bound Q_M on the queueing delay does exist.*

Packets are queued in network nodes while they are contending for a busy link; thus, the queueing delay depends on the load in the network and it does not have a deterministic bound. A bound can be guessed but to have a low probability of breaking it, it must be large. Moreover, the queueing delay can be kept smaller by avoiding bursts of packets that increase queue dimension; for this purpose traffic is shaped at the boundaries of the network.

In this Section we analyze the contribution of network resynchronization and traffic shaping on the end-to-end delay.

3.4.1 Network Resynchronization Delay

Given the bound Q_M on the queueing delay, each packet that has experienced a shorter queueing delay is delayed in the replay buffer. This is shown in Figure 14: the upper diagram shows the arrival time of pictures to the replay buffer, while the lower shows the exit time. The resulting end-to-end delay of pictures is

$$\Delta_{Raw}^{Async} = \frac{F_r}{C} + P + Q_M + P_d. \tag{6}$$

If the network interfaces of sender and receiver are not synchronized, the latter is not able to determine the queueing delay experienced by a packet. Thus, the first packet received for the videoconference call is buffered for a time that allows resynchronization in the worst case: it is assumed that the packet has experienced the minimum queueing delay Q_m and is buffered for a time

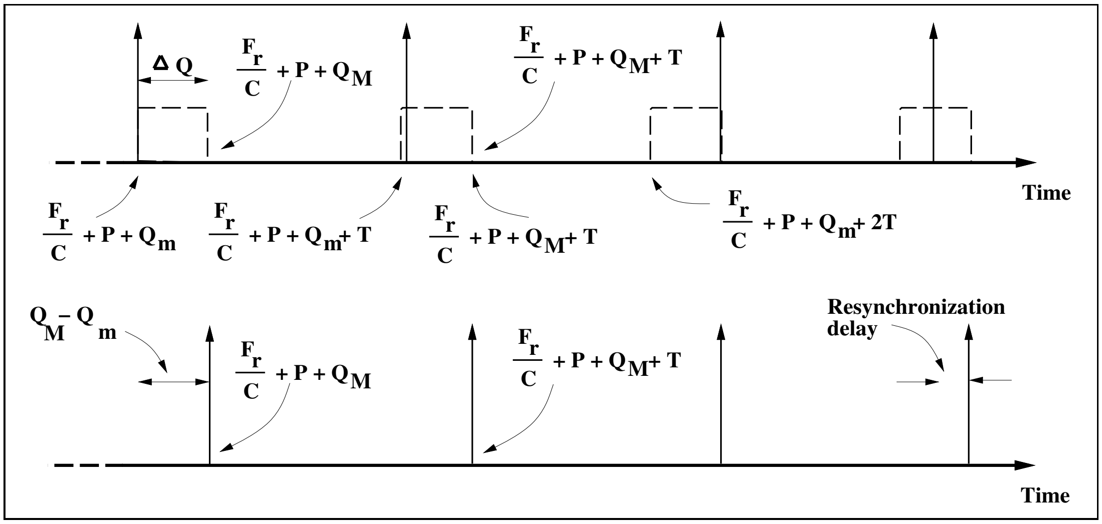

Figure 14: Network Resynchronization Delay to Compensate Queueing Delay Variation.

$$\Delta Q = Q_M - Q_m \tag{7}$$

The following packets are resynchronized accordingly, as shown in Figure 14, because they are retrieved from the replay buffer at the constant pace at which pictures are displayed.

The upper diagram of Figure 15 shows a resynchronization example when the first packet experiences an actual queueing delay Q_M. The middle diagram shows the timing of packets exiting in case the receiver does not know the actual queueing delay experienced by the first packet, delays it by ΔQ according to (7). As a consequence, the overall delay experienced by following packets due to queueing 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), does not delay it at all and the exiting time from the replay buffer are the ones depicted in the lower diagram in Figure 15.

In summary, the end-to-end delay is given by

$$\Delta_{Raw}^{Async} = \frac{F_r}{C} + P + Q_M + E_r + P_d$$

where $E_r \in [0, \Delta Q]$ is the *excess resynchronization delay*; it is constant over a videoconference call and depends on the queueing delay experienced by the first packet received, $E_r = 0$ when sender and receiver network interfaces are synchronized.

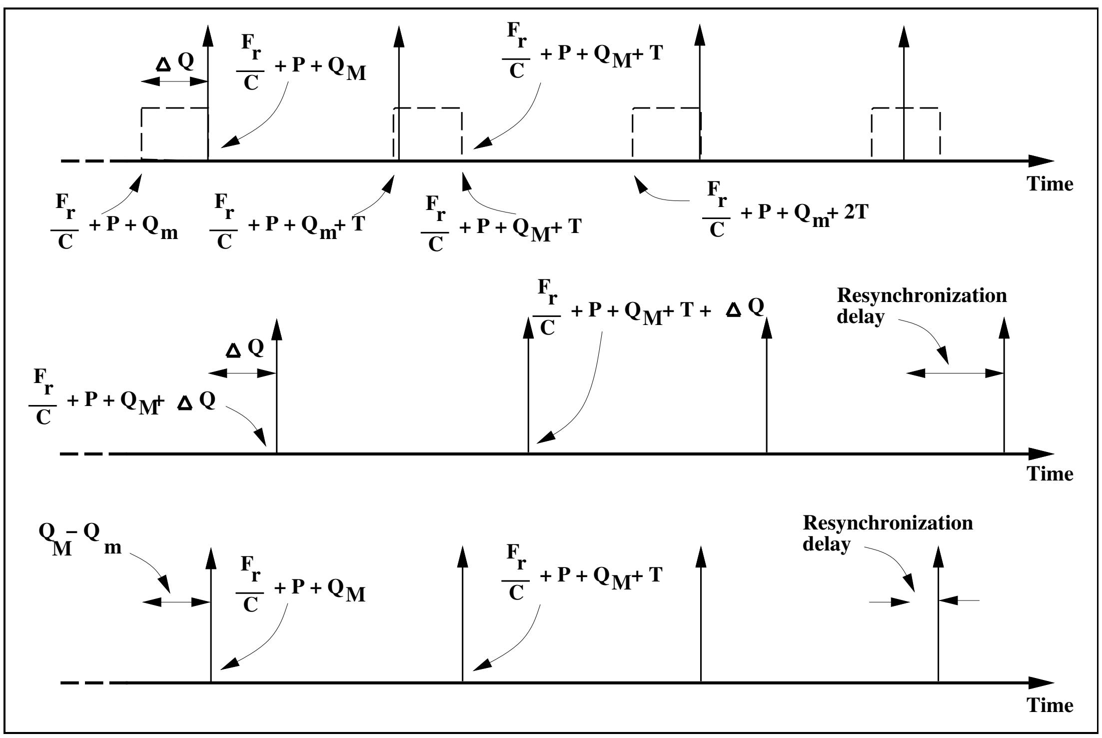

Figure 15: Network Resynchronization Delay to Compensate Queueing Delay Variation.

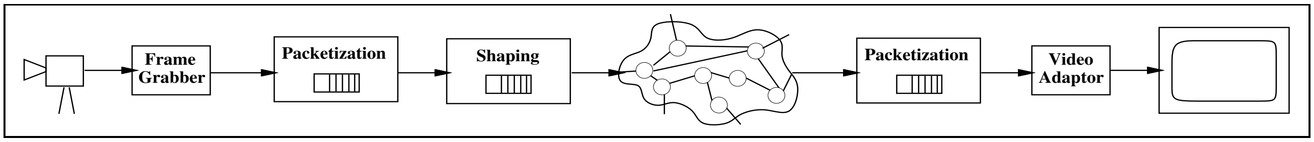

Figure 16: Model of a Videoconferencing System Exploiting a Packet Switched Network with Traffic Shaping at Network Boundaries.

3.4.2 Traffic Shaping at Network Boundaries

Resources must be allocated into the network in order to guarantee the bound Q_M on the queuing delay (and possibly limit the amount of data lost in the network due to congestion). Since transmission of large amounts of data at link speed (*bursts*) causes sudden growth of queues, large amount of resources is reserved for bursty sources: limits queues growth by keeping link utilization low.

Assumption 2 *Traffic shaping is used at the network boundary to provide an average bandwidth between sender and receiver while keeping the burstiness below a predefined level.*

The source shapes the traffic it is generating by using a traffic shaping mechanisms such as a *leaky bucket* [2], as shown in Figure 16. The bursty traffic generated by a source is shaped by a leaky bucket into a flow having average rate B and controlled burstiness. A bucket is assumed to contain at most A tokens and it is filled at the rate of B tokens per second. When packets are presented to the traffic shaper, they are sent into the network at link speed[5] as long as the number of tokens in the bucket is larger than the packets dimension (in data units). A token is removed from the bucket for each data unit sent. We assume to exploit a buffered leaky bucket which is modeled as shown in Figure 17. When the bucket is empty, packets may be dropped or buffered (also known as *buffered leaky bucket*) until enough tokens are present in the bucket to send the packet. This buffer adapts the source rate to the characteristics of the shaped traffic at the expenses of a network shaping delay. Since, given the video frame rate and the resolution, the traffic characteristics of a raw video sequence are known in advance, the buffer can be dimensioned properly so that it never overflows.

The choice of the parameters of the traffic shaper determines its effects on the end-to-end delay. According to Requirement 5, as we are dealing with real-time video transmission, the time taken to refill the token bucket must be smaller than the video frame period, i.e.,

$$B \geq \frac{F_r}{T}. \tag{8}$$

If $A \geq F_r$ the whole picture is inserted in packets which are sent at the link speed as soon as it is captured, as shown in Figure 18 where $A = P_s$. Traffic shaping does not have any effect

[5]There is a variant called dual leaky bucket that sends burst of data at a rate lower than the link speed. Such a traffic shaper can be built by cascading two simple leaky buckets.

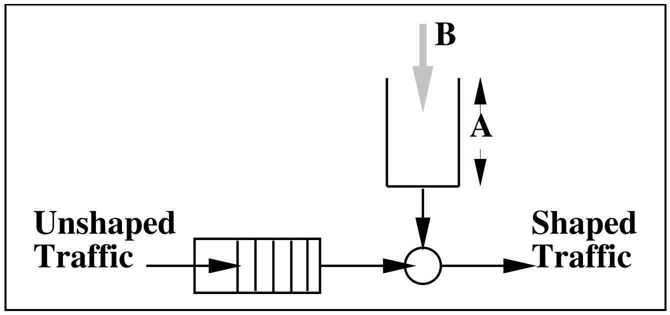

Figure 17: Model of a Leaky Bucket.

and the average rate is F_r/T, as determined by the video frame rate $(1/T)$ and dimension (F_r), and the end-to-end delay is given by (6).

If $A < F_r$, A/P_s packets of dimension P_s are sent as soon as the picture is digitalized, as shown in Figure 19 where $A = P_s$; afterwards, the time needed to generate P_s tokens (i.e., P_s/B) is waited. Thus, the network shaping delay S_n is given by the time needed to generate $F_r - A$ tokens (since, thanks to (8) the token bucket is full before starting transmitting a frame):

$$S_n = \frac{F_r - A}{B} \tag{9}$$

The time needed to transmit a picture being the network shaping delay S_n plus the transmission delay of a packet (P_s/C), the end-to-end delay is given by:

$$\Delta_{Raw}^{Async-Sh} = S_n + \frac{P_s}{C} + P + Q_M + E_r + P_d \tag{10}$$

where P is the propagation delay on the links on the path between sender and receiver, Q_M is the maximum queueing delay, and $E_r \in [0, \Delta Q]$ is the excess resynchronization delay.

Comparing (6) and (10) the main difference is that in the former, where the burstiness is maximum, the network shaping delay is not present. Nevertheless, the lack of traffic shaping at network boundaries results is lower link utilization or larger maximum queueing delay Q_M.

The maximum queueing delay delivered by the network depends on the reservation and queueing policy used by nodes. For example, if network nodes are performing weighted fair queueing, the bound is inversely proportional to the allocated bandwidth [3]:

$$Q_M = o\left(\frac{1}{B}\right)$$

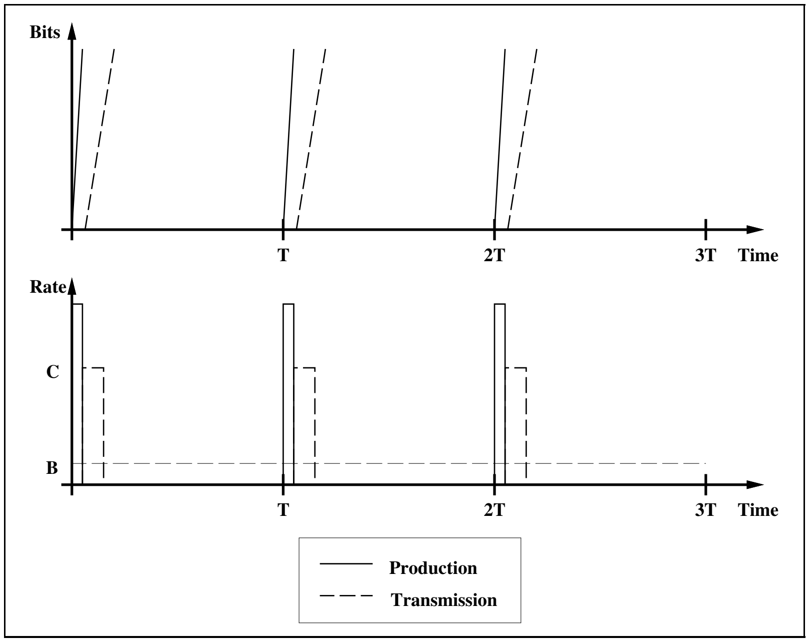

Figure 18: High Burstiness Transmission of Raw Video over Packet Switched Networks.

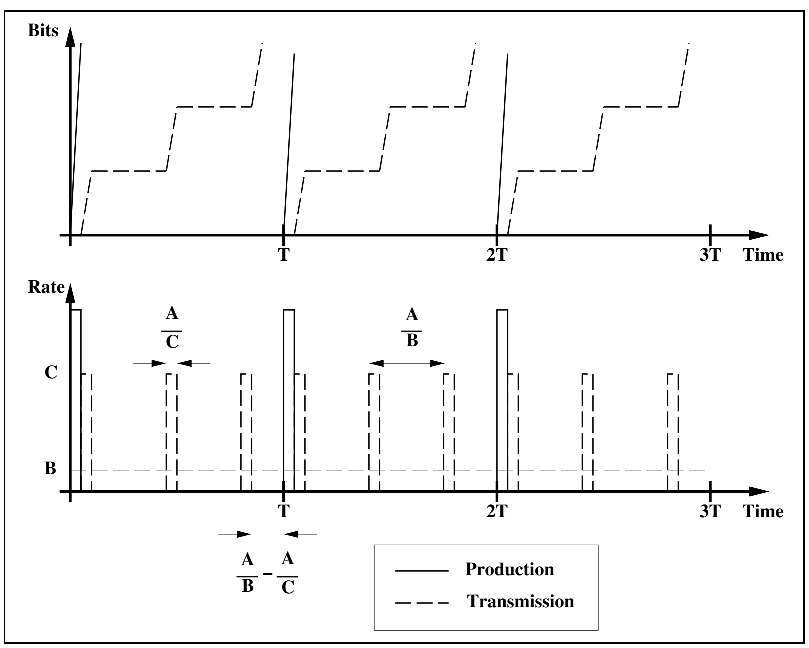

Figure 19: Low Burstiness Transmission of Raw Video over Packet Switched Networks.

4 Transmission of VBR MPEG Video

Transmission of raw video requires large amount of bandwidth, particularly if the end-to-end delay has to be kept under the 100 ms bound. Thus, compression is used to reduce the amount of bits needed to encode each picture by exploiting spatial and possibly temporal redundancy present inside pictures and between subsequent pictures, respectively. The compressed pictures are sent over the network to the receiver where they are decoded and display.

4.1 MPEG Overview

The Motion Picture Expert Group (MPEG) [11, 25, 8] video encoding standard was designed for digital storage of quality video for later re-playing. A digitalized uncompressed image is composed of one matrix of nxm pixels of luminance information and two matrices of $(n/2)$x$(m/2)$ pixels of chrominance information. The encoder receives in input a sequence of digitalized pictures and encodes each of them in one of two different ways[6].

4.1.1 Intra-frame Coding

Spatial redundancy inside the picture is exploited to gain compression. 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; it yields an 8x8 matrix with the coefficients of the spatial frequencies present in the block.

Each coefficient is *quantized* by integer dividing it by an integer called quantization stepsize. Actually, a different quantization stepsize is used for each coefficient because human eye is more sensitive to high spatial frequencies [26]: high frequency coefficients are quantized more coarsely than low frequency ones, but the perceived visual quality does not degrade. Thus, each coefficient is multiplied by the corresponding element of an 8x8 quantization matrix.

The result of the quantization is run-length encoded in order to minimize the number of bits needed to represent it thanks to the fact that many coefficients (particularly the high frequency ones) are zero. Finally, the run-length encoded symbol sequence is Huffman Encoded. The picture encoded in this fashion is called *I-frame*.

4.1.2 Predictive Coding

Temporal redundancy is eliminated to reduce the number of bits needed to encode the picture, with respect to I-frames. The picture is divided into *Macro Blocks* (MBs), each composed of

[6] Actually, a third type of encoding, called bidirectional predictive coding, exists. In order for a picture to be coded, a reference subsequent picture must be captured and coded. This introduces a delay of some frame periods that we consider not acceptable if the end-to-end delay be kept below 100 ms. Thus, this compression technique is not considered.

a 16x16 pixel matrix of luminance information and the two corresponding 8x8 pixel blocks of the two chrominance components. Given a MB to encode, *motion estimation* is performed, 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.

Pictures coded in this fashion are called *P-frames* and their dimension is from 2 to 4 times smaller than I-frames. The smaller the difference between the MB being coded and the reference one, the higher the compression gained through DCT and run-length/Huffman encoding. The more similar two subsequent pictures, the higher probability of finding a MB "similar" to the one being coded and thus yielding better compression; subsequent pictures are similar if the scene is slow moving because if does not change much from a video frame period to the other.

4.1.3 Bit Generation Rate and Quantization

A video sequence is compressed by encoding a picture out of N as I-frame, and the remaining $N - 1$ as P-frames; the sequence starting with one I-frame and terminating with the last consecutive P-frames is called a *Group Of Pictures* (GOP). The larger N, the smaller the overall amount of bits used to encode a scene. Nevertheless, if the network introduces an error in an encoded picture, it propagates in the following ones; the next I-frame is the first picture not affected by the error and stops its propagation, i.e., the larger N the more sensitive to errors the stream.

The rate of the bit stream produced by the encoder has high variability: each picture is encoded within the video frame period, but a P-frame is from 2 to 4 times smaller than an I-frame. Moreover, even video frames of the same type have different dimension: the dimension of an I-frame depends on the spatial redundancy present in the image, while the P-frame one depends also on the amount of temporal redundancy. Figure 20 shows the amount of bits produced by the software MPEG encoder `dvdenc` [17] during the encoding of four video sequences with a resolution of 720x480 pixels; an image from each sequence is shown in Figure 21. The overall number of bits encoding each sequence and the way it is distributed among the pictures are different due to the different amount of spatial (detail) and temporal (motion) redundancy in each scene.

A different quantization matrix is used for a MB, depending on the type of coding (intra-frame or predictive) being performed on it. Intra-frame coded MBs typically contain energy at all frequencies and coarse quantization of low frequencies is more noticeable to human eye than high frequencies. Instead, predictive coded MBs have higher energy at high frequencies and human eye is less sensitive to coarse quantization of predictive coded MBs.

Moreover, human eye is more sensitive to quantization error in flat areas, than very detailed zones [26]. Thus, the actual quantization step size is obtained by multiplying the quantization

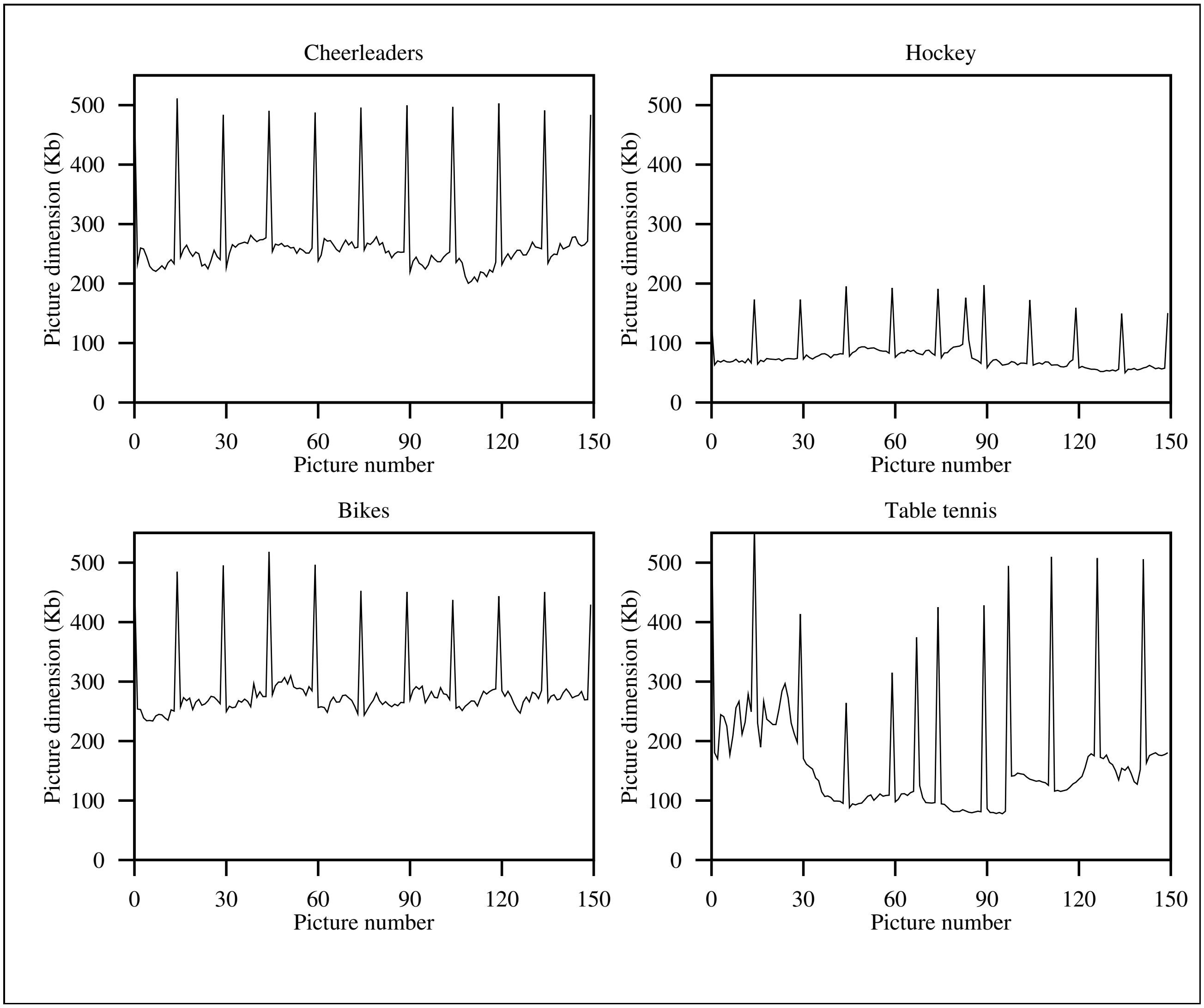

Figure 20: Natural Dimension of MPEG Encoded Pictures.

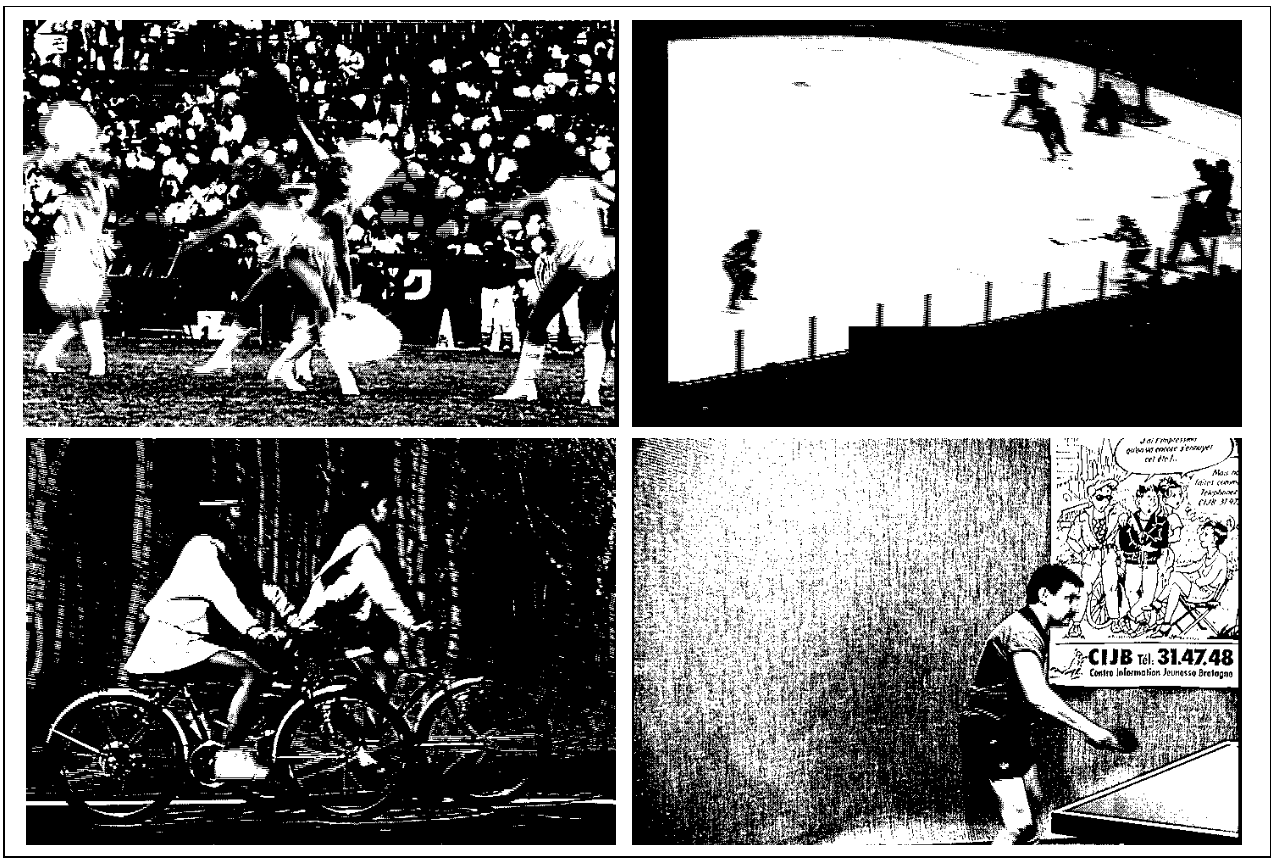

Figure 21: Images from the Four Video Sequences Used in the Experiments.

matrix by a *MB local activity* factor that changes on a MB by MB basis with the goal of keeping constant the visual quality throughout the picture while minimizing the number of bits produced. The way in which the MB local activity factor is calculated is dependent on the implementation of the encoder and is based on the estimation of the "activity" level of the MB [9] [26].

Moreover, a *global distortion level* factor that possibly changes on a picture by picture basis, provides overall control on picture dimension. The product of the MB local activity factor p_{mb} and the global distortion level factor G gives the *quantization parameter* Q_{mb} for MB mb, i.e.,

$$Q_{mb} = p_{mb} \cdot G$$

In summary, quantization is performed according to the following formula:

$$\hat{c}_{i,j}^{mb} = \frac{c_{i,j}^{mb}}{q_{i,j}^{\{I|P\}} \cdot Q_{mb}}$$

where $c_{i,j}^{mb}$ is the (i,j) DCT coefficient of MB mb, $\hat{c}_{i,j}^{mb}$ is its quantized value, Q_{mb} is the for MB mb, and $q_{i,j}^{\{I|P\}}$ is element (i,j) of the quantization matrix used for I-frame or P-frame coding, according to the type of encoding being performed on the current MB.

Four sequences were encoded using four values for the global distortion level G, which was kept constant over the whole video sequence. The dimension yielded for the pictures is plotted in Figure 22: the larger G, the larger the pictures. For small values of G (finer quantization), there is a small difference between the dimension of I-frames and P-frames. In fact, on one side, P-frames dimension grows because the small DCT coefficients of differentially encoded MBs are not reduced to zero by the fine quantization. On the other side the high frequency DCT coefficients of intra-frame coded MBs are coarsely quantized by the corresponding elements of the quantization matrix. This is confirmed by the plots in Figure 23 which show that for $G = 1$ the Signal to Noise Ratio (SNR) due to the encoding[7] is smaller for I-frames than for P-frames. Figure 24 plots the average quantization parameter used on each picture; even though it changes due to the characteristics of the pictures (i.e., because b_{mb} changes), it is almost constant. The variation is larger for larger values of G because G amplifies b_{mb} variations.

In the following we study the end-to-end delay in a videoconferencing system that uses VBR MPEG compression; the impact of different transmission schemes on the delay is taken into account.

[7] The SNR is calculated according to the formula reported in Section B.4. The SNR is used to provide a mathematical measure of the quality of a picture, but it does not exactly correspond to the perceived visual quality because it does not take into account the characteristics of the human visual system. Thus, even though coarse quantization of high frequency components introduces an error, it does not necessarily degrade the perceived visual quality.

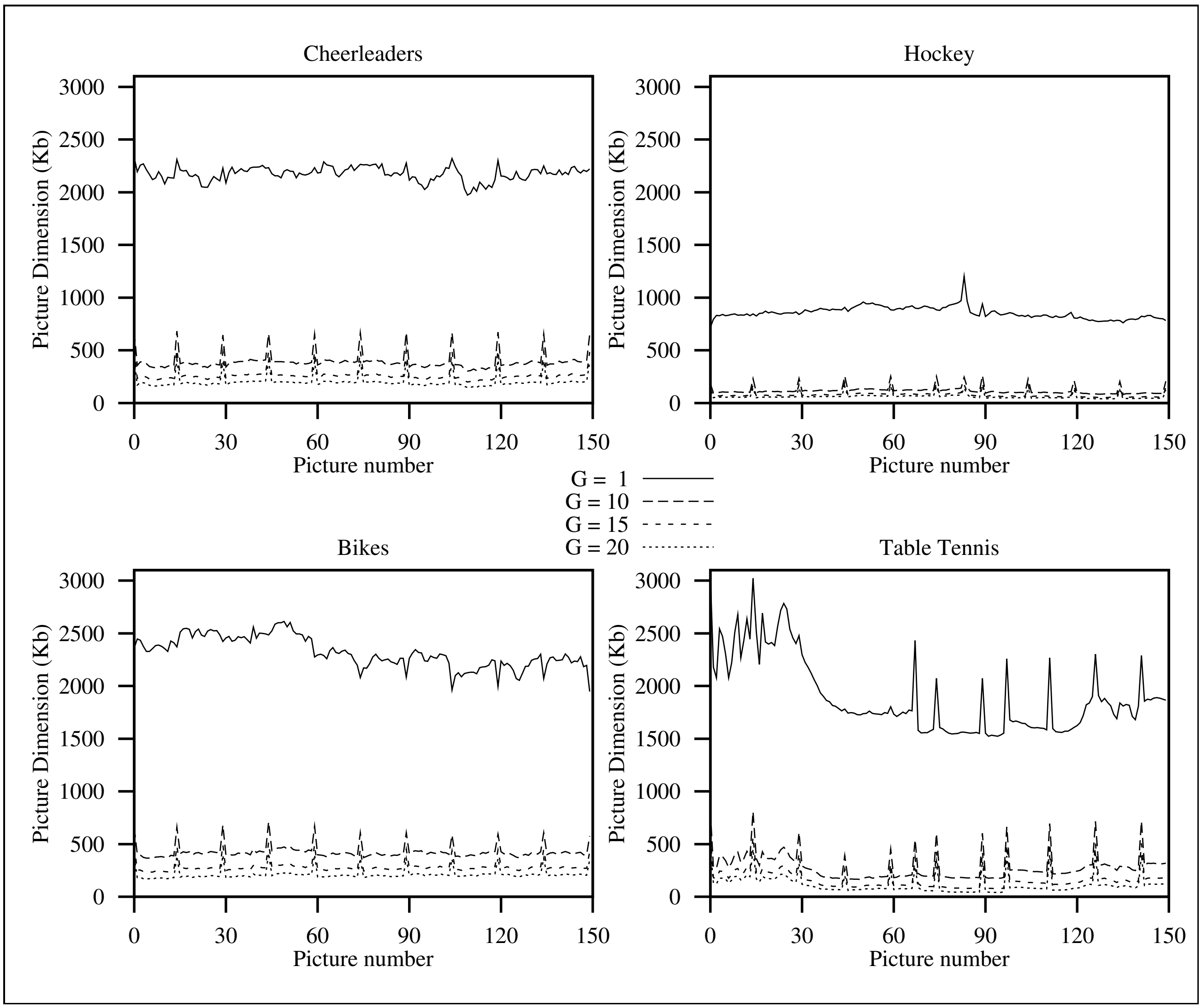

Figure 22: Dimension of Pictures Coded according to MPEG Standard.

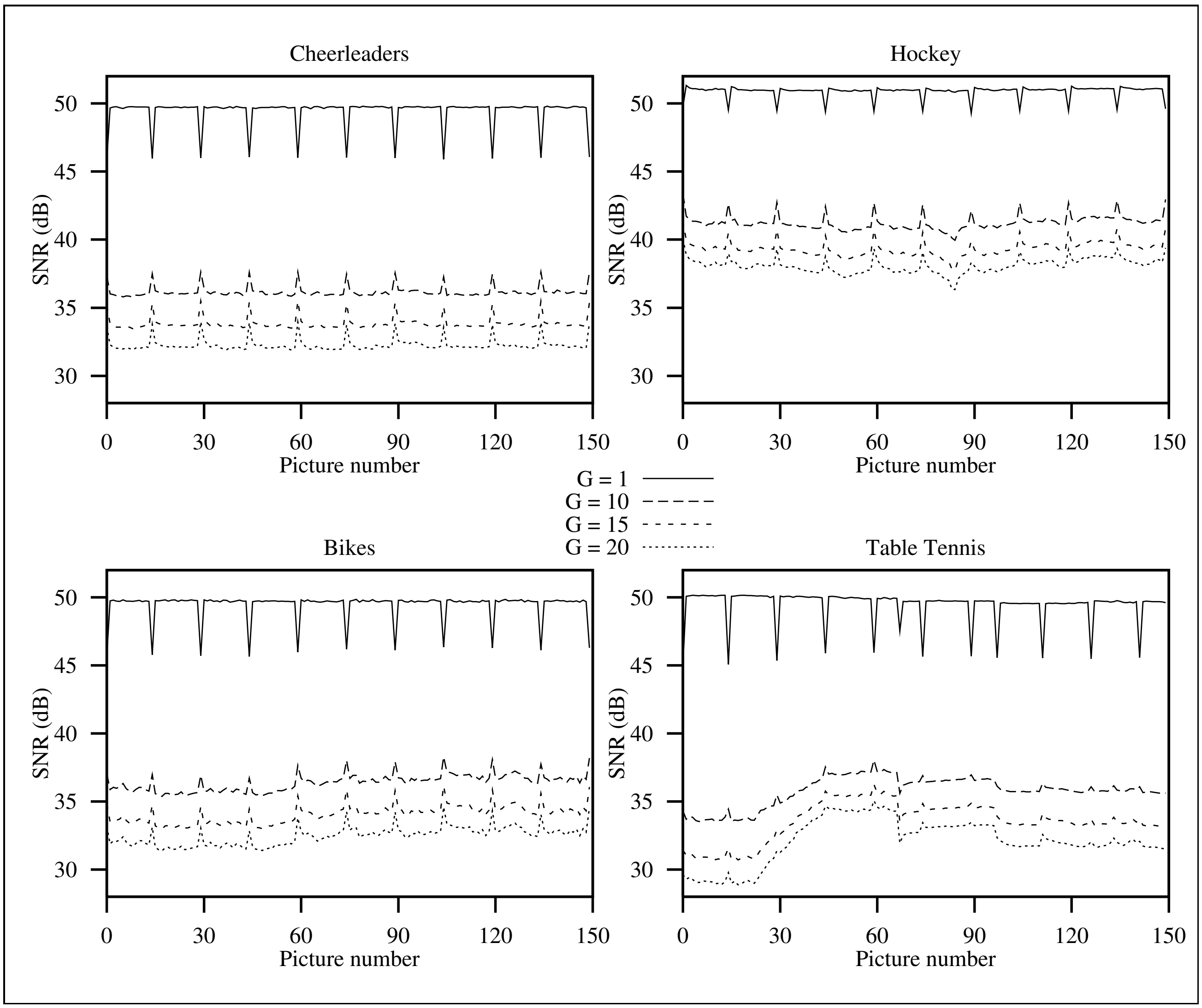

Figure 23: SNR of VBR MPEG Encoded Video Sequences.

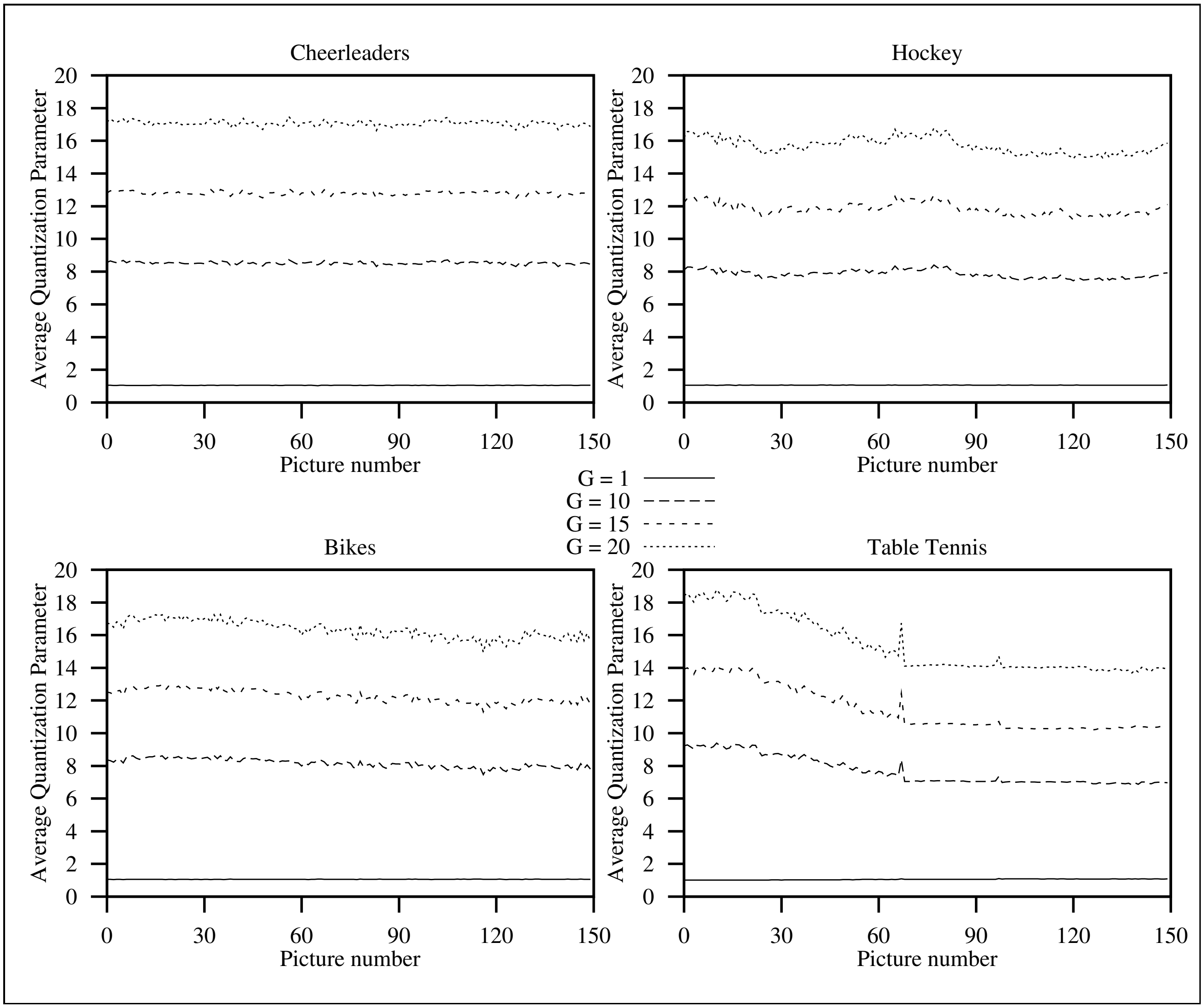

Figure 24: Average Quantization Parameter Used for VBR Encoding according to MPEG Standard.

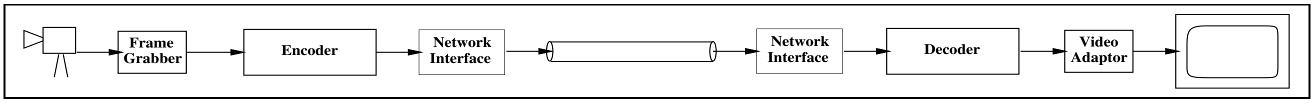

Figure 25: Videoconferencing System Configuration with Dedicated Link between Sender and Receiver.

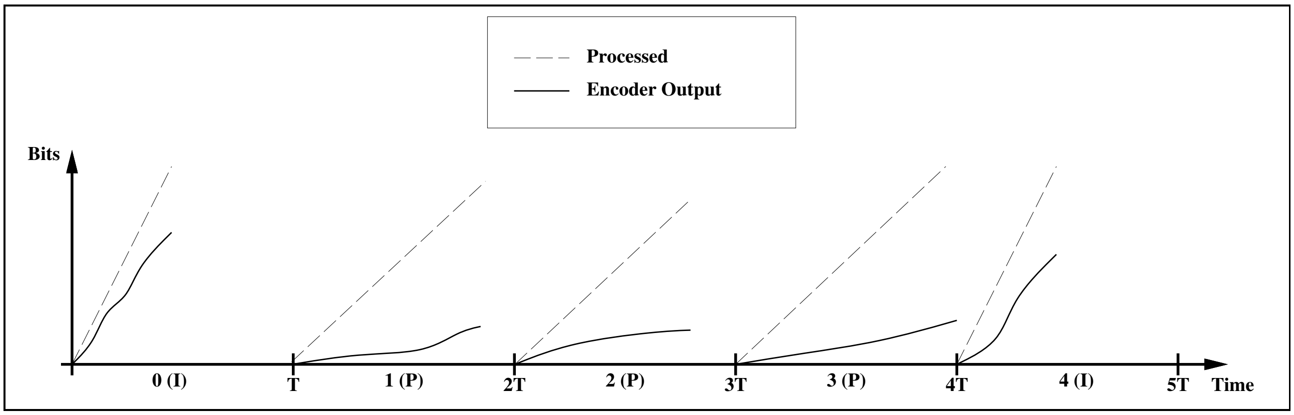

Figure 26: Bit Production of a Natural VBR MPEG Encoder.

4.2 Dedicated Link Between Sender and Receiver

Minimum delay can be obtained if sender and receiver are directly connected by a dedicated link, as shown in Figure 25. The encoder processes pictures and produces the bits encoding them as I-frames or as P-frames with the timing shown in Figure 26: the time taken to encode a picture is not constant as well as the amount of bits produced.

In principle, if the link speed is higher than the natural instantaneous rate of the encoder[8], bits can be sent over the dedicated link as soon as they are produced. After a propagation delay P, bits get to the decoder which decodes the stream generating pictures in the format suitable for the video adaptor.

The time needed to encode a picture is not constant; also decoding time is not constant even though it is more predictable and its variations are smaller. The decoder keeps constant the delay synchronization between the input of the encoder and its own output (continuous playing) by buffering the encoded or decoded pictures. This compensates for the variability of the encoding and decoding time and introduces the processing resynchronization delay.

The decoding time strongly depends on the decoder implementation: i.e., both on the availability of special purpose hardware, and on how early the decoder starts processing the bits it is receiving. For the purpose of this work we consider constant[9] the *decoding delay* D.

[8]This is likely because the peak rate of the encoder is of the order of few tens of Mb/s.

[9]If the decoding delay is not constant the decoder compensates its variations and (11) holds with $\max D$ instead of D.

The end-to-end delay is given by

$$\Delta_{VBR}^{Ded} = C_M + P + D + P_d \tag{11}$$

where C_M is the maximum *coding delay*; when a frame takes less than C_M to be encoded, it is delayed by the remaining time in the decoder buffer.

The faster the encoder and decoder, the smaller the end-to-end delay; in fact, since we are dealing with real-time video

Requirement 6 *A picture should be encoded within the video frame period,*

i.e., $C_M \leq T$. The maximum encoding delay is usually the video frame period corresponding to the highest frame rate the encoder supports. Thus, if a scene is captured at 30 frames per second and the propagation delay between sender and receiver is 20 ms (e.g., they are connected through a transoceanic link), the end-to-end delay is around 60 ms, i.e., acceptable to guarantee interactivity between the participants in a videoconference.

Motion estimation is the most time consuming part of the encoding process: E_M (and consequently the end-to-end delay) can be reduced by using only intra-encoding (I-frames). The yielded compression is lower and a larger number of bits is needed to encode a video sequence.

The system configuration discussed in this Section provides a lower bound on the end-to-end delay obtained in a videoconferencing system exploiting MPEG compression. Nevertheless, it is not efficient given the expensiveness of bandwidth: in fact, the capacity of the link connecting sender and receiver is only partially used. In the following Sections we study how the end-to-end delay is affected by the usage of multiplexing schemes to more efficiently exploit bandwidth of links.

4.3 Packet Switching with Time Driven Priority

A packet switched network with time driven priority can be exploited for transmission of encoded pictures, as shown in Figure 27. As soon as all the bits encoding a picture are produced by the encoder, they are inserted into a packets and sent at the full speed of the ingress link, as shown in Figure 28. The end-to-end delay of the system is given by

$$\Delta_{VBR}^{TDP} = C_M + L \cdot T_f + D + P_d \tag{12}$$

where L is the number of TFs a packet takes to travel from sender to receiver and P_d is the presentation delay. For the sake of simplicity, in this work we consider that the processing required for packetization does not introduce delay. Expression (12) is the actual end-to-end delay only if the nodes on the path from sender to receiver are able to perform RISC-like forwarding of packets (see Section 3.3.1).

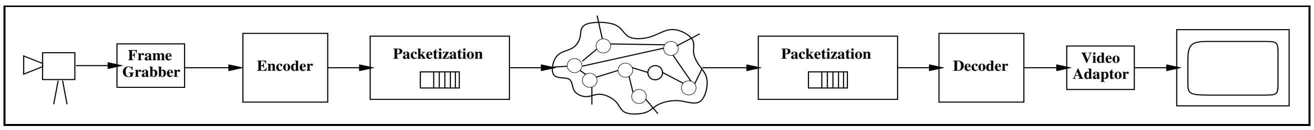

Figure 27: Videoconferencing System Configuration with Packet Switched Network between Sender and Receiver.

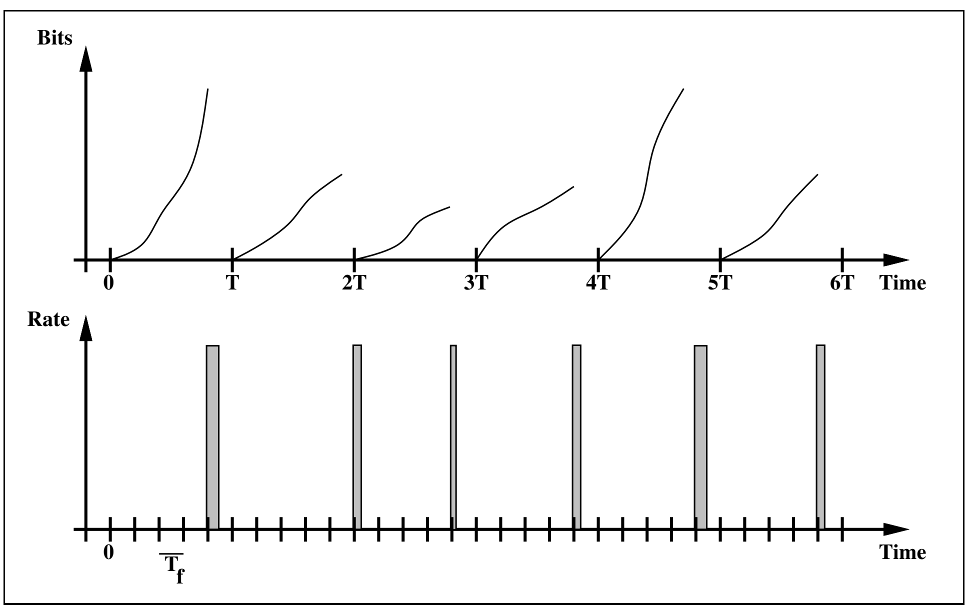

Figure 28: Time Driven Priority without Reservation.

To guarantee the fixed bound $L \cdot T_f$ on the 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 time driven priority, 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 20, the number of bits encoding a picture depends on the type of coding used for the corresponding picture, and the spatial and possibly temporal redundancy present in the image. Since the amount of redundancy depends on the scene being encoded, picture dimension is not known in advance and resource reservation cannot be accurate. If during a TF a user sends more bits than the amount reserved to it, the network does not provide any guarantee on the delivery of the excess bits. 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. 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.

In the following of this Section we show 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. Then, we show how the encoding process can be tuned in order to control picture dimension so that the videoconferencing system never uses more bandwidth than the amount allocated and exploits as much of it as possible. Lastly, we analyze how the synchronization between encoder and network interface, and the scheduling (i.e., the choice of the TFs to be reserved) impact 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 (12).

4.3.1 Choosing a Bound on Picture Dimension

As explained in Section 4.1.2, 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

$$B^{tdp} = \frac{F^I + (N-1) \cdot F^P}{N \cdot T} \tag{13}$$

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 in Section 4.1.3, the relative dimension of I-frames and P-frames yielded by a natural MPEG encode depends on the amount of motion in the scene. The *picture ratio*

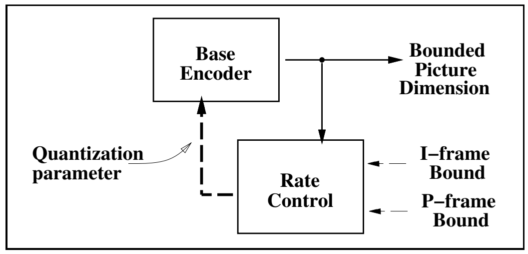

Figure 29: MPEG Encoder for Controlling Dimension of Frames.

$$\alpha = \frac{F^I}{F^P} \tag{14}$$

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 application, scenes are likely to be slow.

Combining Equations (13) and (14), the amount of bits to be reserved to each frame can be expressed as function of the bandwidth to be allocated as

$$\begin{cases} F^P = \frac{B \cdot N \cdot T}{N+\alpha-1} \\ F^I = \alpha \cdot F^P \end{cases} \tag{15}$$

4.3.2 Controlling Dimension of Encoded Pictures

The reserved bandwidth is used more efficiently if the encoding process is controlled and the number of bits encoding each picture is kept below the amount of bits reserved in the TF during which the picture is sent. This amounts are upper bound picture dimension and can be used to devise a target dimension for each type of video frame. As shown in Figure 29, the encoder is extended with a *rate control function*. It tunes the parameters of the basic MPEG coding process so that the dimension of each picture is best fitted to the target associated to its type.

Among the parameters of the MPEG encoding process the quantization parameter is the most suited to this purpose. For example, Figure 22 shows that the dimension of both I-frames and P-frames is strongly dependent on the value of the global distortion level G (which is one of the components of the quantization parameter): the larger, the smaller the encoded pictures. Nevertheless, G is not suitable for our purpose of controlling picture dimension because its value must be chosen before starting the encoding of the picture. There exist proposals to predict the amount of bits an encoding process is going to produce based on the raw scene [22] or on the portion of stream already produced [12]. However, to get service guarantees from the network, the target must never be exceeded and an adaptive approach based on feedback from the output of the encoder must be exploited.

Other authors propose iterative approaches [28, 16] to determine a suitable value of G. This can lead to coding times not acceptable for real-time encoding as required in videoconferencing applications (Requirement 6). [5] proposes to exploit a rate-quantization model to choose G. 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.

Alternatively, the *MB level rate control* factor b_{mb} is used for calculating the quantization parameter for each MB as

$$Q_{mb} = G \cdot p_{mb} \cdot b_{mb}$$

b_{mb} can be changed on a MB by MB basis in order to adaptively tune the quantization according to the target and the feedback obtained by monitoring the stream being produced [9, 17].

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 [10] possibly taking into account the characteristics of the human visual system [26].

The foregoing 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 requires. The software encoder `dvdenc` has been exploited; it has a rate control function which implements the algorithm shown in Figure 30.

The function which calculates the MB level rate control factor b_{mb} is provided with a target F_t and a tolerance on it (in terms of maximum and minimum acceptable dimensions). b_{mb} is incremented (decremented) with respect to b_{mb-1} if the amount of bits F produced so far exceeds (is below) the expected number of bits. This is calculated according to the target F_t and the total number of MBs Mb. The increment (decrement) is proportional to the excess (lack) of bits and the tolerance on the target. The average of the MB level control factor over the whole picture is used to calculate the global distortion level for the next picture of the same type. The experimental data shown in Figures 31 36 are obtained using $G = 15$ for both the first I-frame and the first P-frame. This algorithm is not wise from the point of view of yielding uniform visual quality throughout pictures. Nevertheless, the objective here is to show that the dimension of frames can be kept below (and very close to) a predefined bound, not to devise an optimal method to control the dimension of frames.

Figures 31 and 32 plot the picture dimension obtained encoding four different scenes with the shown bounds (F_M in the algorithm in Figure 30): the bounds are never exceeded. The bounds are determined according to Equation (15) assuming three bandwidth amounts and picture ratios (one in each row of the Figures).

As shown by Figures 33 and 34, the average quantization parameter has high variation for two complementary reasons:

```
according to the type of the picture to be encoded, define:
    a target dimension F_t (e.g., 5 % below reserved number of bits)
    a maximum dimension F_M (reserved number of bits)
    a minimum dimension F_m (e.g., 10 % below reserved number of bits)
b_1 = 1
choose G^{I|P} according to the type (I-frame or P-frame) of coding
F = number of bits encoding MB 1 with quantization parameter Q_1 = p_1 · G^{I|P} · b_1
for mb = 2 to total number of MBs Mb
    if F > (F_t/Mb) · mb then
        if F − (F_t/Mb) · mb > F_M − F_t then
            b_mb = MAXb
        else
            b_mb = decr(F, mb, F_M, F_t, Mb)
    else if F < (F_t/Mb) · mb then
        if (F_t/Mb) · mb − F > F_t − F_m then
            b_mb = MINb
        else
            b_mb = incr(F, mb, F_m, F_t, Mb)
    else
        b_mb = b_{mb−1}
    Q_mb = p_mb · G^{I|P} · b_mb
    F = F + number of bits encoding MB mb with Q_mb
G^{I|P} = (G^{I|P}/Mb) · Σ_{mb=1}^{Mb} b_mb according to the type of the encoded picture
```

Figure 30: Video Frame Dimension Control Algorithm.

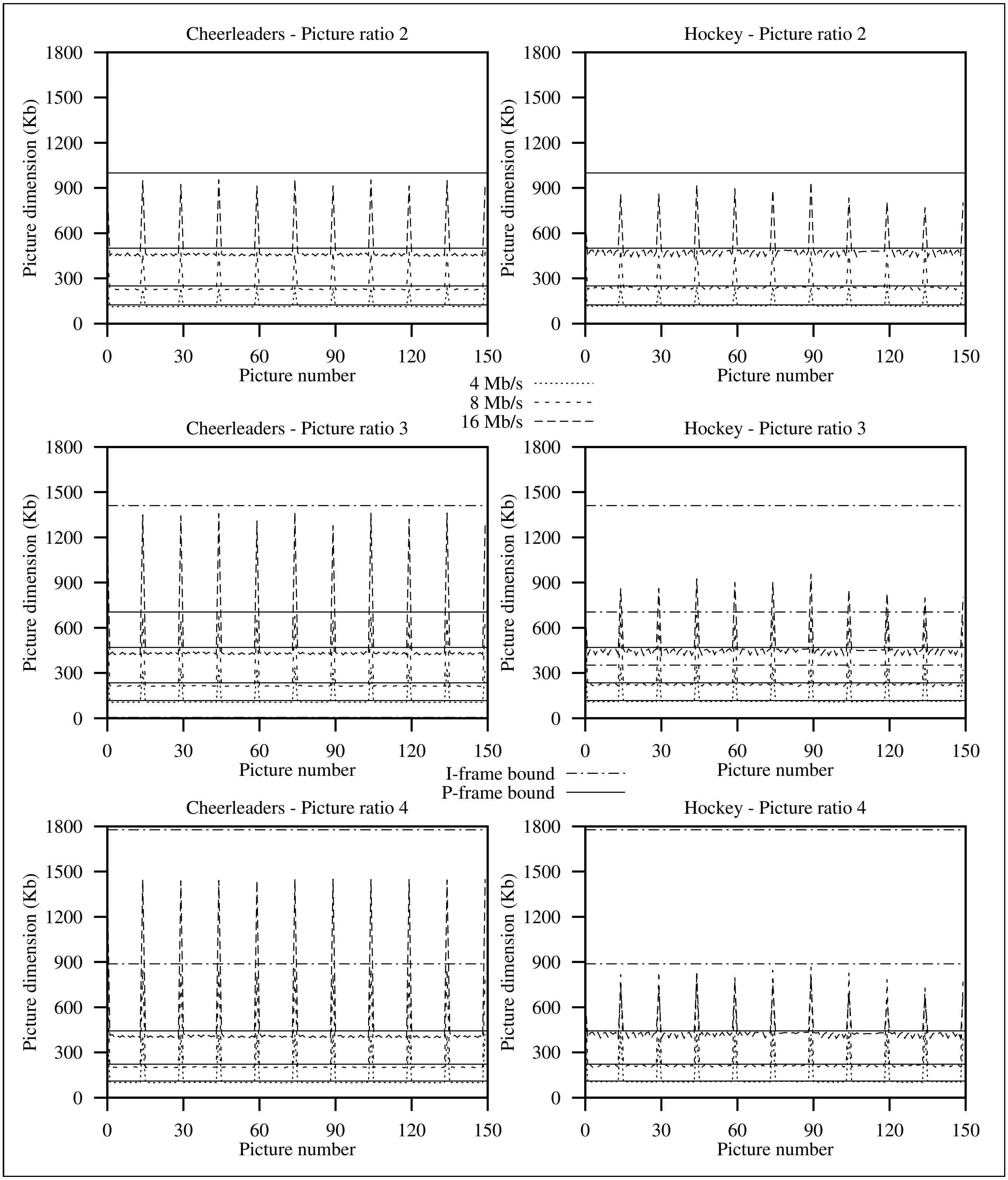

Figure 31: Dimension of Pictures in Controlled MPEG Encoding.

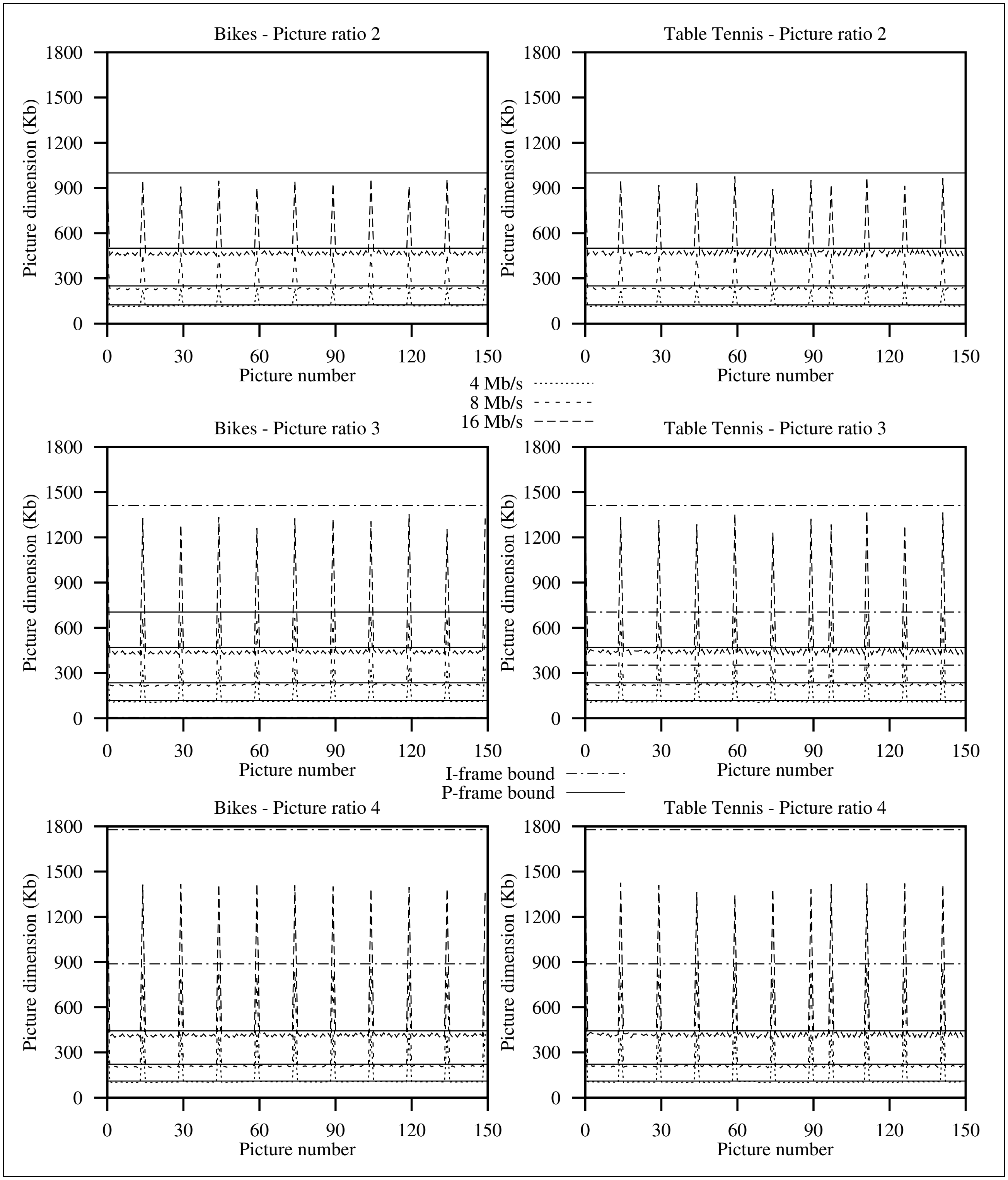

Figure 32: Dimension of Pictures in Controlled MPEG Encoding.

1. The variation of picture dimension is limited by the parameters provided to the rate control function in the experiments: the target (F_t in the algorithm shown in Figure 30) has been set 5% below the upper bound (F_M) and the lower bound of picture dimension (F_m) 10% smaller than the target.

2. The rate control function does not have the objective of delivering uniform quality throughout the scene.

As a result the quality of pictures is not uniform throughout each scene, as shown by the SNR plotted in Figures 35 and 36. The rate function control is not designed to keep the quality uniform inside pictures as well. As a consequence, the MB level rate control factor has high variability within each frame.

The scenes used in these experiments have pictures more detailed and a higher amount of motion than typical videoconferencing head and shoulder scenes. The same simple control function would deliver more uniform quality when dealing with with typical videoconferencing scenes. Moreover, enhancing the rate control function to reduce the quality variation in this scenario is not hard.

Having the rate control function bounding picture dimension, bandwidth is efficiently allocated by reserving fixed amount of bits for the transmission of pictures of the same type. The amount of bits is calculated using Equation (15). In the following we analyze how the choice of the TFs in which the bits are reserved affects the end-to-end delay; the conditions under which it can be reduced to Expression (12) are also presented.

4.3.3 Synchronization between Encoder and Network

save Scheduling, i.e., the choice of the TFs to be reserved to a videoconference call, is simplified by considering the nature of the application generating traffic. According to Requirement 6, videoconferencing systems must exploit real-time encoders that guarantee the delivery of a picture every T seconds.

To better understand the implications of scheduling on the end-to-end delay, we first consider that only intra-frame coding is performed. Under this assumption, resource reservation is performed as described in Section 3.3 for transmission of raw video. The time cycle duration must be an integer multiple M of the video frame period; for the sake of simplicity, we now consider $M = 1$ keeping in mind that the what is described in Section 3.3.3 applies to the general case. In each time cycle, F^I bits are allocated during one TF.

The bits encoding a picture are buffered from when they are produced to when the TF reserved for their transmission is scheduled. This buffering can take place, for example, in the packetization function. Temporarily assuming that encoding each picture takes the maximum coding time C_M, each picture spends in the packetization function buffer the application synchronization component of the network shaping delay $S_n^{AS} \in [0, T]$, as shown in Figure 37. As explained in Section 3.3.2, if the clock driving the capture card (and hence the pace at which

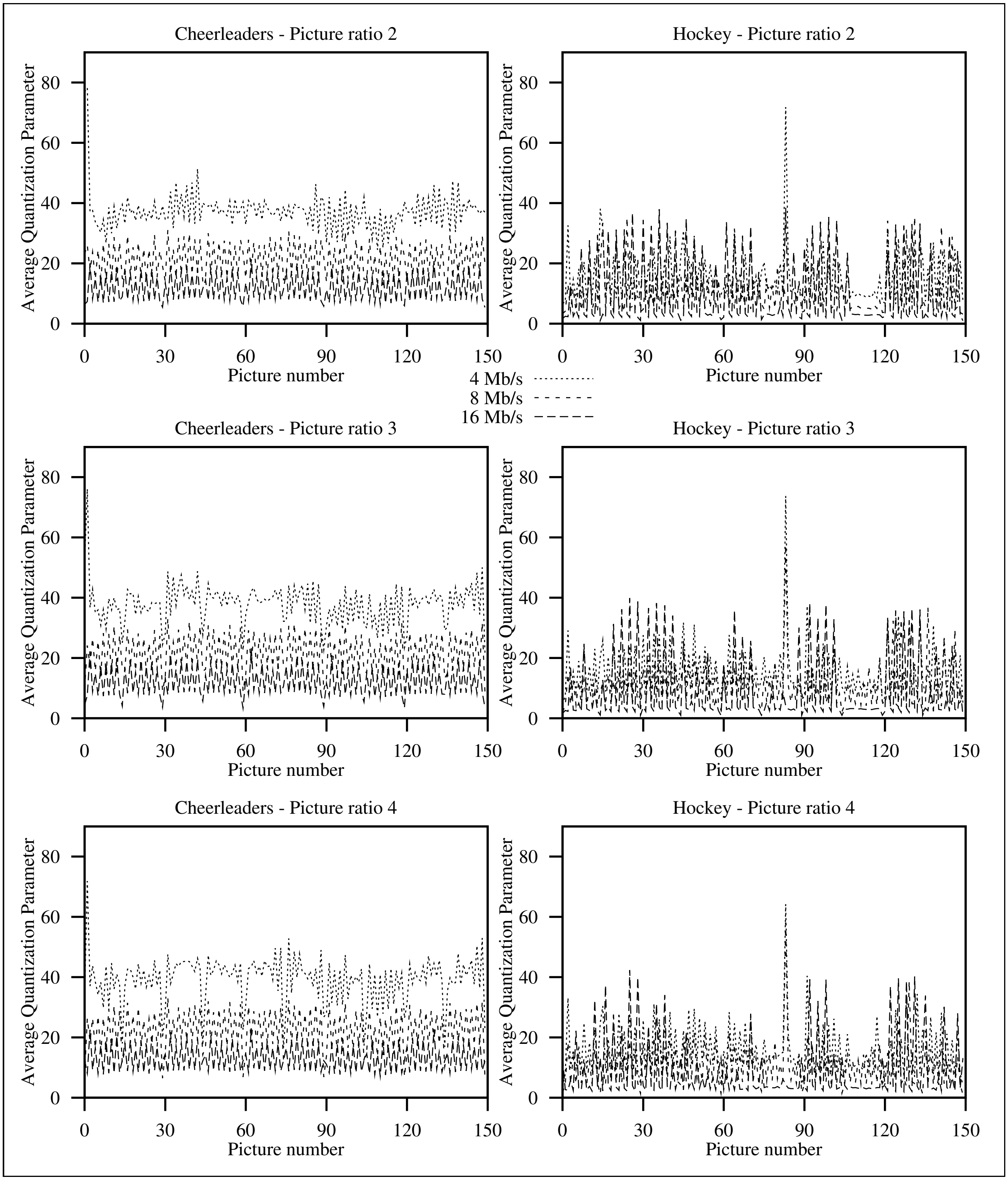

Figure 33: Average Quantization Parameter in Controlled MPEG Encoding.

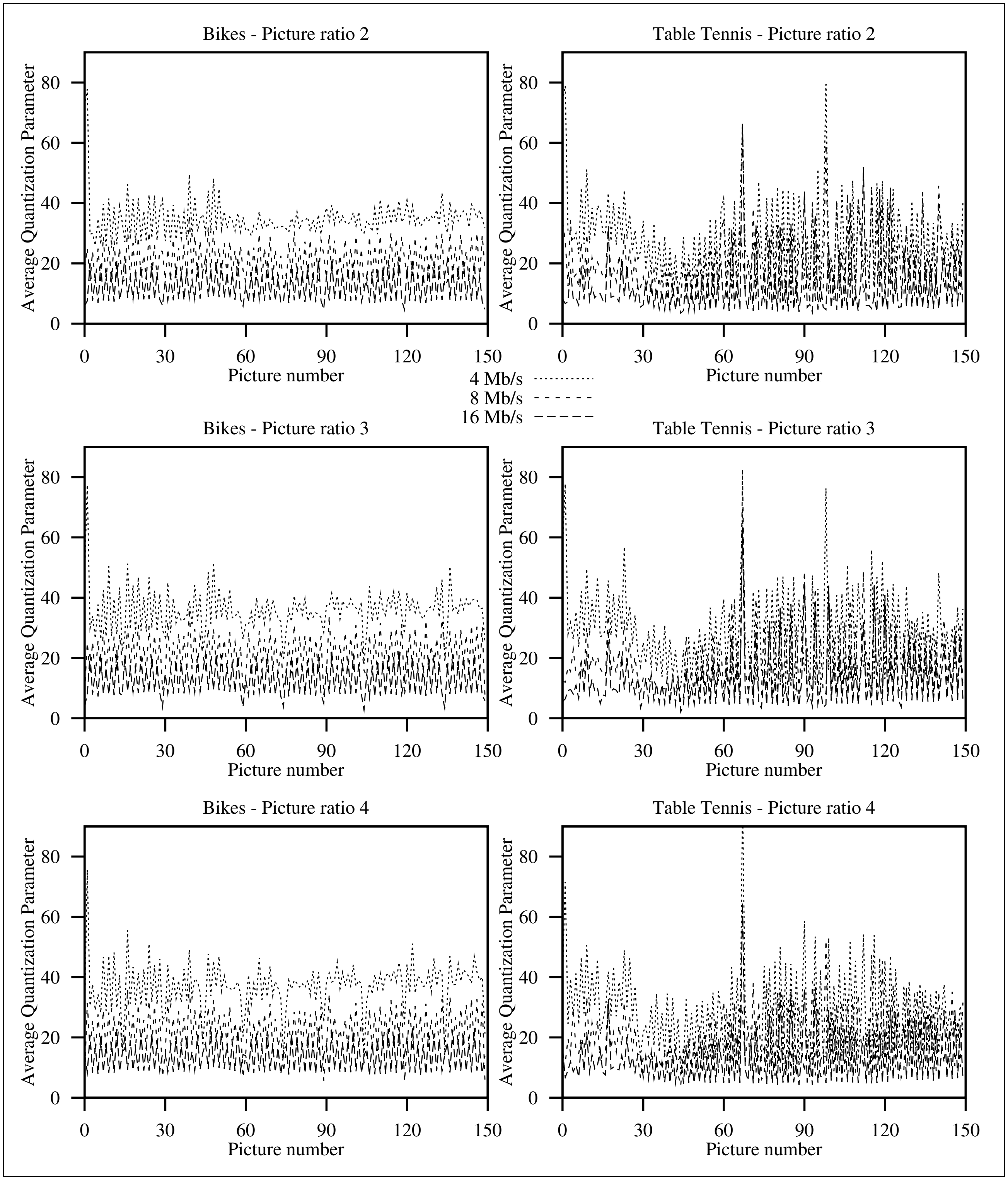

Figure 34: Average Quantization Parameter in Controlled MPEG Encoding.

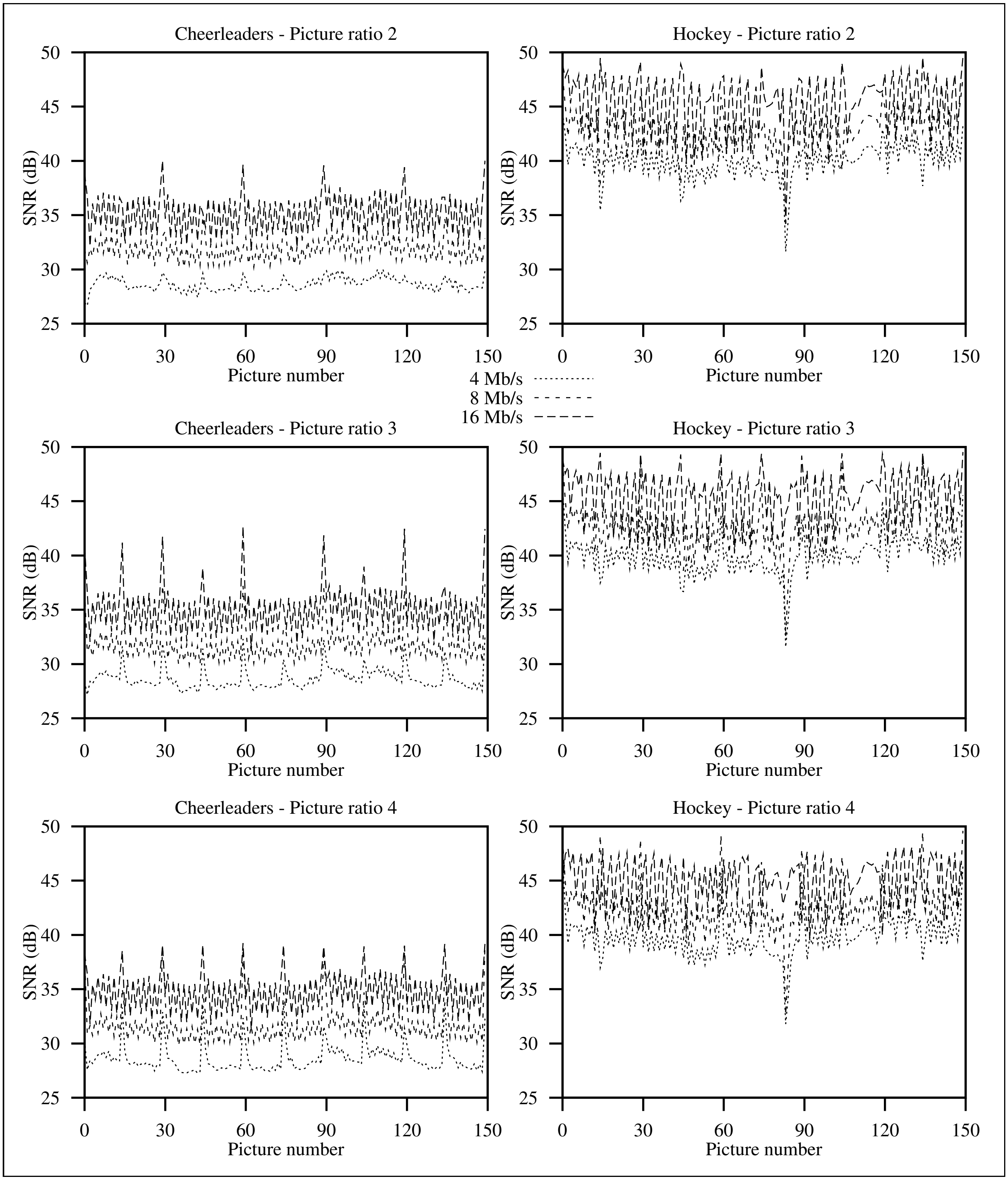

Figure 35: SNR in Controlled MPEG Encoding.

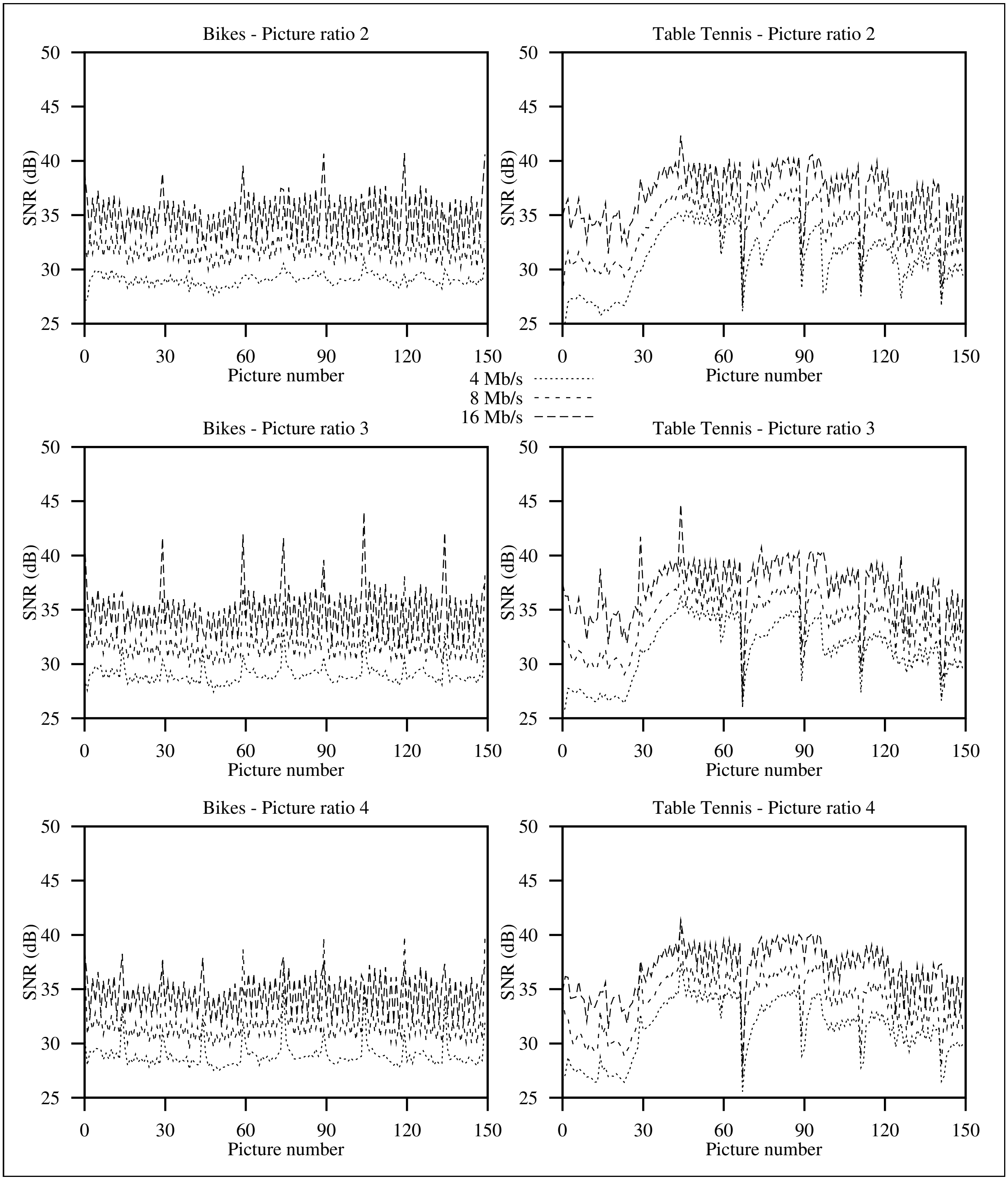

Figure 36: SNR in Controlled MPEG Encoding.

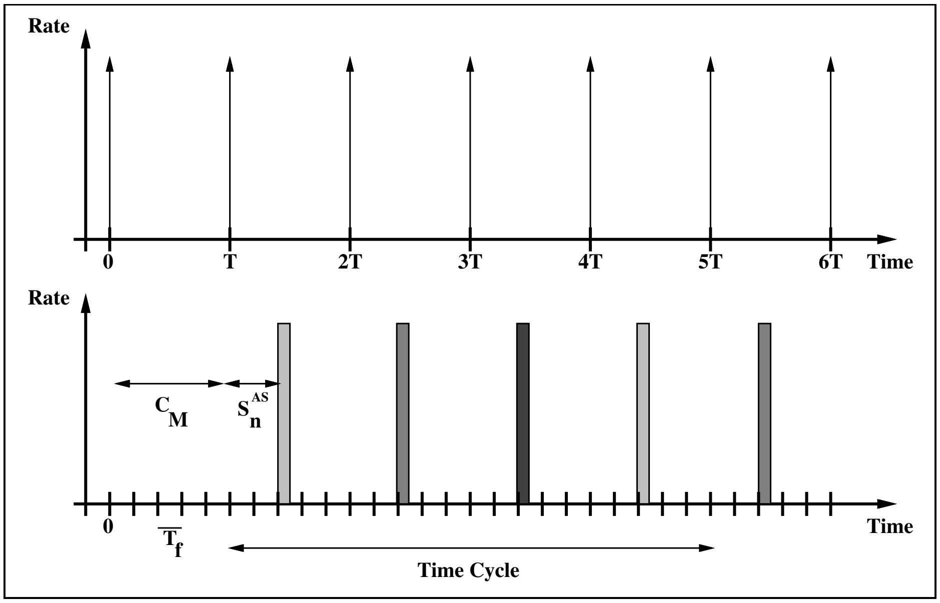

Figure 37: I-frame Only Coding and Bandwidth Reservation with Time Driven Priority

encoded pictures exit the encoder) is not synchronized with the network interface, S_n^{AS} slowly varies in the given interval. If the two capture card and network interface are synchronized, S_n^{AS} is constant and the time in which pictures are captured can be chosen so that $S_n^{AS} = 0$. Otherwise, the end-to-end delay is given by

$$\Delta_{VBR-I}^{TDP} = C_M + S_n^{AS} + L \cdot T_f + D + P_d \tag{16}$$

If the encoding time of a picture is less than C_M, it is buffered by the packetization function longer than S_n^{AS}. This does not affect the end-to-end delay because if the picture was sent as soon as encoded, it would be delayed by the decoder to compensate the processing delay variation. Thus, this configuration allows to simplify the receiver that does not need to compensate network jitter and processing delay variation.

4.3.4 Complex Scheduling

When both intra-frame and predictive coding are exploited, two different bounds are imposed for the two different types of frames: 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. Figure 38 shows a sample reservation with $M = 1$ and $N = 4$; the upper diagram shows the amount of bits generated to encode each picture assuming that they are instantly available

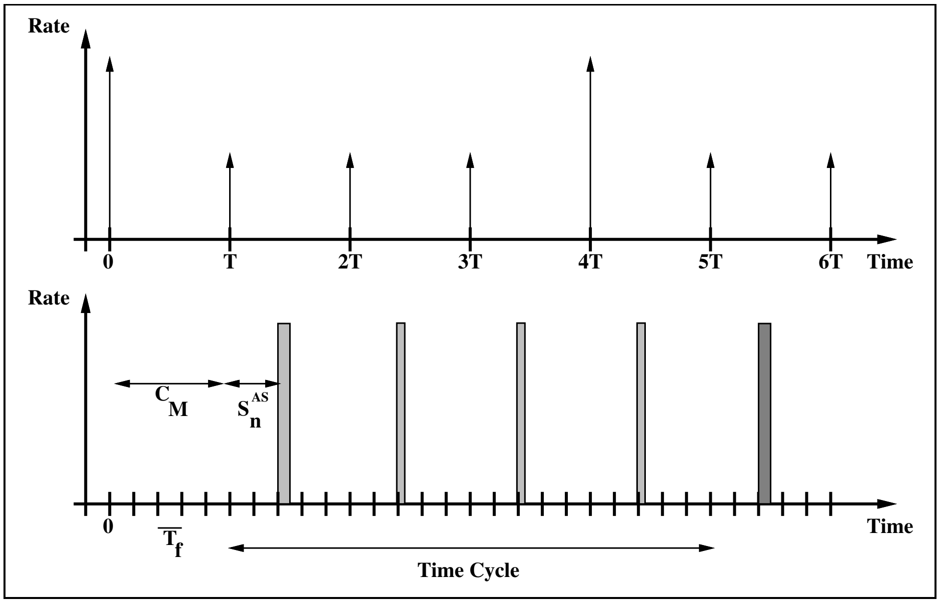

Figure 38: Time Driven Priority and Complex Scheduling.

after the maximum encoding delay C_M from the capture[10]. The lower diagram shows the amount of bits reserved in the TFs: F^I in one TF and F^P in the following $N-1$ TFs. The choice of the TFs to be reserved on each link on the path between sender and receiver is called *complex scheduling.* 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 videoconference call.

Assuming uniform distribution within the time cycle of the reserved TFs, the end-to-end delay is given by (16) as in the case in which only intra-frame coding is exploited. Nevertheless, in this case the application synchronization component of the network shaping delay has a different variation interval; for $M=1$, $S_n^{AS} \in [0, N \cdot T]$. A network shaping delay of the order of the GOP period is definitely not acceptable especially in videoconferencing applications where, due to the slow motion in the video sequence, it is convenient to use a long GOP (large N). Thus, synchronization between video encoder and network interface is essential to keep the end-to-end delay below the 100 ms bound and the buffer of the packetization function small.

Assumption 3 *The capture card (as well as the encoder) is synchronized with the network interface.*

If the TFs reserved to a videoconference call are not uniformly distributed within the time

[10]It has been already shown in the case of intra-frame only coding that this assumption does not constitute a limit of the system, but instead allows the decoder to be simplified.

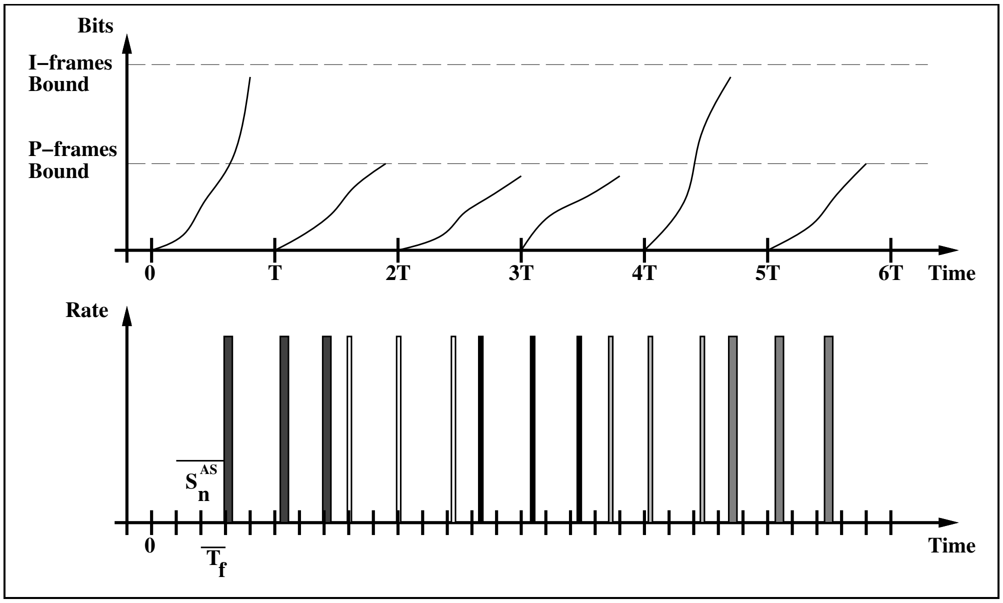

Figure 39: Time Driven Priority and Complex Scheduling.

cycle the interarrival time between video frames is not constant. The receiver compensate to this variability by buffering video frames received before schedule and delaying the whole video stream. Since this delay is due to the adaptation of the transmission time to the characteristics of the network, we categorize this delay as a *scheduling component* of the network shaping delay S_n^{Sched}. If Assumption 3 holds, the end-to-end delay can be written as

$$\Delta_{VBR}^{TDP-CxSc} = C_M + S_n^{Sched} + L \cdot T_f + D + P_d$$

where $S_n^{Sched} \in [0, N \cdot T]$: it is fixed when the videoconference call is placed and is kept small by a wise scheduling in the network.

4.3.5 Reducing Decoding Time

The decoding time D can be reduced if the decoder does not have to wait the whole picture before starting processing it. This can be obtained by inserting encoded video frames in smaller packets that are sent as soon as the corresponding bits have been produced by the encoder. This does not affect any other component of the end-to-end delay given by (16).

It is thus better to allocate more than one TF per picture and send a packet in each of them (Figure 39), than to allocate a single TF during which a whole encoded picture is sent (Figure 38). 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 reserved TF is scheduled on the sender link, 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 critical not just because the allocated bandwidth is underutilized (it can be exploited by best effort traffic), but because, 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 all the TFs, the remaining reserved TFs will not have enough capacity to carry the other bits that has to be sent in the scheduled 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), the benefit of using more TFs to transmit a picture is reduced, i.e., the end-to-end delay is increased.

The software encoder that has been used in this work shows the timing plotted in Figure 40 and Figure 41 when encoding frames with bounded dimension. The curves show that the production rate is quite constant, i.e., the position of TFs can be estimated with a simple linear model which introduces some safety margin (by shifting the TFs towards the end of the video frame period). Figures 40 and 41 show that a safety margin of few ms yields low probability of encoder underflow. Moreover, a videoconference scene has usually more uniform image complexity and motion, than the scenes used in these experiments, thus showing a more regular production rate.

If the instantaneous amount of bits produced by the encoder is below the amount predicted by the model, the bits that cannot be fit in the TFs reserved for the picture can be sent as best effort; if the network is congested, they do not timely get to the receiver and the visual quality of the reconstructed scene temporarily degrades.

The number of the reserved TFs per picture should be kept as high as possible because the smaller the transmission unit, the earlier the decoder can start decoding. Nevertheless, the higher the number of reserved TFs, the higher the probability of encoder underflow (the model averages the real production rate on a shorter interval). Moreover, even though in this work packet header overhead is being neglected, it has to be taken into account when choosing the dimension of the transmission unit.

The amount of bits to be reserved in each TF must be chosen finding a trade-off between the reduction in the decoding delay, on one hand, and the possibility of encoder underflow and the transmission overhead, on the other hand.

4.4 Asynchronous Packet Switching

A videoconferencing system can be constructed over an asynchronous packet switched network. As soon as the encoder produces enough bits to assemble a packet of dimension P_s, the packet is sent into the network where it is experience a variable queueing delay. As shown in Figure 42, the receiver must exploit a replay buffer to compensate the network delay variation. The end to end delay is thus given by

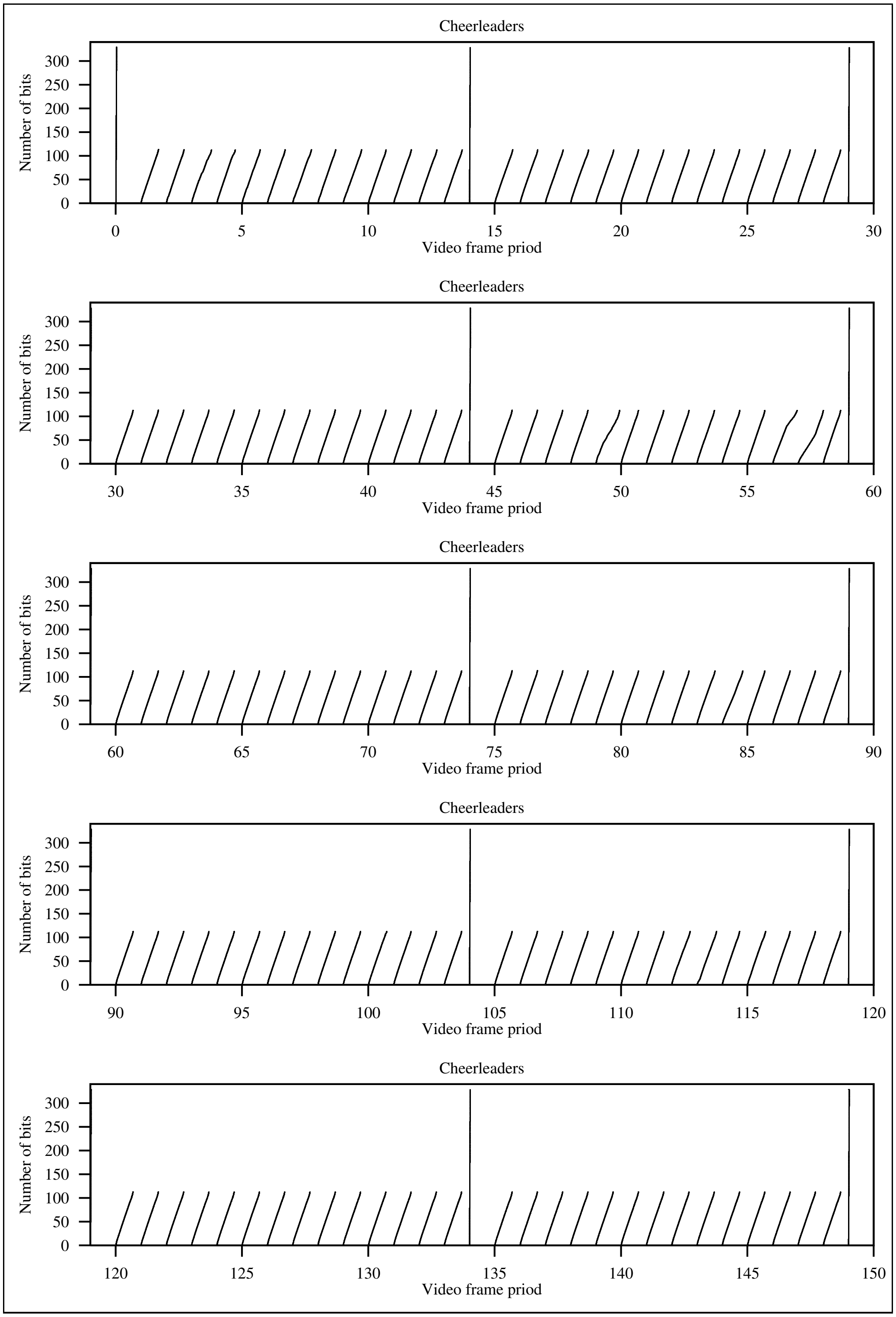

Figure 40: Production of Bits when Encoding the "Cheerleaders" Sequence.

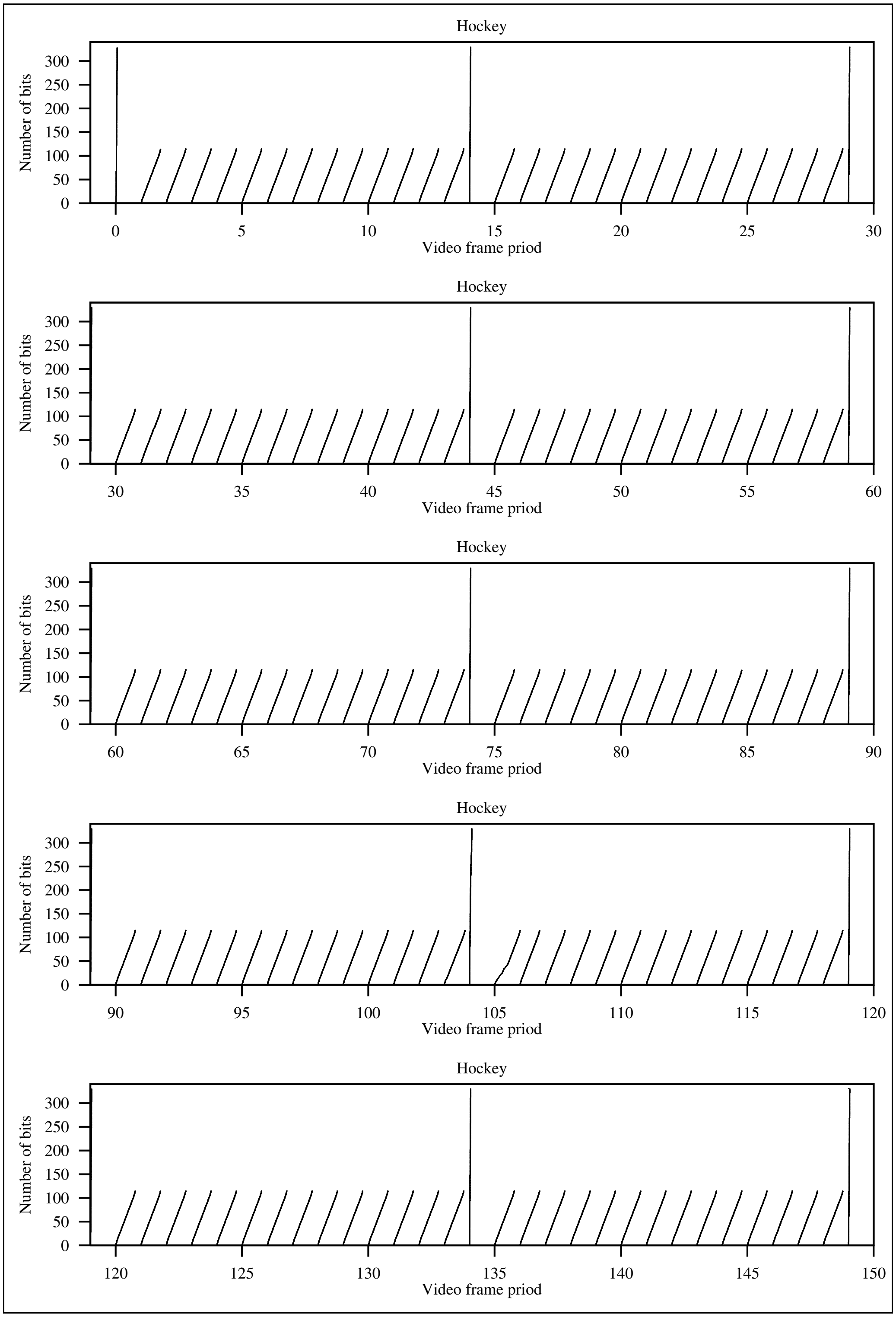

Figure 41: Production of Bits when Encoding the "Hockey" Sequence.

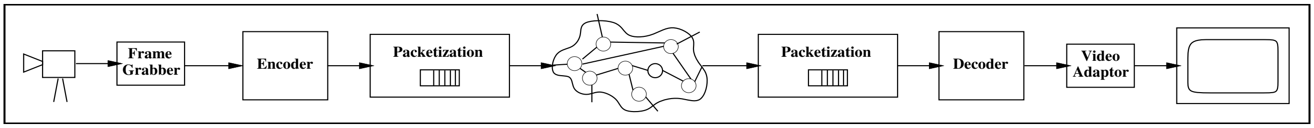

Figure 42: Videoconferencing System Configuration with Packet Switched Network between Sender and Receiver.

$$\Delta_{VBR}^{Async} = C_M + \frac{P_s}{C} + P + Q_M + E_r + D + P_d$$

where C_M is the maximum time required to encode a picture, Q_M is the maximum queueing delay, $E_r \in [0, \Delta Q]$ is the excess resynchronization delay introduced by the replay buffer, and D is the decoding delay. Using small packets and a decoder which starts decoding video frames as soon as data are received, can reduce D. Anyway, the end-to-end delay is dominated by C_M and Q_M.

4.4.1 Traffic Shaping at Network Boundaries

Resources are reserved in the network in order to bound the queueing delay. As discussed in Section 3.4.2, resource reservation is more efficient if traffic shaping is performed at the boundaries of the network, even though it introduces a network shaping delay. 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 on the service (i.e., the bound Q_M on the queueing delay) 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 a replay buffer which introduces a network resynchronization delay. Thus, each packet experiences a network shaping delay S_n^{TS} partly in the traffic shaper and partly in the replay buffer. In Section 3.4.2, we have discussed the constraints on the choice of the leaky bucket parameters and the impact on the end-to-end delay in the case of transmission of raw video. Since in the case of VBR MPEG encoded video the same task is harder and it is not the goal of this work, we do not analyze S_n^{TS} in more detail. The end-to-end delay of the videoconferencing system is given by

$$\Delta_{VBR}^{Async-TS} = C_M + S_n^{TS} + \frac{P_s}{C} + P + Q_M + E_r + D + P_d$$

4.4.2 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 [21]. For example, if a leaky bucket is exploited to shape the traffic, the token generation rate B and token pool size A determine the burstiness and average rate 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 with "best effort" service. 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 [6], 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 [24, 23]. Due to the unpredictable output of MPEG encoders, this approach does not guarantee against packets not compliant with the traffic description.

Alternatively, the 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) [20, 4], as shown in Figure 43. If this requires to significantly degrade the visual quality of pictures, the resource allocation is renegotiated according to a rate-quantization model that is tuned as the encoding progresses.

Hierarchical or *layered encoding* [19] provides another means to adapt the encoded video stream to the service provided by the network. The bits encoding each frame are divided into two separate layers, the high priority layer - which contains the most critical data according to the sensitivity of the human visual system - and the low priority layer. The former must comply to the traffic description because it needs service guarantees, while the latter is sent as best effort. Even though the low priority component gets lost, the video sequence can be reconstructed on the receiver with acceptable quality. The encoded stream can be partitioned dynamically according to feedback information from the traffic shaper (e.g., the fullness of the token bucket) [18]. The fraction of data sent at the high priority layer to better fit the corresponding stream better to the traffic description.

Hierarchical encoding can get the best from a packet switched network by using all the available bandwidth when the network is not overloaded, and getting reasonable visual quality when it is congested. On the countrary, the approach based on controlling the rate of the MPEG encoder varies the quality of images with the objective of fitting the encoded video

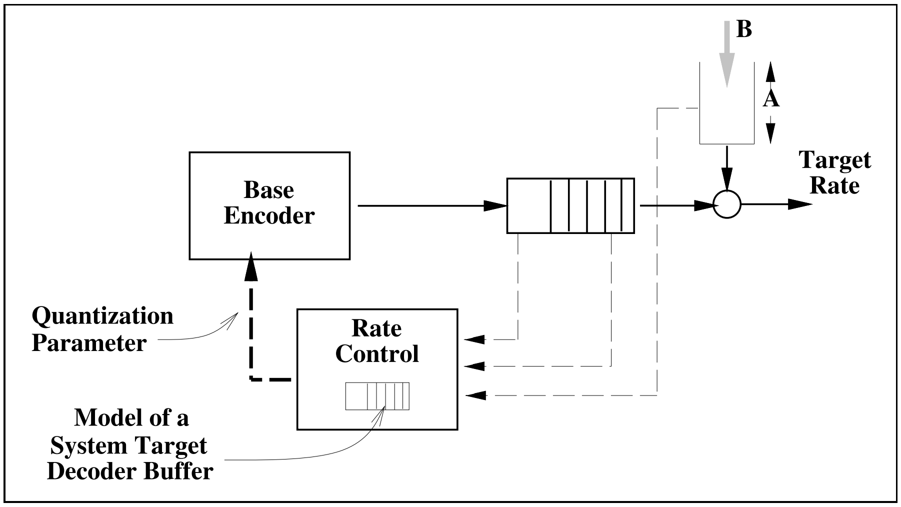

Figure 43: MPEG Encoder Controlled using Feedback from a Traffic Shaper.

stream into the traffic description. Thus, visual quality can sometimes be decreased even though the network could possibly support a high rate (i.e., higher quality).

Nevertheless, the hierarchical encoding approach is not efficient because the low priority stream puts a harmful burden on the network when it is overloaded. During congestion, sending low priority pakets does not yield any advantage since they do not get to the destination. Even worse, low priority packets waste resources until they are discarded. The waste is even larger if packets make their way to the destination and there they are discarded due to excessive delay[11].

4.5 Circuit Switching

A circuit switched network can be used for the transmission of VBR MPEG encoded video. If the videoconference call is allocated a circuit with bandwidth larger than the instantaneous rate of the encoder and the end-to-end delay is

$$\Delta_{VBR}^{CS} = C_M + Sw + P + D + P_d,$$

where Sw is the switching delay and P the propagation delay in the network. This expression is similar to (11) given in Section 4.2 for the end-to-end delay of a videoconferencing system that uses VBR MPEG encoding and a direct link between sender and receiver. When a circuit switched network is exploited the switching delay has to be added; moreover, not all the capacity C of links is dedicated to the videoconference call.

[11]Take into account that the delay experienced by best effort traffic has a large bound and high variability.

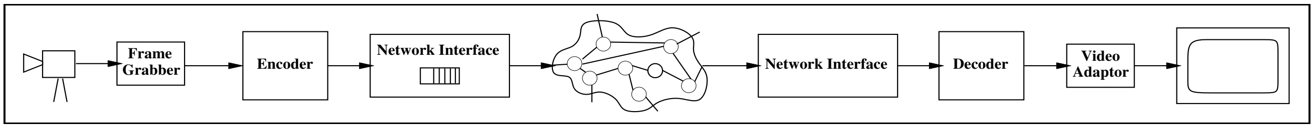

Figure 44: Videoconferencing System Transmitting VBR MPEG video over a Circuit Switched Network.

Since the circuit has been dimensioned to support the burstiness on the VBR MPEG encoder, a large part of its bandwidth is wasted. The bandwidth of the circuit can be reduced below the peak rate of the MPEG stream if a buffer is used between encoder and network, as shown in Figure 44. It smoothes the rate of the encoder to fit it in the circuit bandwidth at the expenses of a network shaping delay S_n^{CS} added to the end-to-end delay. What is discussed in Section 4.4.1 about the network shaping delay S_n^{TS} introduced by a traffic shaper when using packet switching, applies also to S_n^{CS}. In fact, the shaping performed by the smoothing buffer and the circuit switched network is equivalent to the one due to a traffic shaper with average bandwidth B and maximum burstiness 1 bit (e.g., a leaky bucket with token generation rate B and token pool size $A = 1$). Pictures are buffered in the sender for a time that depends on the rate of the MPEG encoder and the bandwidth of the circuit. A replay buffer on the receiver side compensates the variation of this time thus yielding the fixed contribution S_n^{CS} to the end-to-end delay which is given by

$$\Delta_{VBR}^{CS-TS} = C_M + S_n^{CS} + P + Sw + D + P_d$$

The buffer between encoder and network has the same purpose as the buffer at the ingress of a leaky bucket, which has been described in Section 3.4.2; when it overflows part of the encoded stream is discarded. Dimensioning the buffer and the circuit bandwidth so that discarding of packets is avoided is not possible because the bit rate produced by the MPEG encoder is not known in advance. Thus, the bit generation rate of the encoder must be monitored and regulated by a rate control function which exploits the buffer status as feedback information. The system composed by the basic MPEG encoder, the buffer, and the rate control function is actually a CBR MPEG encoder which is the subject of the next Section.

5 Transmission of CBR MPEG Video

The basic MPEG coding process produces a variable bit rate stream. As has been shown in the previous Section, to prevent loss, the natural variability of the bit generation rate must be controlled and adapted to the transmission service. If the transmission service requires data to be sent at Constant Bit Rate (CBR), a CBR MPEG encoder is the choice for building the videoconferencing system.

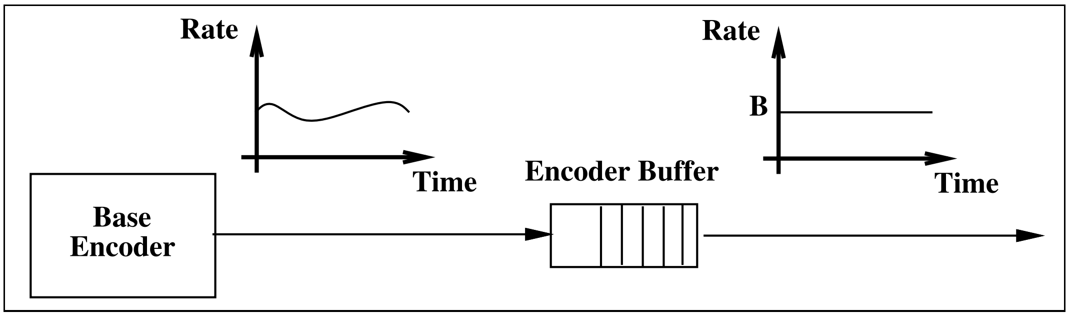

Figure 45: Basic MPEG Encoder and Encoder Buffer.

A CBR MPEG encoder/decoder system introduces a delay that is independent of the transmission scheme adopted and widely varies according to the parameters of the encoder. In the following, the time needed to encode and transmit CBR MPEG video is studied for further steps with the aim of highlighting the different components and the system parameters that influence them. First, the MPEG encoding/decoding system is analyzed from the point of view of delay when only intra-frame coding is used; the implications on the delay when using also predictive coding are considered next. Lastly, the transmission with different schemes is analyzed in order to devise the end-to-end delay in various configurations.

5.1 Intra-frame Coding Only

The case in which only spatial redundancy is eliminated (i.e., the encoded video sequence is composed only of I-frames) is first considered; basically, this is the compression process recommended by Moving JPEG [7] [25] and thus the basic principles discussed in this Section also apply to a videoconferencing system which exploit JPEG to compress video frames.

5.1.1 Coding Shaping Delay

A CBR MPEG encoder is built by smoothing the natural rate of a basic MPEG encoder into a constant *target rate* B. As shown in Figure 45, the smoothing is performed by the *encoder buffer*, which is filled at the natural bit rate and emptied at the target bit rate B. The encoder buffer delays each bit depending on the natural bit rate and the target bit rate. Also, bits are buffered in the decoder buffer until a whole picture is received and can be decode. A picture encoded with F bits takes a time F/B to exit the encoder buffer and enter the decoder one. This is the processing delay of the CBR encoder that depends on the picture dimension. Since it not constant, the decoder must compensate it to comply with Requirement 3 which impose that video frames are continuously displayed at a fixed rate. The composition of the processing delay of the encoder/decoder system and the processing resynchronization delay introduced by the decoder is a constant *coding shaping delay* S_c.

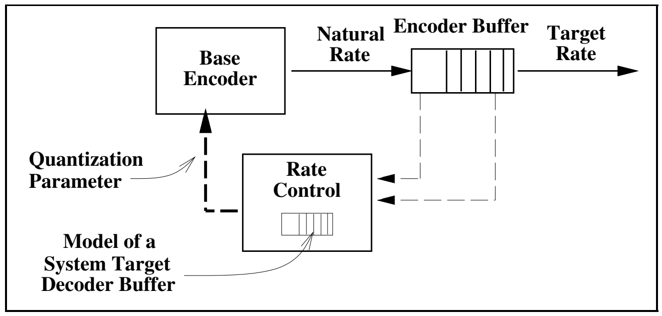

Figure 46: Basic Encoder and Rate Control Function.

Since bits exit the encoder buffer at rate B,

$$S_c \geq \frac{\max_{seq} F}{B}, \tag{17}$$

where $\max_{seq} F$ is the maximum number of bits used to encode a picture over the all sequence. (17) guarantees that in the time between when a picture is grabbed, to when it is scheduled for displaying, up to $\max_{seq} F$ bits can exit the encoder buffer at rate B. In order for the encoded video stream to be continuous (i.e., actually having a constant rate),

$$S_c \geq T, \tag{18}$$

otherwise there would be a time interval between two subsequent frames during which no bits are produced. (18) shows that the end-to-end delay of a videoconferencing system exploiting a CBR MPEG encoder is larger than the video frame period.

Even though the coding shaping delay is not related to the transmission over a network, it is analogous to the network shaping delay S_n^{TS} introduced by a traffic shaper cascaded to a VBR MPEG encoder (see Section 4.4.1). If a basic MPEG encoder freely fills a finite dimension buffer which is emptied according to a fixed traffic description, the buffer can possibly overflow. In Section 4.4.2 we surveyed some proposals to control the basic MPEG encoding based on feedforward or feedback information, in order to avoid a traffic shaper to discard packets. In the case of a CBR encoder, the rate at which the buffer is emptied undergoes to a very strict traffic description (being it constant). Thus, a rate control function must be introduced to control the basic MPEG encoder in order to avoid the encoder buffer to overflow or underflow.

5.1.2 Rate Control Function

Figure 46 shows the components of a CBR MPEG encoder. In principle the rate control function monitors the fullness of the encoder buffer and adjusts the quantization parameter to change the natural bit rate and avoid the buffer to underflow or overflow.

The compliance of an encoded video stream with the MPEG standard requires that feeding a system target decoder with the stream does not overflow or underflow its buffer, whose dimension is included in the MPEG stream [11]. The MPEG standard specifies that the system target decoder is fed at the constant rate indicated in the stream and it retrieves data from its buffer one *access unit* at a time. The access unit is a whole picture and all the bits encoding a video frame are removed when its decoding is started according to the decoding time specified in the MPEG stream. The buffer of the system target decoder underflows if a whole picture is not present in the buffer when its decoding is scheduled.

Thus, beside monitoring the encoder buffer, the rate control function should keep an updated model of the buffer of a system target decoder, as shown in Figure 46. It is filled at the target rate B, and emptied by one picture at the instant in which decoding is scheduled, as shown in Figure 47. The quantization parameter (and thus the natural bit rate) is tuned so that the stream compliance is granted.

Actually, if the encoder buffer has the same size of the system target decoder one and the time to encode a picture is assumed to be null, the fullness levels of the two buffers are strictly related. In particular the encoder buffer overflows (underflows) if and only if the system target decoder one underflows (overflows).

In practice, the rate control function can model *virtual buffer* which is updated according to the dimension of pictures and the target rate B. If the virtual buffer does not underflow or underflows, neither the encoder buffer nor the system target decoder buffer do. The MPEG standard provides guidelines on how to manage the virtual buffer and modify the quantization parameter [9]; anyway, each CBR MPEG encoder implementation can use its own approach [28, 26, 5, 23].

This the virtual buffer constrains the number of bits used to encode each picture. Since according to (17) the maximum number of bits used to encode a picture provides a lower bound on the coding shaping delay, we analyze in more detail the virtual buffer implementation of the software encoder `dvdenc` with the objective of devising its impact on the end-to-end delay.

The rate control function models a buffer, called Video Buffer Verifier (VBV), that has the dimension of the system target decoder buffer and is drained at the constant bit rate B. Before encoding a picture, the rate control function sets the maximum and minimum dimensions of the picture. The maximum dimension is set by conservatively assuming that it must not exceed the space available in the buffer[12]. The minimum dimension guarantees that the buffer is not emptied before the end of the video frame period. The VBV is then updated according to the actual number of bits produced. In the following Section, the lower and upper bounds on the dimension of pictures in an CBR MPEG stream are given in terms of the size of the VBV.

[12]This conforms to conservative assumption that even if all the bits were injected in the VBV instantly, it would not overflow.

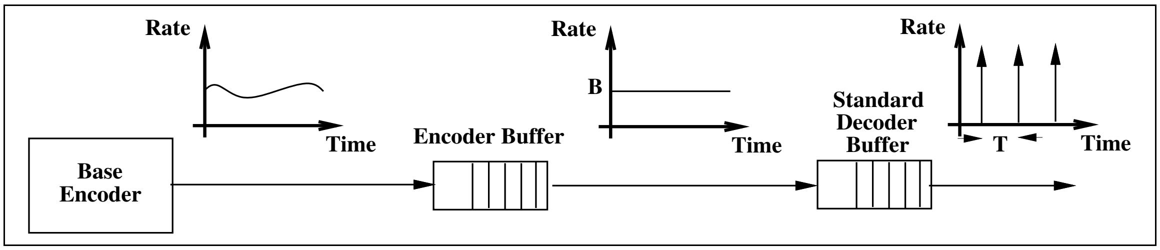

Figure 47: Standard Decoder Buffer Model Kept by the Control Function.

5.1.3 Dimension of Encoded Pictures

In the video frame period T, a CBR MPEG encoder working at the target rate B produces $B \cdot T$ bits. The rate control function can be designed in order to use exactly $B \cdot T$ bits to encode each picture. Even though this would guarantee a constant bit rate, the visual quality of images would not be uniform. Instead picture dimension is allowed to vary and is controlled according to the VBV.

Claim 1 *The dimension of the VBV is the upper bound on the dimension of pictures in an MPEG compliant stream:*

$$V_s \geq \max_{seq} F$$

where V_s is the dimension of the VBV and F is the dimension of a picture.

Proof The VBV must be larger than the largest encoded picture because it must contain a whole picture before it is started being decoded. Moreover, as the VBV overflow must be avoided to guarantee compliance with the MPEG standard, the rate control function controls the basic ncoder so that the number of bits used to encode a picture is not arger than the VBV dimension.

The lower bound on encoded picture dimension is also related to the VBV dimension. To show how this lower bound is determined, we first analyze how picture dimension affects the encoder buffer level. We assume that the encoder buffer and the VBV as the same dimension; this is not a restriction because if the VBV does not underflow, the encoder buffer never contains more bits than the VBV dimension.

In order to have enough bits in the encoder buffer so that it is emptied at the constant rate B during the time T between two pictures, a picture should be encoded with at least $B \cdot T$ bits. If some backlog is present in the buffer at the beginning of the video frame period, the number of bits used for encoding the picture can be smaller. Thus, the lower bound F_m

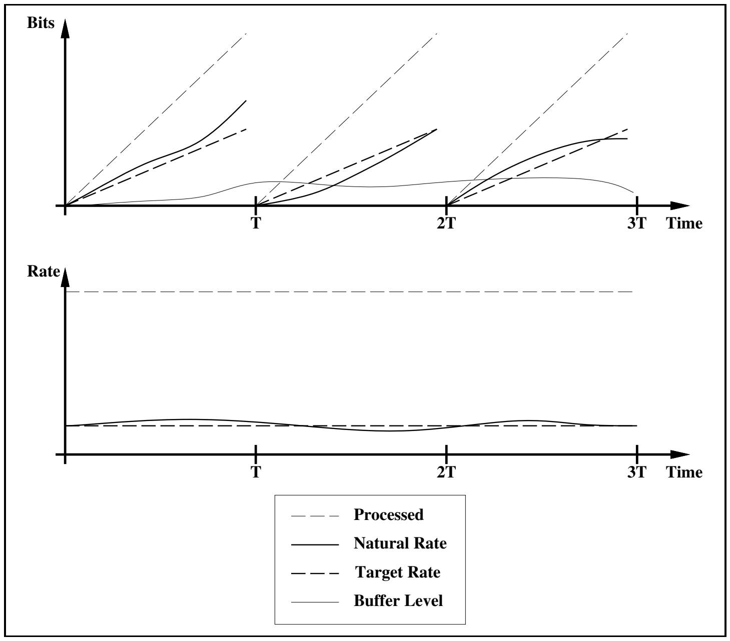

Figure 48: Bit Production Rate and Encoder Buffer Fullness.

on picture dimension is the minimum allowed picture dimension when the buffer contains the largest backlog.

The backlog is due to bits encoding the previous picture that has not been sent out yet, as exemplified in Figure 48. The upper diagram shows that the encoder processes the bits of each raw image during a single video frame period (thin dashed line). The diagram also plots the number of bits generated by the basic encoder versus time for each picture (thick continuous line); the total number of bits per picture varies for different pictures. The CBR encoder emits bits at constant rate B (as shown by the thick dashed line in the lower diagram), thus sending out the same amount of bits ($B \cdot T$) in each video frame period (last point of the thick dashed line in the upper diagram). If the dimension of a picture exceeds $B \cdot T$ (e.g., the first picture in Figure 48), excess bits are temporarily buffered in the encoder buffer (whose level is depicted by the thin continuous line) until they have been all sent out (after the end of the video frame period).

The largest backlog V_b remains in the buffer after a picture has filled it. Before starting encoding a picture the rate control function checks the VBV and allows the picture to be large enough to fill it. If the maximum allowed amount of bits is produced, the level of the VBV (and the encoder buffer) at the end of the video frame period is

$$V_b = V_s - B \cdot T$$

because during the video frame period $B \cdot T$ bits have been drained from the buffer. The rate control function determines the minimum allowed dimension of the next picture as the number of bits that must be inserted in the VBV so that it contains at least $B \cdot T$ bits, i.e., the amount of bits to be drained during the video frame period:

$$F_m = B \cdot T - V_b = 2 \cdot B \cdot T - V_s$$

This is the lower bound on the picture dimension.

In summary, the dimension of the VBV is the maximum number of bits usable to encode a picture. Moreover, the minimum number of bits usable to encode a picture is bounded by the VBV dimension, the video frame rate, and the target rate of the encoder. Since the natural dimension of MPEG encoded pictures is highly variable (see Section 4.1.3, the more the dimension of pictures is allowed to vary, the more uniform is their visual quality. The maximum possible variation of picture dimension is given by

$$\Delta F_M = V_s - F_m = 2 \cdot (V_s - B \cdot T) \tag{19}$$

The larger the VBV dimension, the larger the possible variation of dimension of encoded pictures.

The actual number of bits used to encode each picture is determined by the rate control function. The rate control function must use wisely the allowed variability in picture dimension is fundamental to keep the visual quality uniform throughout pictures and scenes. The task is made more difficult by the fact that, under Assumption 3 of continuous playing made in Section 2.2, the amount of bits used to encode a picture is constrained by the amounts used in the preceding ones. This is stated by the following Claim.

Claim 2 *If a picture has its maximum allowed dimension (i.e., it fills the VBV), the following picture must be smaller than $B \cdot T$.*

Proof When a picture is encoded with the maximum allowed number of bits, at the end of the video frame period in which it has been captured, $B \cdot T$ has been drained from the VBV. Being this the number of bits needed to fill the VBV, it provides the maximum allowed dimension for the following picture.

5.1.4 Startup Shaping Delay

The rate control function uses the VBV to guarantee that the MPEG stream complies with the MPEG standard and that the encoder buffer does not overflow or underflow. In this Section we show that, even if the dimension of a picture is larger than the minimum allowed by the VBV, the encoder buffer can underflow. We also describe how the CBR encoder copes with this situation.

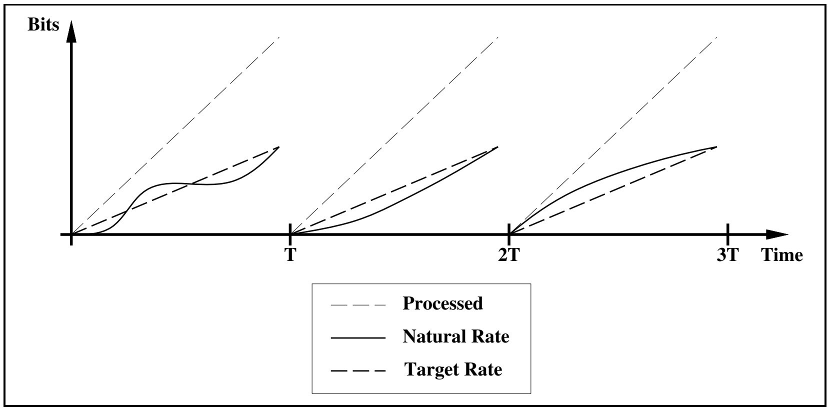

Figure 49: I-frames Only Encoding.

Figure 48 shows the timing associated with intra-frame coding. The thin dashed line in the upper diagram shows the amount of bits processed versus the time. This is larger than the amount of bits produced by the basic encoder (continuous line) because of the yielded compression; the punctual slope of the line is the instantaneous natural rate. The objective of the CBR encoder is to produce bits at the constant target rate B, as shown by the thick dashed line; the encoder buffer is used to smooth the variations of the natural bit rate, as exemplified by the lower diagram of Figure 48.

When the instantaneous natural bit rate is smaller than the target rate, the encoder buffer level lowers, as shown, for example, in the upper diagram of Figure 48 during the second video frame period by the thin continuous line. If the natural rate is smaller than the target rate when the buffer is empty, the encoder buffer underflows. This happens, for example, in the situation depicted in Figure 49: at the beginning of the encoding of a scene, the natural rate is lower than the target rate. This can be prevented by waiting a *startup shaping delay* when the encoding of a scene starts, before starting sending bits out of the encoder buffer. The startup shaping delay is obtained by fixing a fullness threshold in the buffer and waiting it to be crossed before the MPEG stream starts flowing out of the CBR encoder.

The threshold represents the tolerance of the system to a natural bit rate lower than the target rate. Since the natural rate is controlled by the rate control function, the threshold can be as low as desired, provided that the rate control is aggressive enough to increment the natural rate so that the buffer does not underflows. Nevertheless, the lower the threshold, the smaller the area of the picture over which the natural production rate can be smaller than the target rate, i.e., the area over which the rate control function must intervene. Thus, the visual quality of images is not uniform because the quantization parameter changes suddenly. On the other hand, the higher the threshold, the larger the startup shaping delay.

The choice of the tradeoff between long startup shaping delay and non-uniform visual quality (and aggressive control function) is left to the specific encoder implementation. The

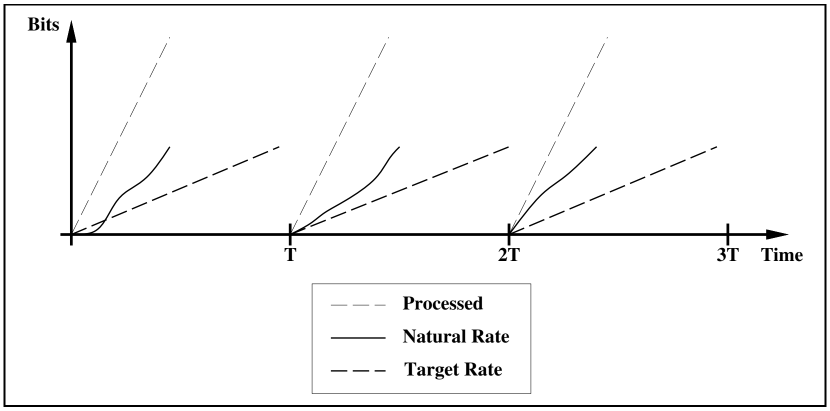

Figure 50: Fast I-frame Only Encoding.

chosen value of the startup shaping delay is not very critical because, if the encoder is powerful enough the target rate is much lower than the average natural rate, as shown in Figure 50. This means that the main aim of the encoder buffer is not really smoothing the variation of the natural rate, but lowering it. Thus, it is not necessary to introduce a large startup shaping delay to cope with actual bit production rate variation, i.e., the fraction of buffer to be filled before starting transmitting should not be large because the buffer is emptied much slower than it is filled on the average.

5.1.5 Shaping Delay Implementation

After having described which are the bounds on the dimension of pictures, we analyze their impact on the delay and how it is introduced by the CBR encoder/decoder system.

Pictures experience the startup shaping delay entirely in the encoder buffer. The encoder introduces the startup shaping delay when it starts encoding a scene: no bit exits from its buffer until the buffer level crosses a predefined threshold. In order to keep the startup shaping delay small, the rate control function must increase the natural rate if the threshold is not crossed within a predefined amount of time.

The coding shaping delay is experienced by pictures partly in the encoder buffer and partly in the decoder one. Nevertheless, the value of the coding shaping delay is chosen by the encoder that accordingly sets the decoding and presentation time stamps in the MPEG stream [11]. This determines the amount of time spent by each picture in the decoder buffer so that the overall delay be S_c.

The MPEG standard does not state how the encoder should choose the coding shaping delay, i.e., set the time stamps; nevertheless, this choice is crucial to the visual quality of the resulting stream. In fact, according to Equation (17), given the coding shaping delay, picture dimension cannot exceed $S_c \cdot B$. The upper bound on the dimension of pictures encoded by a CBR encoder is determined by the VBV dimension, as stated by Claim 1. As a consequence, once the coding shaping delay has been set, the encoder must set the VBV dimension according to

$$V_s \leq S_c \cdot B \tag{20}$$

In Section 5.1.3 it has been shown that VBV dimension affects picture dimension, its variability, and hence visual quality of the encoded stream. Thus, a small coding shaping delay reduces the end-to-end delay of the system, but delivers non constant quality. The implementation of the rate control function must find a tradeoff.

Once the startup and coding shaping delays has been chosen (and the VBV dimension set accordingly), the production of a suitable MPEG stream is guaranteed by the rate control function avoiding the VBV to overflow or underflow. It is always possible to increase the natural bit rate because unencoded pictures are much larger than encoded ones. It is always possible to decrease the bit production rate, at worst by stopping encoding and using a particular code in the MPEG stream that represents a MB in which all the DCT coefficients are null.

5.1.6 Experimental Data

The `dvdenc` software MPEG encoder used to provide the experimental data shown in this work has been implemented for MPEG compression of video sequences to be stored on video disks. Thus, it is not optimally designed for a real-time videoconferencing system. As a consequence, the control function of the CBR encoder is not aggressive enough to properly limit picture dimension when the VBV is small.

The minimum VBV dimension is $B \cdot T$. According to Claim 1, a smaller dimension would deliver pictures smaller than $B \cdot T$; in this case the encoder would not be able to emit bits at the target rate for a whole video frame period. Due to the limited aggressiveness the control function, the dimension of the buffer must be chosen quite larger that this minimum and thus its impact on picture dimension is not clearly visible.

Figure 51 shows the number of bits produced by intra-frame coding the four sequences used throughout this work. Three different target rates are used for the encoding of each sequence in order to show how the parameter influences the number of bits per frame produced and its variation. The ratio between VBV dimension and target rate is kept constant so that all the configurations deliver the same coding shaping delay.

As the rate increases the dimension of frames grows larger and the quantization parameter (averaged on the whole frame) decreases, as shown in Figure 52. Accordingly, the quality of the encoded sequence becomes higher, as shown by the SNR plotted in Figure 53.

Since in the experiments the ratio between VBV dimension and target rate is constant, according to Equation (19), more variability is allowed for picture dimension as the target rate increases. This is confirmed by the plots in Figure 51.

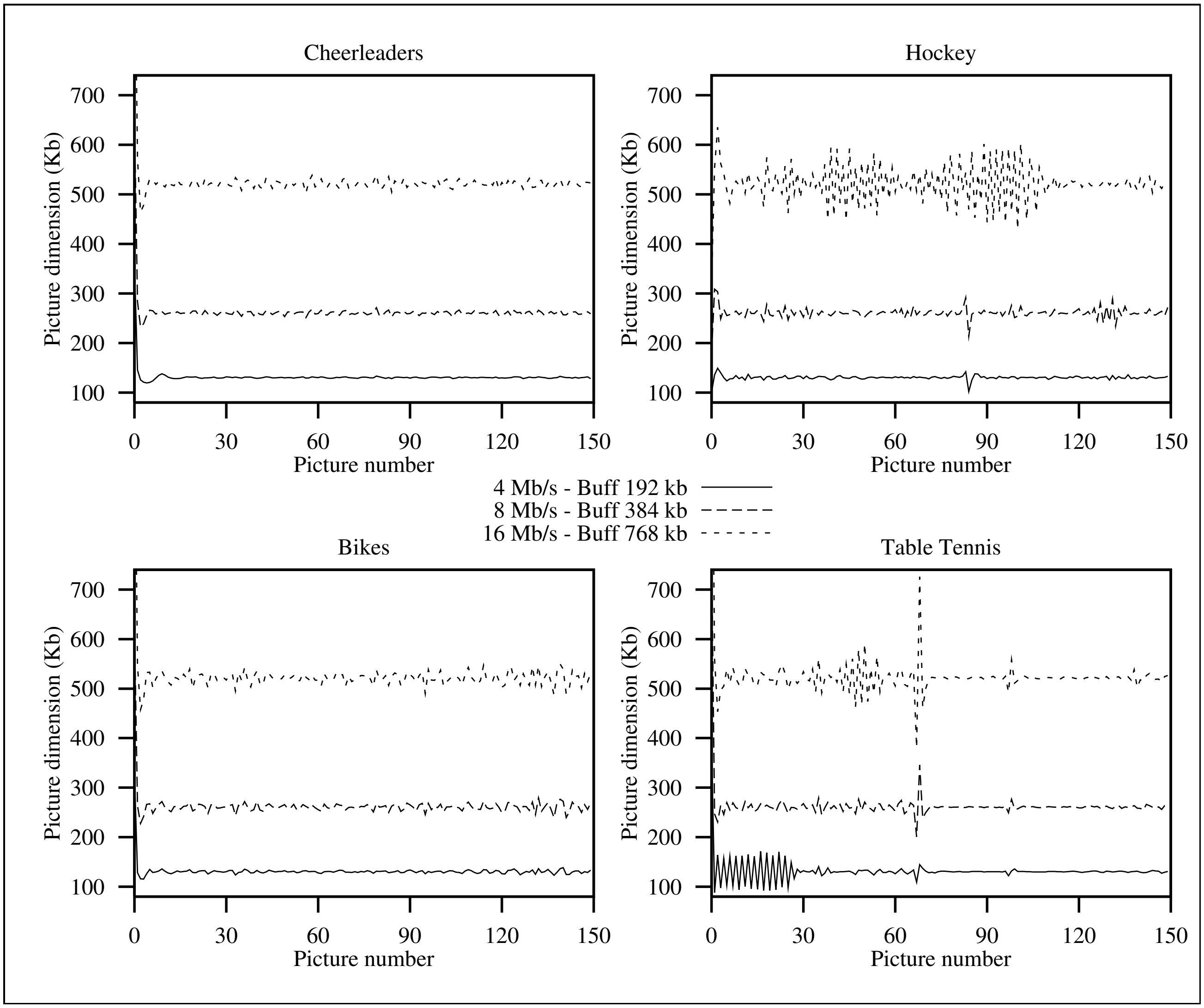

Figure 51: Frame Dimension in Intra-frame Only CBR Encoding.

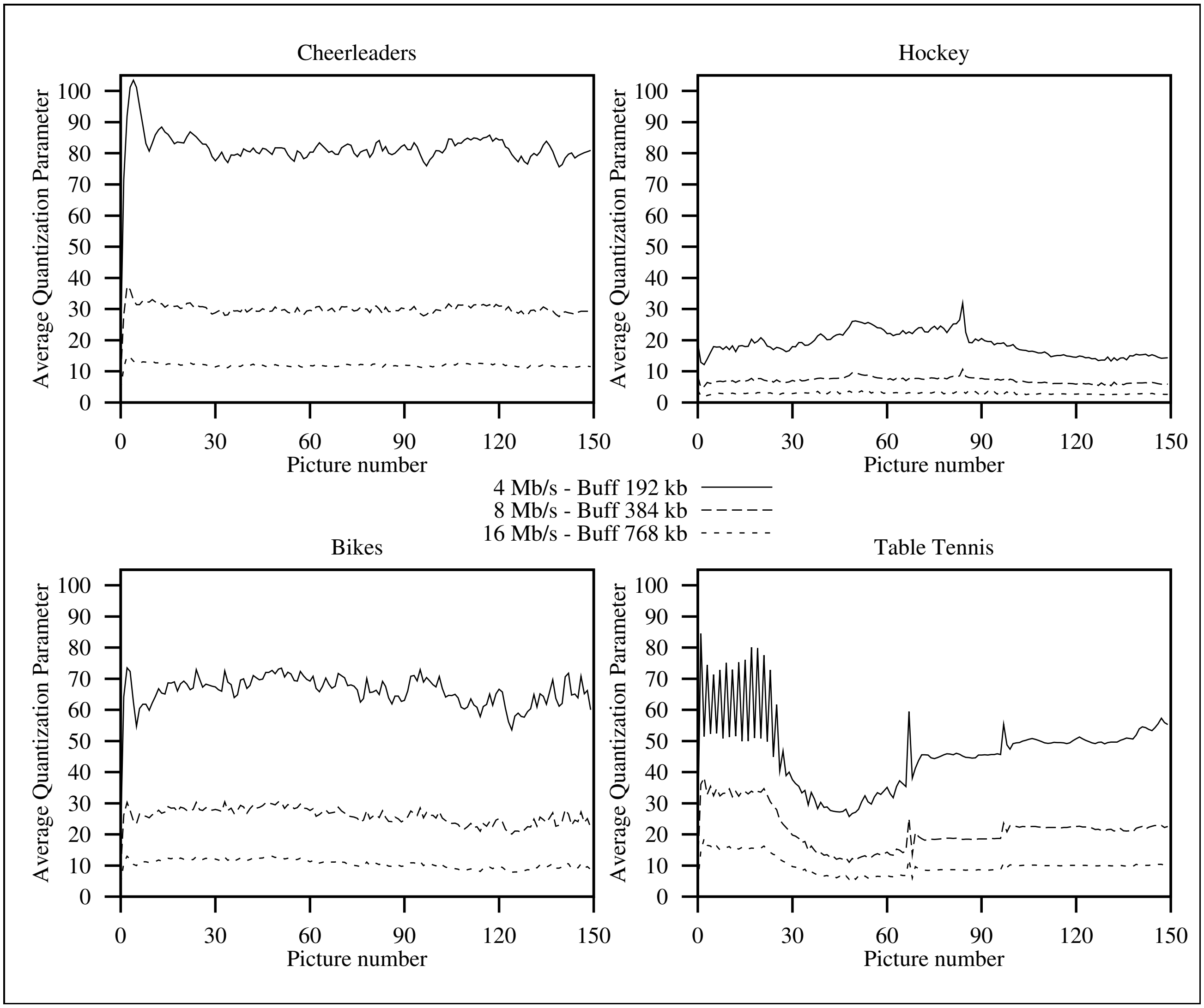

Figure 52: Average Quantization Parameter in Intra-frame Only CBR Encoding.

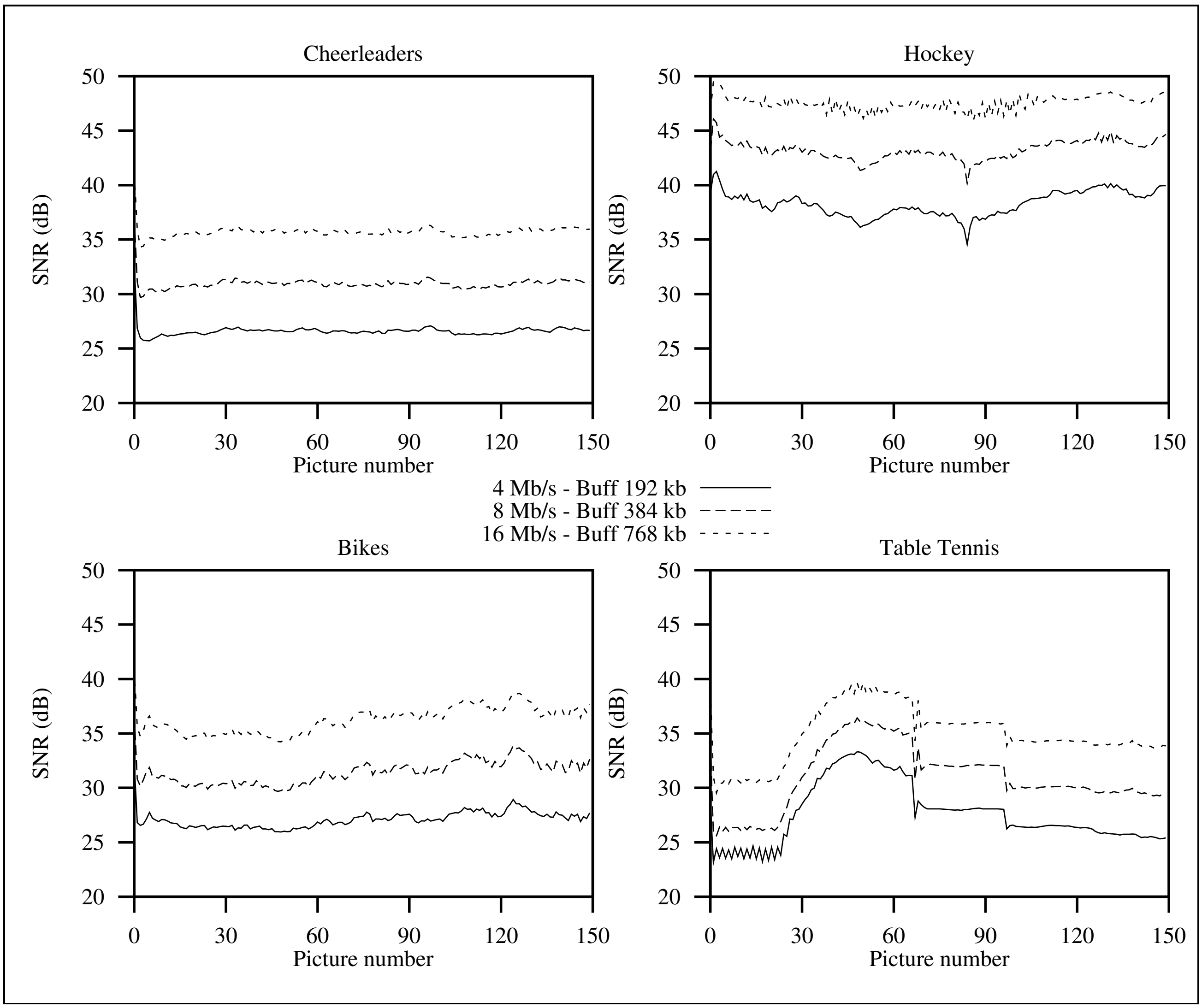

Figure 53: SNR in Intra-frame Only CBR Encoding.

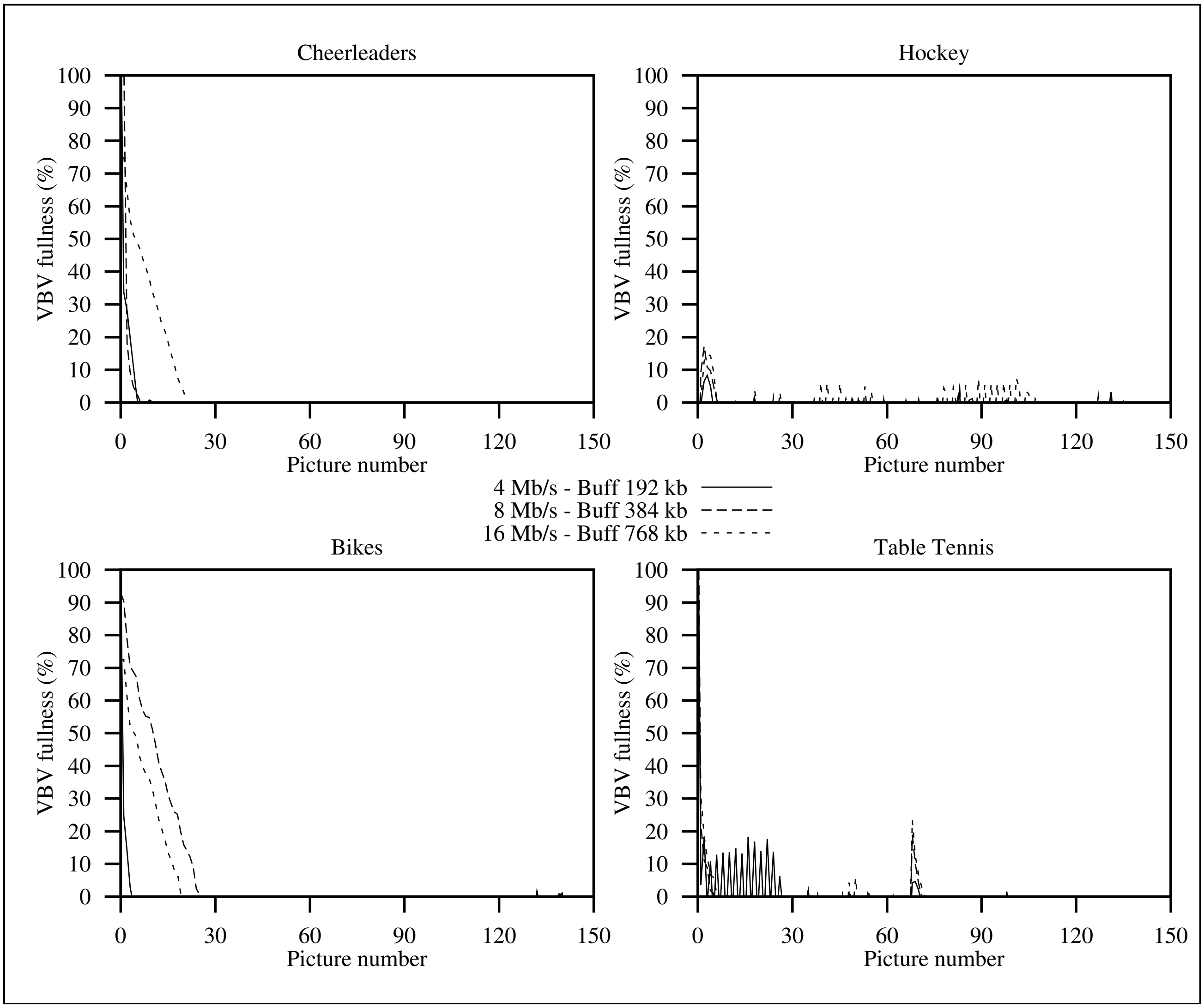

Figure 54: VBV Fullness in Intra-frame Only CBR Encoding.

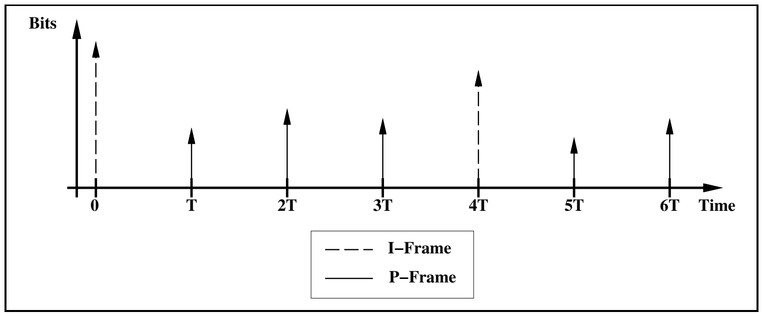

Figure 55: Amount of Bits Encoding Each Picture.

In the "hockey" scene at 16 Mb/s the dimension of adjacent pictures varies very much due to the small value of the quantization parameter (see Figure 52). The global distortion level G is an integer and is varied by 1; due to the low value of G, the relative variation is large and is reflected in a large variation of picture dimension.

5.2 Intra-frame and Predictive Coding

Predictive coding yields better compression than intra-frame coding, as qualitatively shown in Figure 55 and confirmed by the experimental results shown in Figure 20. Thus, if predictive coding is used to encode a scene with a given visual quality, the whole number of bits produced is smaller than if only intra-frame coding is used; the larger the GOP (i.e., the number of P-frames per I-frame), the smaller the amount of bits produced[13]. Moreover, the slower the scene, the smaller the dimension of P-frames; since videoconferencing scenes are likely slow, predictive coding must be used to increase compression.

In the following of this Section, the impact of predictive coding on coding shaping delay is studied. We show that coding shaping delay makes CBR MPEG with predictive coding unsuitable for videoconferencing applications. In order to provide a better understanding of coding shaping delay, we first show where pictures experience it. In Section 5.2.2 we show that when a CBR MPEG encoder uses predictive coding, given the target rate, the slower the sequence, the higher the visual quality of the encoded sequence. Nevertheless, the coding shaping delay is larger than when using only intra-frame coding. Then, we show how predictive coding increases the coding shaping delay when encoding at constant quality. Lastly, the impact of predictive coding on the startup shaping delay is considered.

[13]Actually, as shown in [19], there is a maximum GOP dimension beyond which enlarging the GOP does not decrease the amount of bit generated because the prediction error accumulates and requires many bits to be encoded.

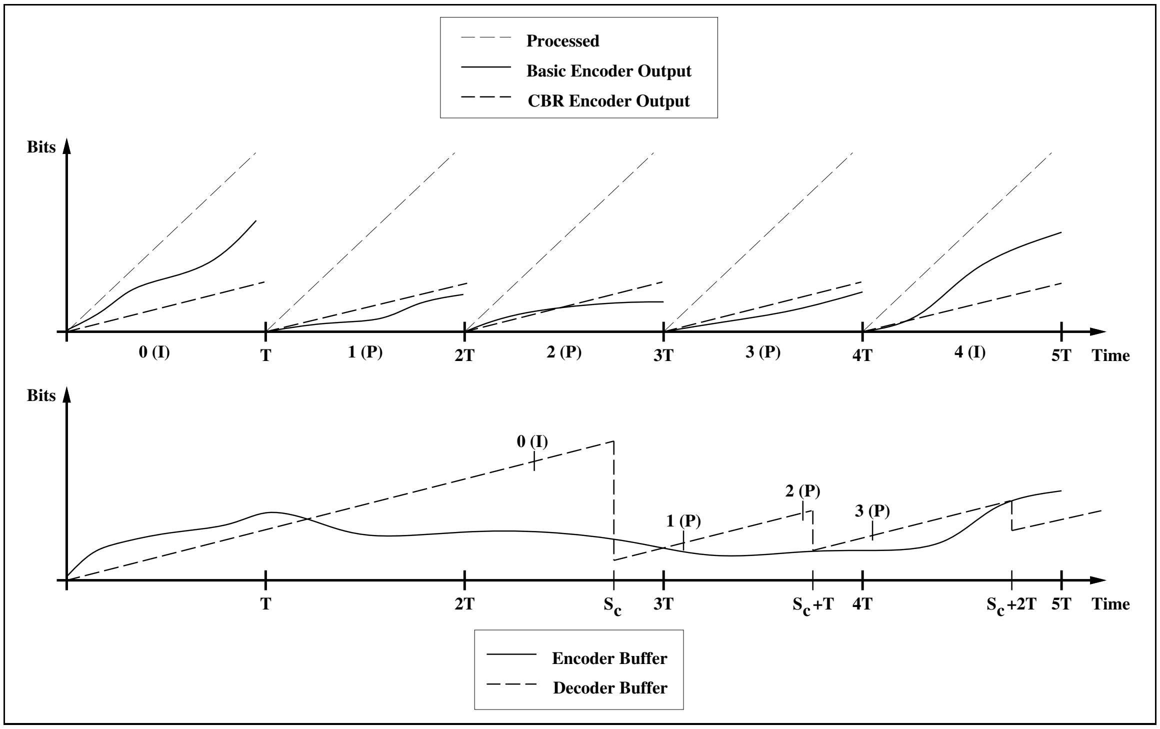

Figure 56: Natural and Target Generation of Bits and Encoder and Decoder Buffer Fullness.

5.2.1 Buffering Issues Related to the Coding Shaping Delay

In Section 5.1.1 it has been shown that a CBR MPEG encoder/decoder system introduces a coding shaping delay in order to allow pictures to be encoded at constant rate with variable amount of bits. Pictures experience part of it in the encoder buffer and part in the decoder buffer, as explained in detail in this Section.

Figure 56 shows how the bits of an encoded video stream are produced and buffered by a CBR encoder/decoder system using both intra-frame and predictive coding. The upper diagram qualitatively shows that I-frames are larger than P-frames. The natural rate averaged on a video frame period while the basic encoder is intra-frame coding, is larger than the target rate; the natural rate while predictive coding is smaller than the target rate.

The lower diagram plots the fullness level of the encoder (continuous line) and the decoder (dashed line) buffer. P-frames find a backlog in the encoder buffer due to the high number of bits produced by the basic encoder for the previous pictures and still not sent out by the CBR encoder. The decoder buffer level is incremented linearly and decremented by retrieval units corresponding to a frame. Encoded pictures are not retrieved from the buffer and decoded as soon as they arrive because they experience the processing resynchronization delay unless their dimension is the maximum allowed by the system. For example, picture 0, which is intra-frame coded, fully enters the decoder buffer (exits the encoder) at the time tagged `0(I)`, but it is not decoded until the instant S_c, i.e., until the coding shaping delay is elapsed from its capture.

5.2.2 Coding Shaping Delay at Constant Target Rate

We consider the encoder operating at a fixed target rate and study the effect of the coding shaping delay on the visual quality of the encoded stream. This is a realistic scenario as CBR MPEG encoders usually have their fixed target rate and change the ratio between I-frame and P-frame dimension according to the motion in the scene being encoded. If it is slow, P-frames are small and more bits out of the GOP budget are used in encoding I-frames. If a scene is fast, P-frames require more bits and the dimension of I-frames decreases accordingly. In order to have a simple (even though not accurate) measure of the visual quality which allows different configuration to be compared, we make the following Assumption.

Assumption 4 *If the same scene is encoded more than once with different encoding parameters, the visual quality of the encoded streams obtained with the different experiments can be compared by comparing the dimension of I-frames: larger I-frames mean better visual quality.*

According to this Assumption and to what stated above, given the target rate, the fast moving is a scene, the lower the quality.

In a videoconferencing scenario, scenes are usually slow and thus, given a target rate, the encoder produces large I-frames. By doing this it introduces a large coding shaping delay which is lower bound by the dimension of the largest picture according to Disequation (17).

We now devise an expression that gives the coding shaping delay as function of P-frame dimension and other parameters of a CBR encoder that exploits predictive coding. We assume that the encoder uses a constant number of bits to encode each GOP: it must produce these bits in a GOP period, i.e., each GOP is encoded with $N \cdot T \cdot B$ bits. Actual CBR do not usually comply to this assumption; thus, the expression we devise does not provide the coding shaping delay of any particular encoder. Nevertheless, it gives the flavor that CBR encoders for videoconferencing applications should better not exploit predictive coding in order to reduce the end-to-end delay.

According to the foregoing assumption, the following equation holds for each GOP

$$F^I + \sum_{i=1}^{N-1} F_i^P = N \cdot T \cdot B \tag{21}$$

where F^I is the dimension of the I-frame in the GOP, F_i^P is the dimension of the i^th P-frame. By extracting F^I from Equation (21), we obtain, for each GOP,

$$F^I = N \cdot T \cdot B - \sum_{i=1}^{N-1} F_i^P \qquad (22)$$

The smaller the number of bits used to encode P-frames, the larger the dimension of the I-frame in the GOP.

Equation (18) in Section 5.1.1 gives the coding shaping delay $S_c = \max_{seq} F/B$, where $\max_{seq} F$ is the dimension of the largest encoded picture in the sequence. When predictive coding is exploited, the largest picture is the I-frame in the GOP with the smallest P-frames. For the sake of simplicity (also in the notation), we assume that each I-frame has the same dimension, i.e., $\max_{seq} F = I$ as given by Equation (22); substituting in Equation (18):

$$S_c = \frac{N \cdot T \cdot B - \sum_{i=1}^{N-1} F_i^P}{B} = N \cdot T - \frac{\sum_{i=1}^{N-1} F_i^P}{B} \qquad (23)$$

Picture Dimension The smaller the P-frames, the longer the coding shaping delay that the encoding/decoding system must introduce. If the encoded scene is completely static, the P-frames are encoded with a very small amount of bits; in principle 0. Thus, the I-frame grows using all the bits intended for the encoding of the GOP, i.e., $N \cdot T \cdot B$; this corresponds to a coding shaping delay equal to the GOP period $N \cdot T$.

Thus, a CBR encoder which aims at yielding a short coding shaping delay must bound the ratio between the dimension of I-frames and P-frames, even though the scene to encode is slowly moving and would allow a large ratio. The key element in determining the number of bits used to encode pictures is the VBV dimension V_s; a dimension equal to $N \cdot T \cdot B$ allows the encoder to take full advantage of predictive coding when the scene is particularly slow and to use a large amount of bits to encode I-frames. A larger dimension allows the encoder to deliver even more uniform quality by smoothing sudden increase in the complexity of pictures and in the motion over more than a GOP period. Nevertheless, this requires larger coding shaping delay and buffers in both encoder and decoder.

Beside the VBV, CBR encoders better have other means for controlling the number of bits used to encode pictures. For example, the software encoder `dvdenc` before encoding a picture sets a target dimension; this is chosen according to a predefined amount of bits to be used in the encoding of each GOP and a preferred ratio between dimension of I-frames and P-frames. This is particularly useful when a particularly complex picture (i.e., with low spatial redundancy) is being intra-frame coded; if the encoder used only the VBV to determine the target for the picture dimension, in order to deliver good quality, it would use the maximum number of bits allowed by the system to encode a picture, i.e., it would fill up the VBV. According to Claim 2, the dimension of following picture(s) would be bound by $B \cdot T$; this bound can be too small, particularly if the scene becomes suddenly fast. In fact, P-frames are encoded with a number of bits that is not sufficient to show the same visual quality of the I-frame corresponding I-frame and thus the quality of the GOP is lowered.

GOP Size Increasing the GOP size (i.e., N) increases the coding shaping delay because the percentage of small pictures (the P-frames) increases and so I-frames are made larger to keep the target rate; even though according to Assumption 4 this increases the visual quality of the encoded scene, it should be avoided to keep the delay smaller[14].

Using a low video frame rate (i.e., large T) increases the coding shaping delay, as shown by Equation (23); the longer the video frame period T, the larger the coding shaping delay. Nevertheless, Equation (22) shows that increasing the video frame period also improves the quality of the encoded scene because I-frames become larger.

Conclusion 1 *Given the target rate of a CBR MPEG encoder, predictive coding increases the visual quality. Nevertheless, every parameter setting aimed at quality improvement increases the coding shaping delay.*

5.2.3 Coding Shaping Delay at Constant Visual Quality

We now study the impact of predictive coding on the coding shaping delay, given the visual quality the CBR encoder must deliver. We assume to encode the same scene more than once with different amounts of compression due to predictive coding. I.e., for each experiment the parameters of the encoder are changed so that predictive coding is more or less heavily exploited and the visual quality delivered is the same.

The visual quality of the encoded streams is compared by comparing the dimension of I-frames, according to Assumption 4; we need a way to compare the effectiveness of predictive coding. The exploitation of predictive coding is as more effective as much P-frames are smaller than I-frames and as much GOP size N is larger.

According to Assumption 4, if the same scene is encoded with same visual quality, but using a different number of P-frames per GOP, the I-frames produced in each experiment have the same dimension F^I. Assumption 4 is a rough approximation, but it is close to reality when dealing with simple slow moving scenes (such as a typical videoconference); moreover, the aim of this Section is only to give the flavor of the impact of predictive coding on the coding shaping delay, not to provide a general analytical relationship between the two. In Section 4.1.2 we stated that usually the dimension of I-frames is from 2 to 4 times the dimension of P-frames; the slower the encoded scene, the larger the ratio between I-frame and P-frame dimension. In the context of the analysis carried on in this Section, we assume that P-frames are encoded with constant amount of bits P and exploit as measure of yielded compression the ratio

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

The rate of the encoded stream can be roughly expressed as

[14]Increasing the GOP size also renders the encoded stream more sensitive to errors and losses.

$$B^N = \frac{F^I + (N-1) \cdot F^P}{N \cdot T} \tag{24}$$

where N is the number of pictures per GOP and F^P is P-frame dimension, which we assume to be constant for all the P-frames. Equation (24) can be rewritten as

$$B^N = \frac{N + \alpha - 1}{\alpha \cdot N \cdot T} \cdot F^I \tag{25}$$

To better highlight the effect of predictive coding on the rate of the encoder, we also provide the encoder rate when only intra-frame coding is used:

$$B^I = \frac{F^I}{T} \tag{26}$$

By merging this equation with (25), the latter can be written as

$$B^N = B^I \cdot \frac{N + \alpha - 1}{\alpha \cdot N} \tag{27}$$

Due to the approximations made to devise this Equation, it does not exactly reflect the real behavior, but it gives the flavor that the more effective predictive coding (i.e., the ratio α), the smaller the target rate needed to deliver the same visual quality. Also, the larger the number of P-frames per GOP (N), the smaller the target rate.

We now analyze the effect of predictive coding at constant quality on coding shaping delay. Equation (18) in Section 5.1.1 gives the coding shaping delay as $S_c = \max_{seq} F/B$, where $\max_{seq} F$ is the maximum dimension of frames and B is the target rate of the CBR encoder. According to Assumption 4, $\max_{seq} F = F^I$ and the coding shaping delay is expressed as

$$S_c = \frac{F^I}{B}$$

Being F^I constant when the quality is constant, the target rate of the encoder B is the only factor which determines the coding shaping delay in different configurations. By substituting the value of B with B^N given by Equation (25)

$$S_c = \frac{\alpha \cdot N \cdot T}{N + \alpha - 1}$$

The more predictive coding is effective (i.e., α), the larger the coding shaping delay. Typical values of α are between 2 and 4; the slower the motion in the scene, the larger the value of α. In a videoconferencing application, it is likely to have $\alpha = 4$; with $N = 15$ at 15 fps, $S_c = 222$ ms, i.e., the CBR encoder alone introduces in the videoconferencing system a delay larger than the 100 ms bound essential for interaction. Similar considerations apply also to the increase of N.

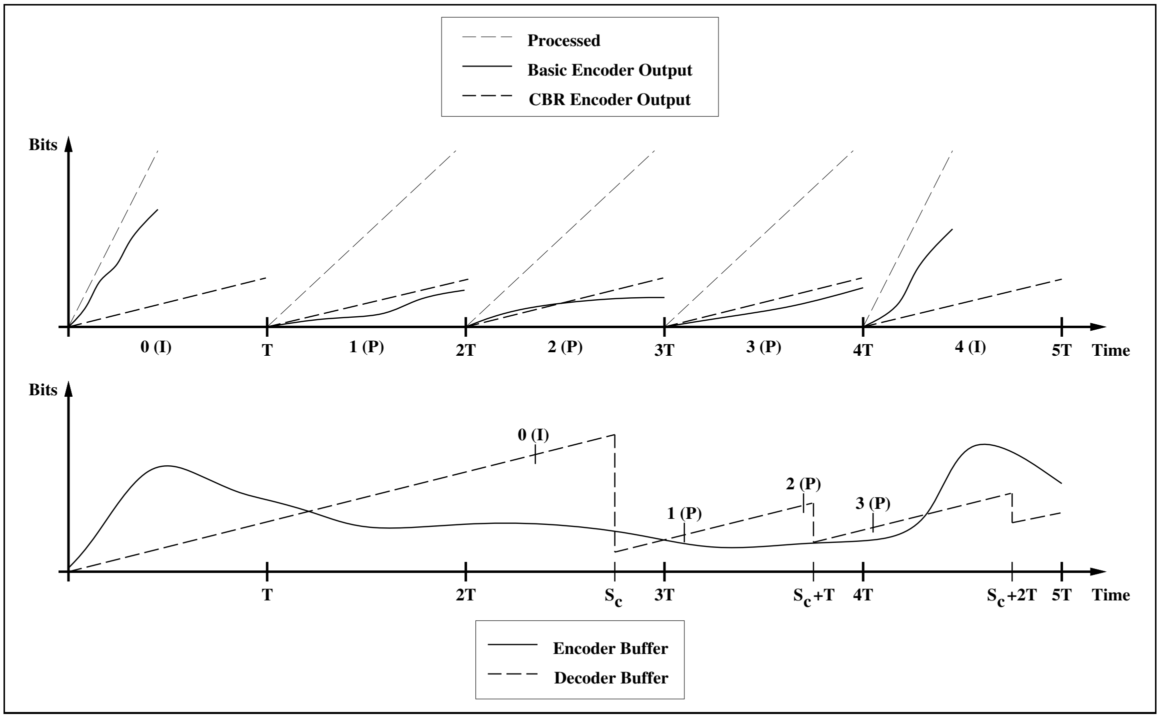

Figure 57: Realistic Bit Generation.

Conclusion 2 *Using a CBR MPEG encoder with predictive coding to build a videoconferencing system, reduces the bandwidth requirement, but it is not acceptable from the point of view of the end-to-end delay.*

5.2.4 Startup Shaping Delay

As shown in Figure 57, a typical MPEG encoder encodes I-frames with a large amount of bits in a small time, with respect to P-frames. As a consequence, it is unlikely that the natural rate be lower than the target rate when the encoding of a GOP begins. After, the buffer has some backlog due to the large amount of bits encoding the I-frame.

The encoder buffer can happen to underflow only when P-frames are particularly small; the highest underflow probability is after the when the buffer gets empty at the end of a GOP with small P-frames. During the rest of the time a backlog is likely to be present in the encoder buffer, as can be seen in Figure 56. Thus, even though the startup shaping delay is underdimensioned, the effects (non uniform quality due to a sudden decrease of the quantization stepsize in order to increase the natural rate) are visible on a small percentage of the pictures.

5.3 Experimental Data

Figures 58-61 show the experimental data obtained by encoding the four sample sequences at three different rates. The ratio between the VBV dimension and the rate is kept constant to a value that guarantees the VBV not to be overflowed, notwithstanding the low aggressiveness of the rate control function. The GOP dimension is set to $N = 15$, i.e., each I-frame is followed by 14 P-frames.

Figure 58 shows the dimension of each picture. I-frames are typically larger than P-frames and the dimension of both I-frames and P-frames is variable. Especially at low bit rates the dimension of I-frames is double or more than double the dimension of P-frames: in order for this to be possible the coding shaping delay must be large.

In the “hockey” sequence encoded at 16 Mb/s there is almost no difference between the dimension of I-frames and P-frames. This is due to the fact that the camera is panning over the hockey court to follow the game. Pictures are simple (most of the screen is the white ice of the court) so that I-frames can be encoded with few bits; the P-frames are quantized with a small quantization parameter (Figure 59). Since the camera is panning, motion concerns the whole images and P-frames dimension is large: the first P-frame in the GOP is not much smaller than the I-frame. In order to reduce P-frame dimension the quantization parameter is decreased; being it small, decreasing it by one unit has a significant effect on the picture dimension: the second P-frame is much smaller than the first one. This causes the VBV level to decrease (Figure 61) and the quantization parameter to be reduced again to avoid underflow. This behavior is kept for the whole GOP thus leading to the oscillation in P-frames dimension and picture quality (see SNR in Figure 60). Figure 59 shows the average quantization parameter value used on each frame.

Figures 62-65 show the experimental data obtained using a GOP of 30 pictures. The results are quite similar to the one obtained with $N = 15$ except for the “Table tennis” sequence in which I-frame dimension increases (bottom right plot in Figure 62) and the VBV gets more full and empties more slowly (bottom right plot in Figure 65).

The scenes used in these experiments are not videoconferencing sequences; a videoconference is usually characterized by low motion and pictures having low complexity. Thus, when compressing them with a CBR MPEG encoder a fairly small amount of bits can be sufficient to encode I-frames with good quality; when encoding a P-frame the error between the MB that is being encoded and the reference MB is small, and thus the amount of bits required is low.

In order to emphasize this behavior of a CBR MPEG encoder, we exploited to encode a static scene. This is an extreme case of slow moving scene but is likely for a videoconferencing application if, for example, the camera is pointed at a whiteboard while somebody is explaining its content. We obtained two static scenes by repeating 120 times the same image of the “cheerleaders” and “hockey” scenes, respectively.

Figure 66 shows the dimension of frames for various target rates and VBV dimensions.

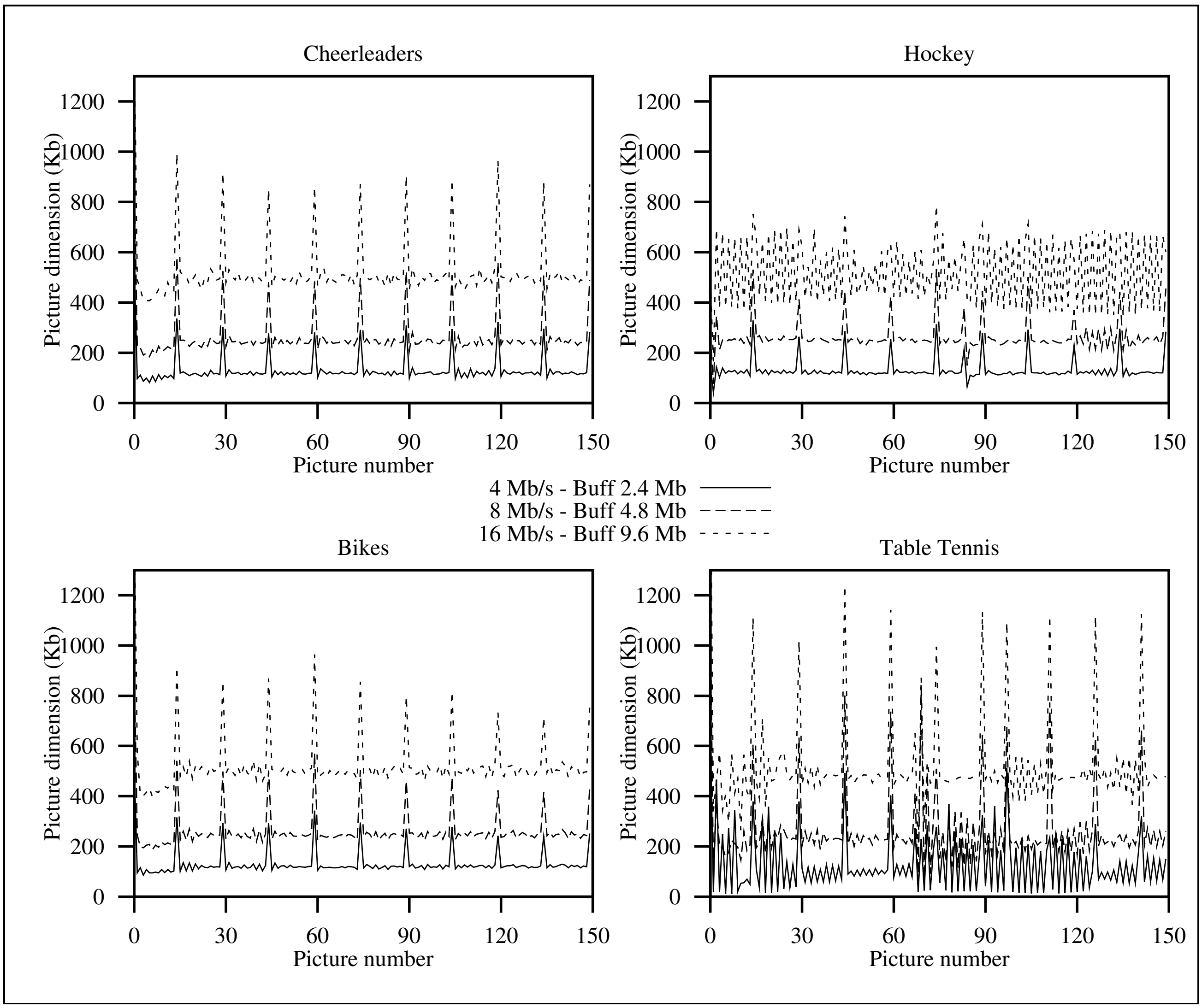

Figure 58: Dimension of Pictures in CBR MPEG Encoding with $N = 15$.

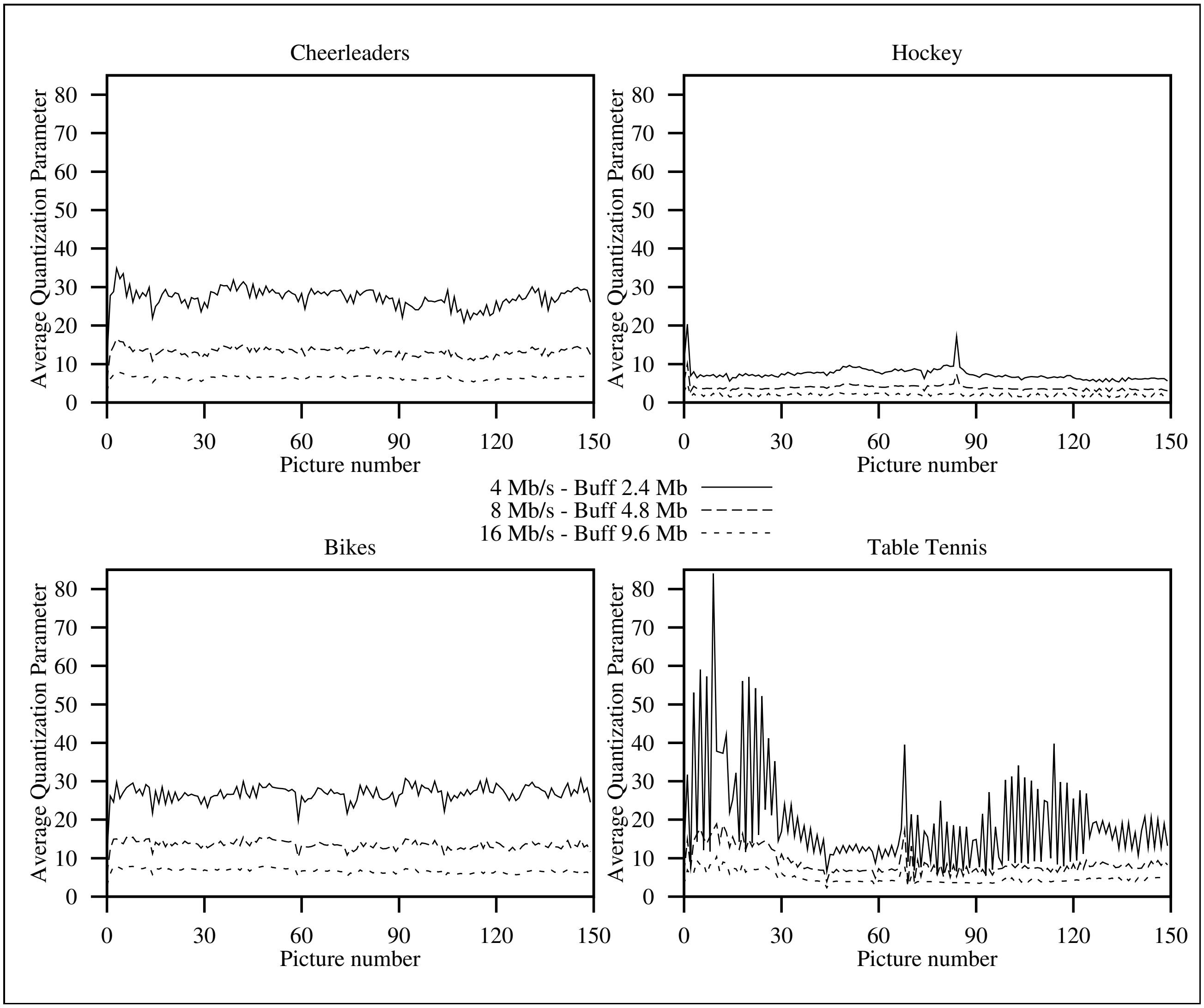

Figure 59: Average Quantization Parameter in CBR MPEG Encoding with $N = 15$.

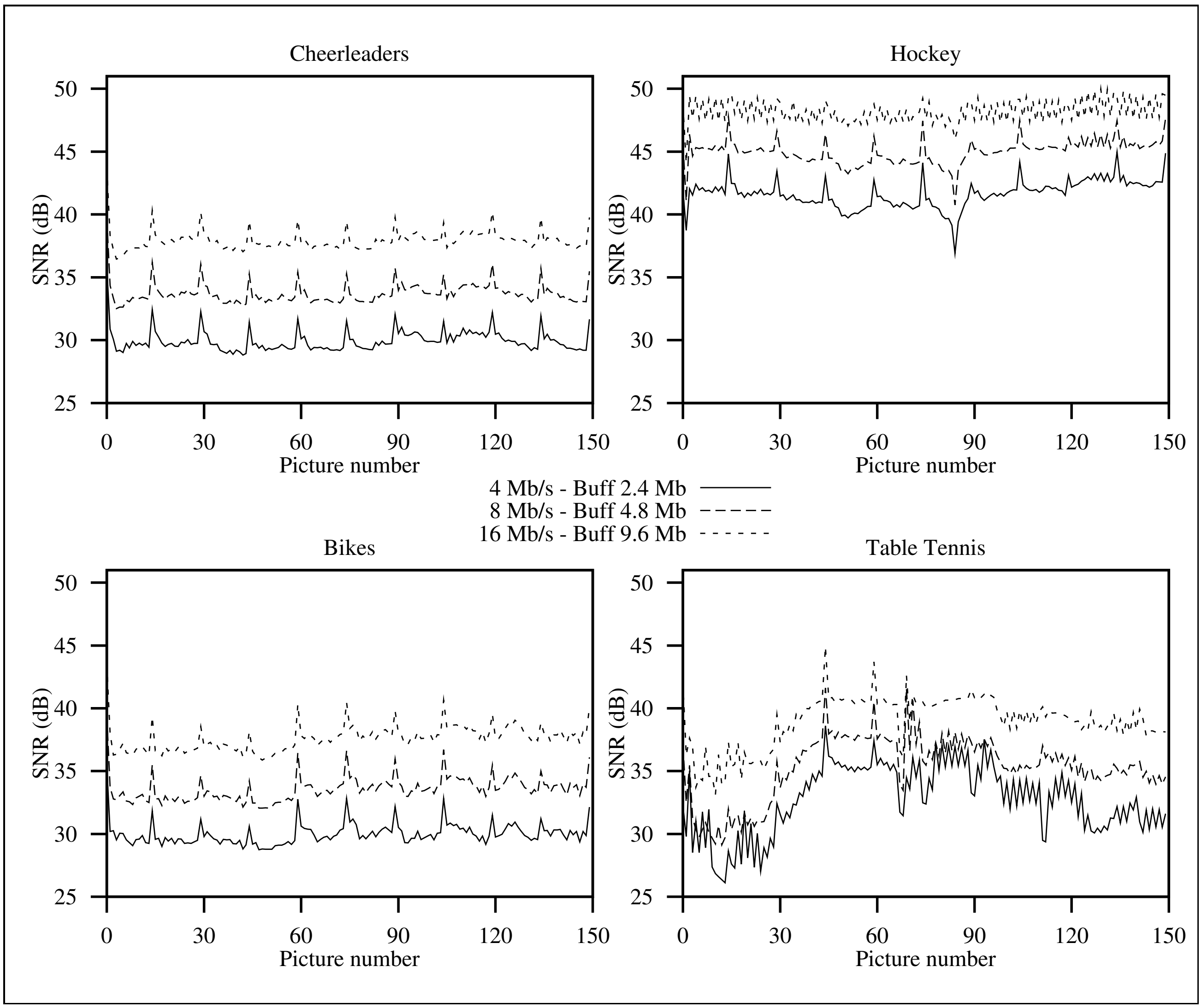

Figure 60: SNR in CBR MPEG Encoding with $N = 15$.

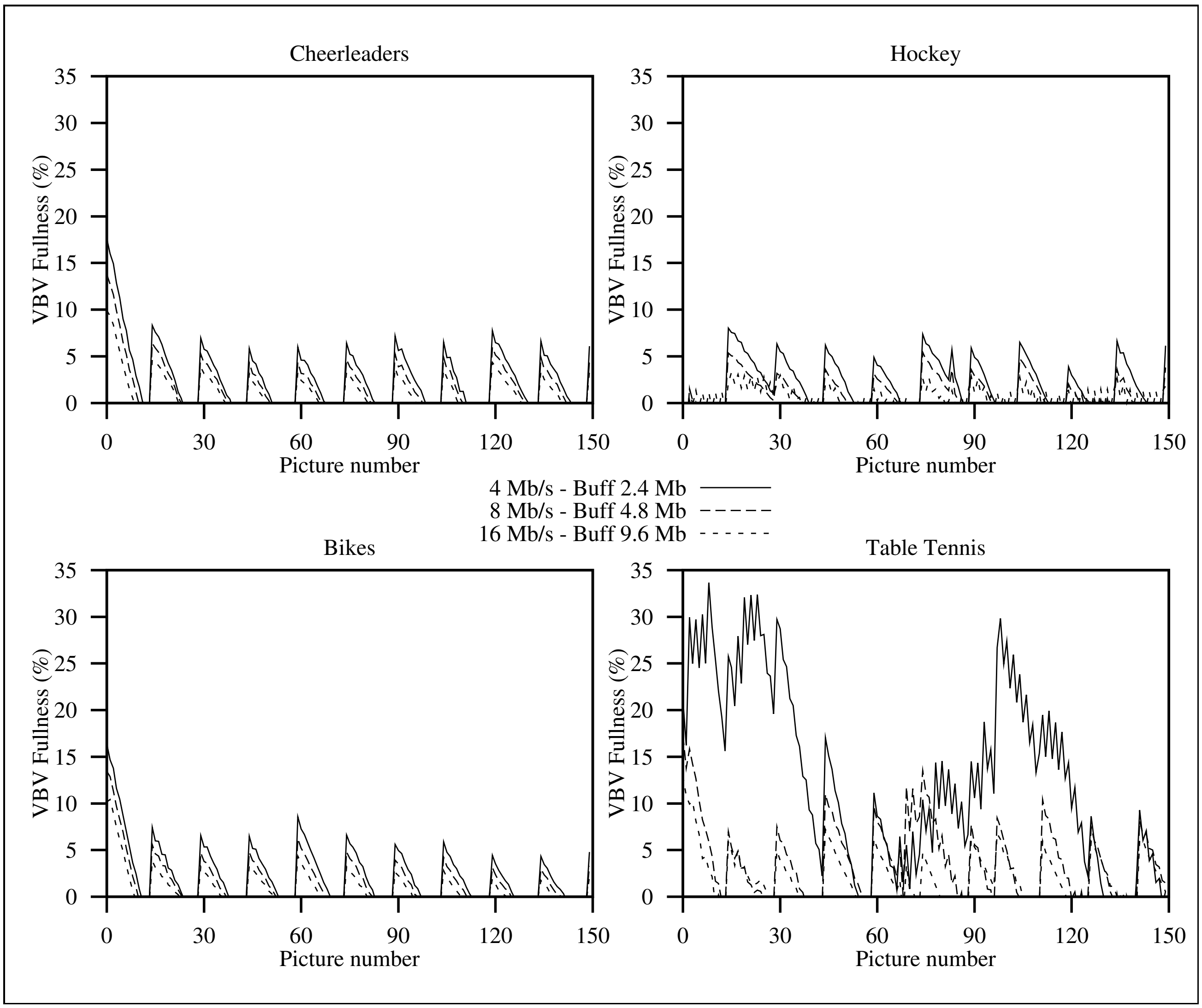

Figure 61: VBV Fullness in CBR MPEG Encoding with $N = 15$.

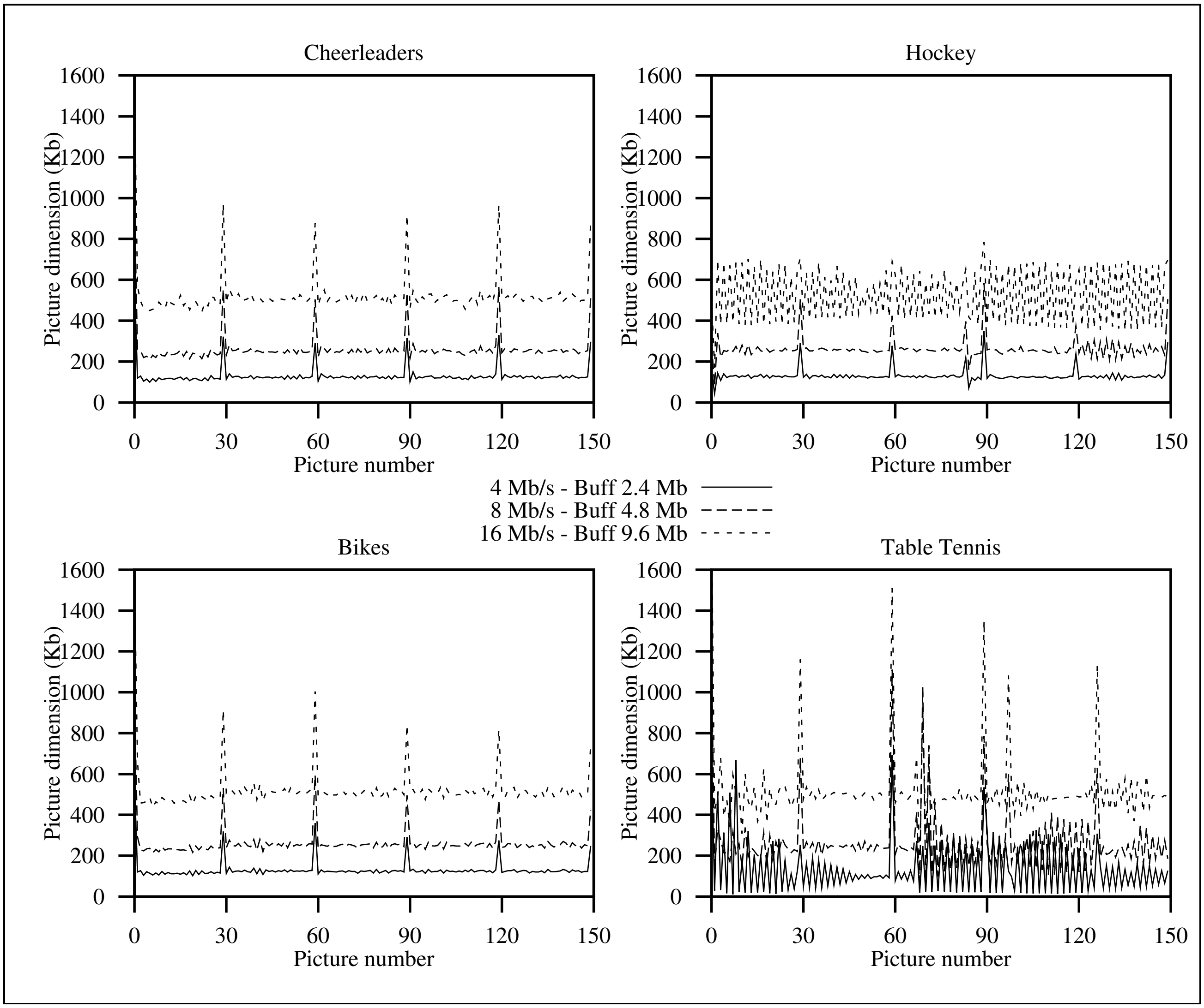

Figure 62: Dimension of Pictures in CBR MPEG Encoding with $N = 30$.

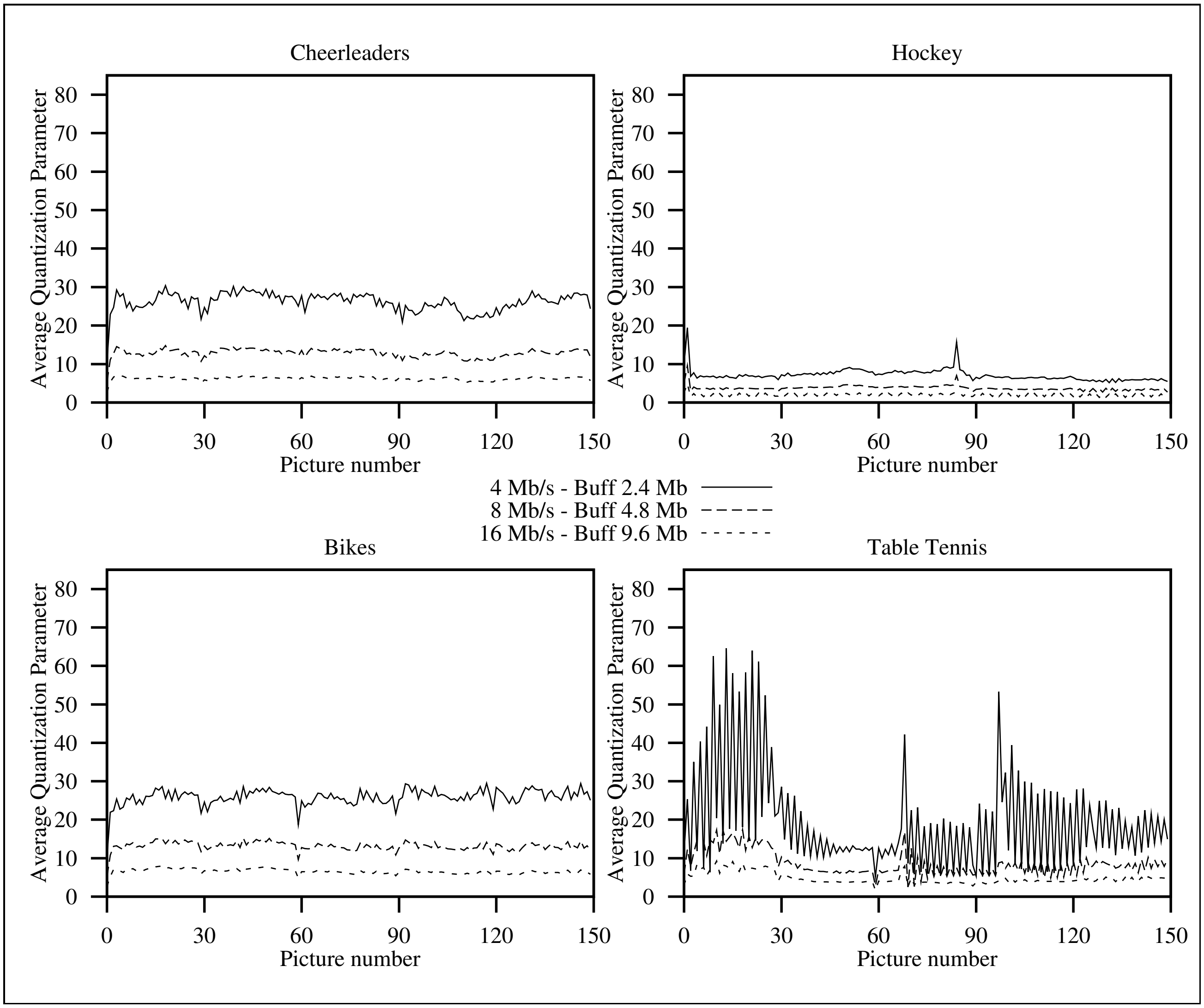

Figure 63: Average Quantization Parameter in CBR MPEG Encoding with $N = 30$.

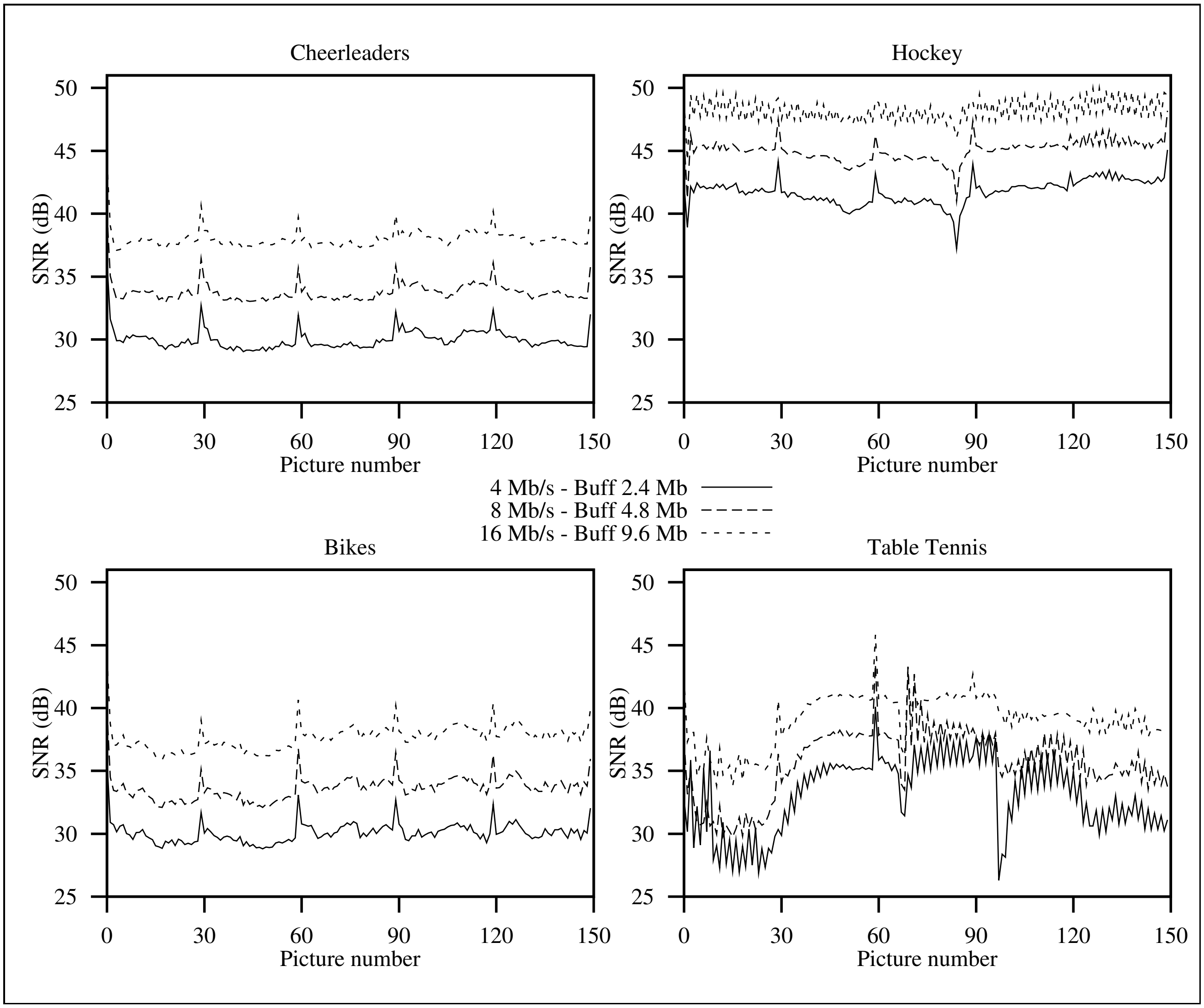

Figure 64: SNR in CBR MPEG Encoding with $N = 30$.

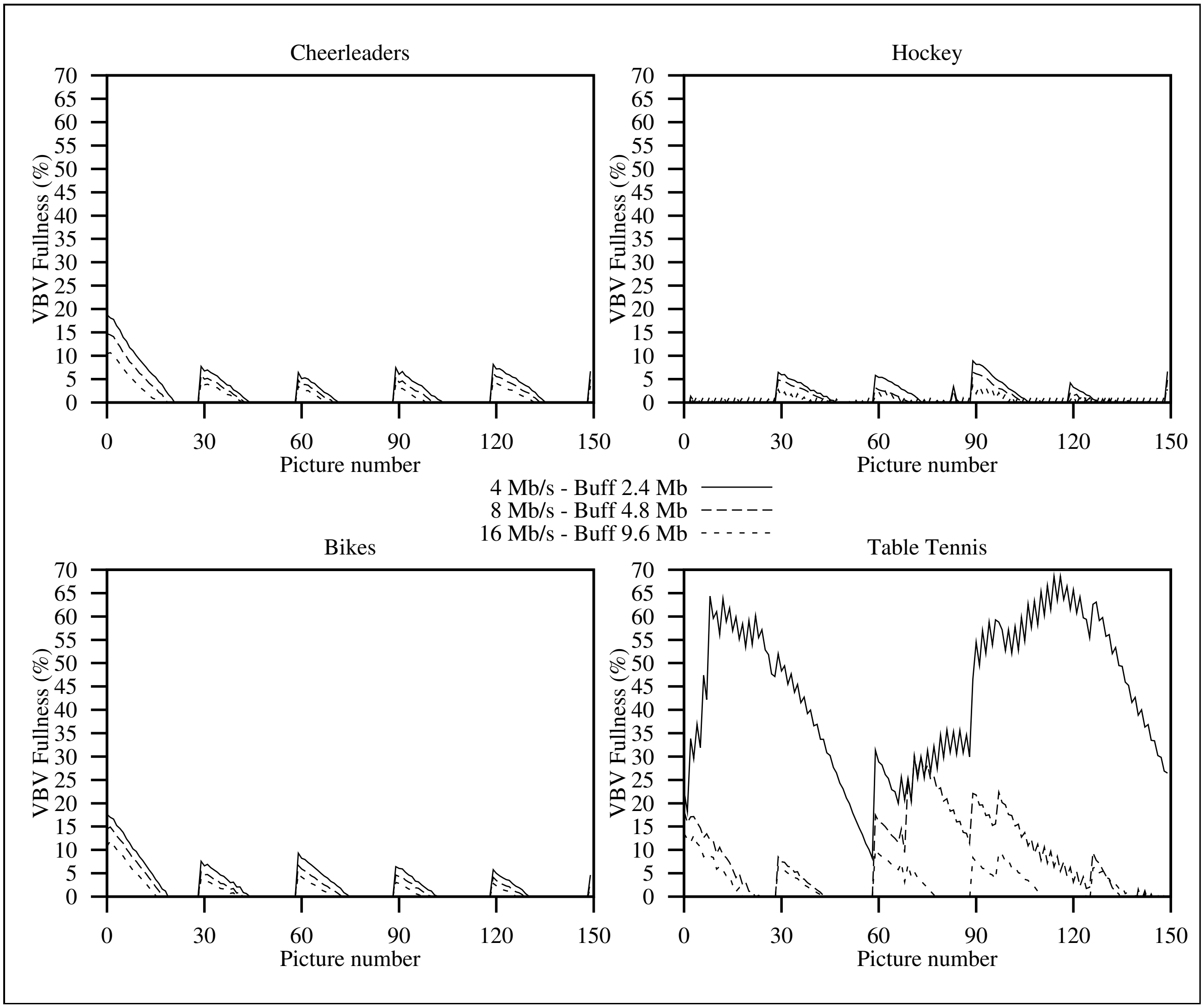

Figure 65: VBV Fullness in CBR MPEG Encoding with $N = 30$.

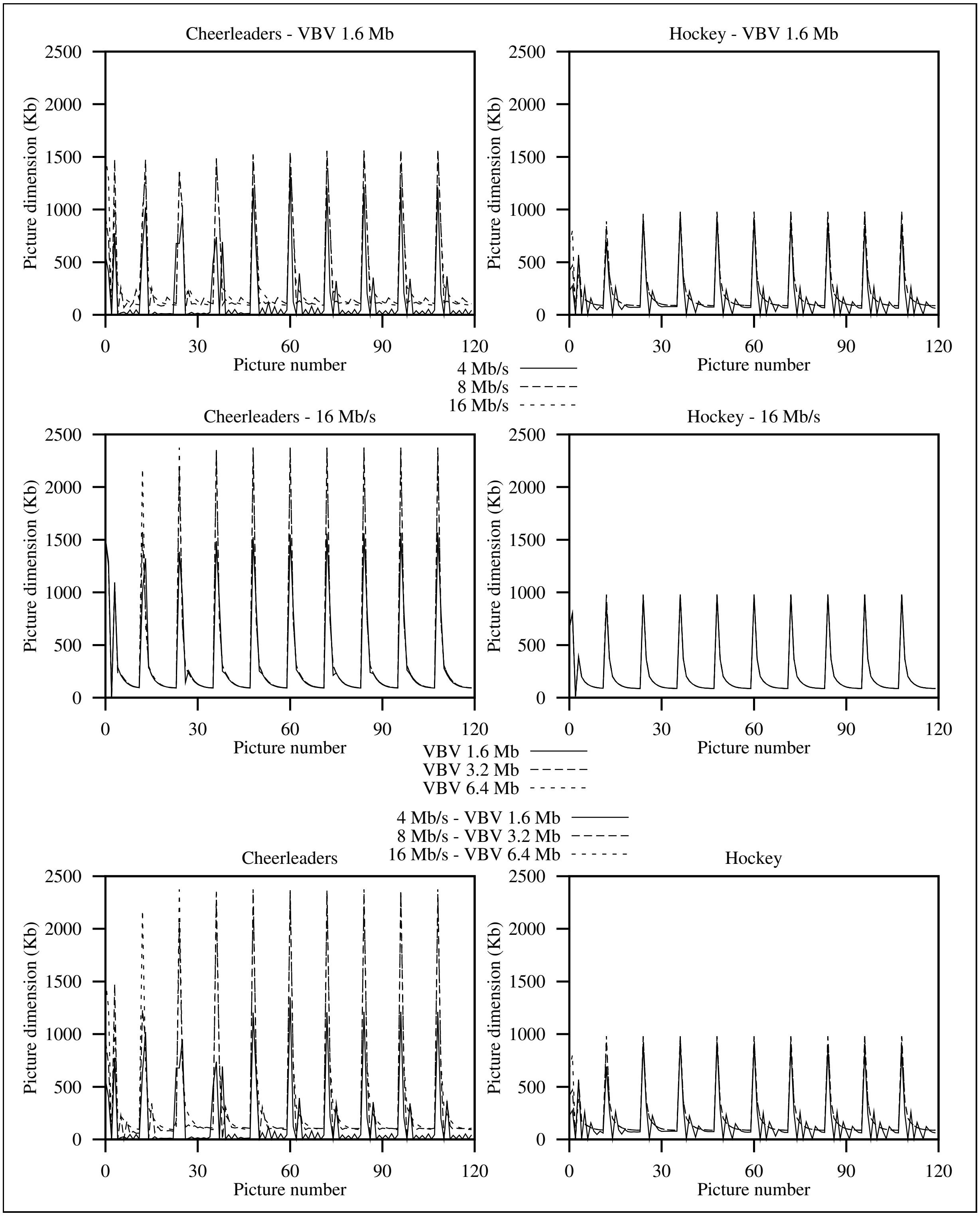

Figure 66: Dimension of Pictures in CBR MPEG Encoding of a Static Scene

In any plot the ratio between I-frame and P-frame dimension is larger than when the corresponding non static scene is encoded at the same target rate. P-frames are small because they must only encode the difference between their reference picture and the raw picture that is due to the quantization. Thus, the SNR delivered by P-frames is high, as shown in Figure 68.

The plot obtained by encoding the "cheerleaders" scene at 16 Mb/s with various VBV dimensions (second row, first column in Figure 66) shows that the larger is the VBV dimension, the largest are the I-frames. This does not hold for the "hockey" scene (second row, second column) because, being the pictures less detailed, even though the rate control function would allow a larger amount of bits to be produced, I-frames do not grow because the quantization parameter has the minimum allowed value and cannot be further reduced (see Figure 67).

Figure 69 shows that in many cases the VBV is (almost) completely filled, while in others the fullness level is of the order of the 20 %. Nevertheless, when dealing with real-time encoding (such as in videoconferencing applications), the scene to be encoded is not known in advance. If the videoconferencing system must be able to deliver a good quality, especially when the camera is statically set on some image, the VBV must be large enough to allow I-frames to grow and capture the details of the image[15] (e.g., second row, first column in Figure 69). When pictures are not detailed (e.g., second row, second column in Figure 69) or the motion is higher (Figure 61) only a fraction of the VBV is filled. In any case, the coding shaping delay is large because it must be set according to the VBV dimension in order to guarantee continuous playing.

5.4 Dedicated Link between Sender and Receiver

For the sake of completeness of this study, the system configuration in which sender and receiver are connected through a dedicated link (Figure 70) is taken into account. Having the whole link capacity at disposal for sending the CBR MPEG stream does not provide particular advantages. Only a fraction of the link capacity, corresponding to the target rate B of the encoder is actually exploited and the coding shaping delay provides a significant contribution to the end-to-end delay.

Given the propagation delay P, which depends on the distance between sender and receiver, the end-to-end delay of the system is

$$\Delta_{CBR}^{Ded} = S_c + S_s + P + D + P_d \tag{28}$$

where D is the decoding delay, S_c is the coding shaping delay, and S_s is the startup shaping delay. This represent a lower bound in the end-to-end delay obtainable from a videoconferencing system exploiting CBR MPEG encoding.

[15] Note that as the scene is not moving, the human eye is more sensitive to errors.

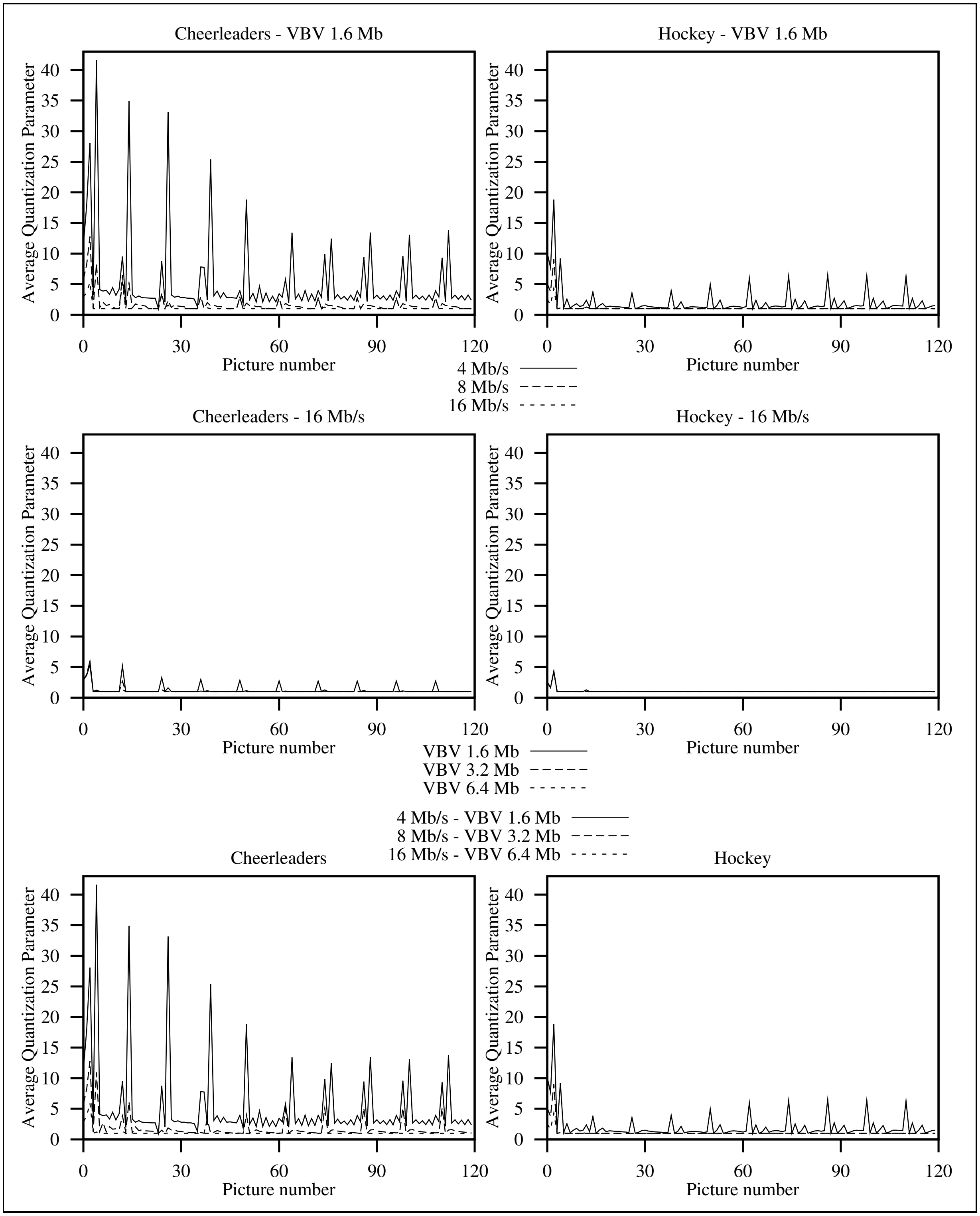

Figure 67: Average Quantization Parameter in CBR MPEG Encoding of a Static Scene

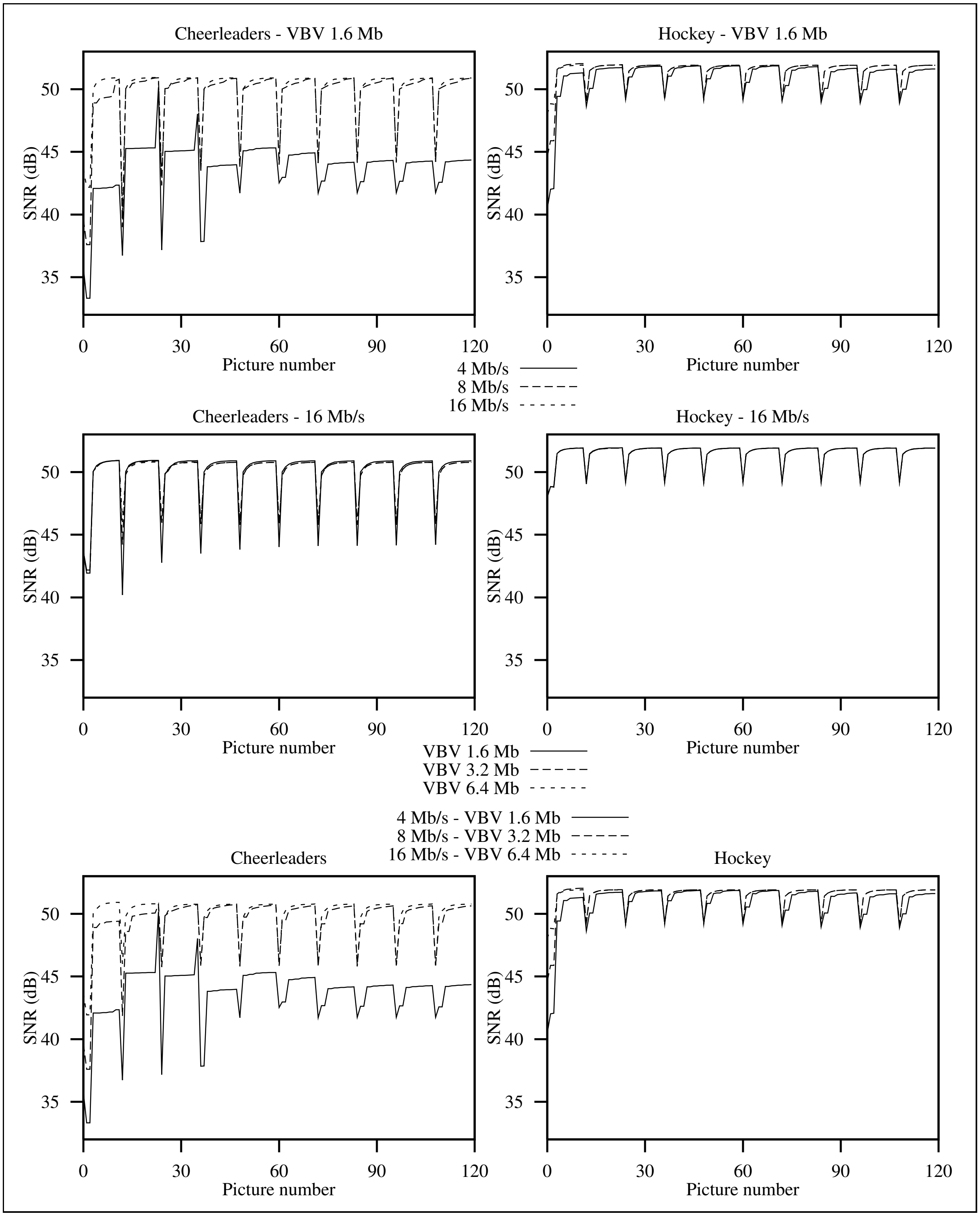

Figure 68: SNR in CBR MPEG Encoding of a Static Scene

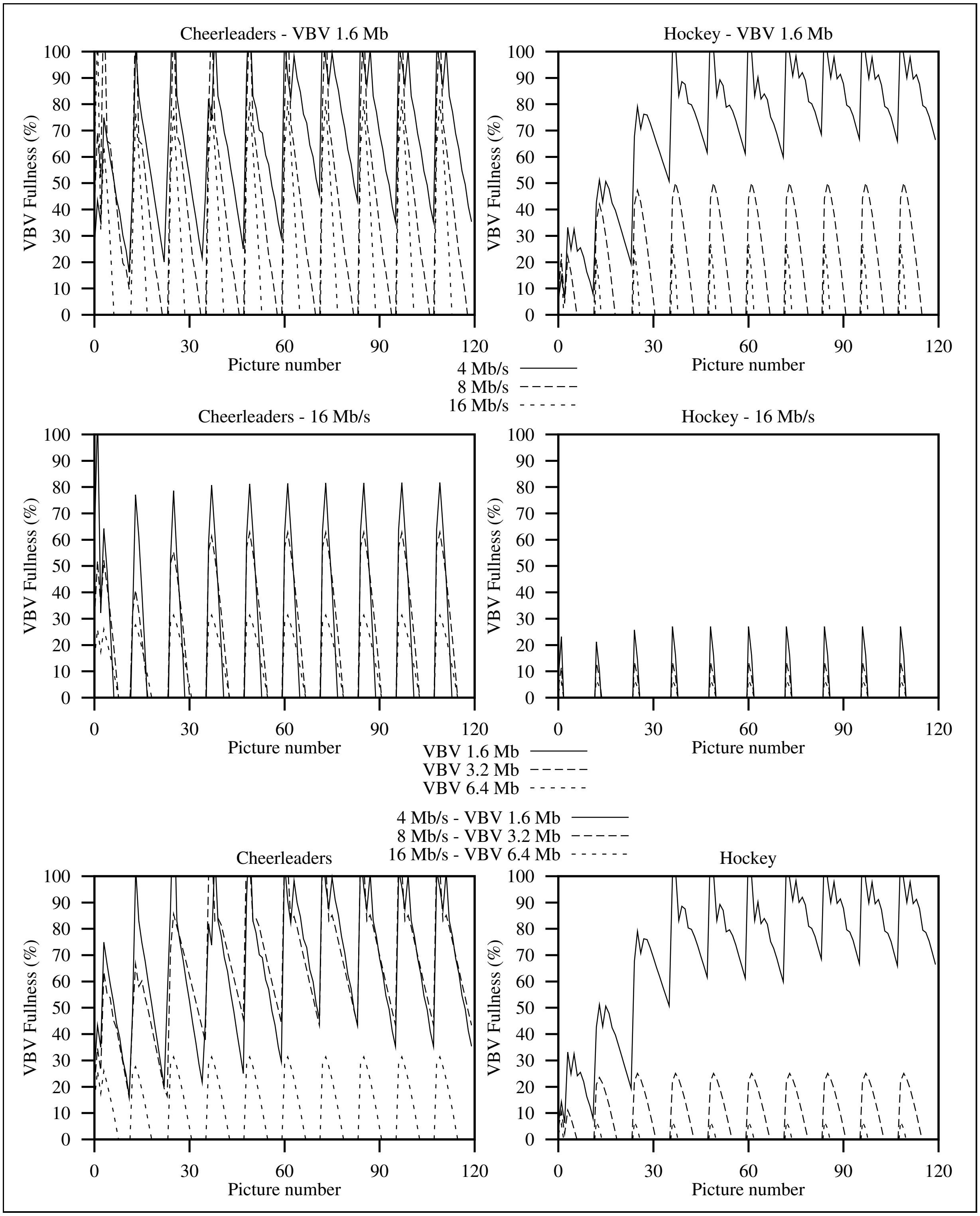

Figure 69: VBV Fullness in CBR MPEG Encoding of a Static Scene

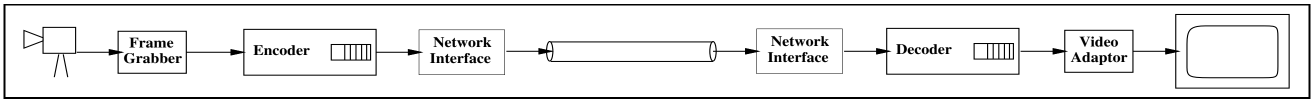

Figure 70: Architecture of a Videoconferencing System Exploiting a CBR MPEG Encoder and a Dedicated Link between Sender and Receiver.

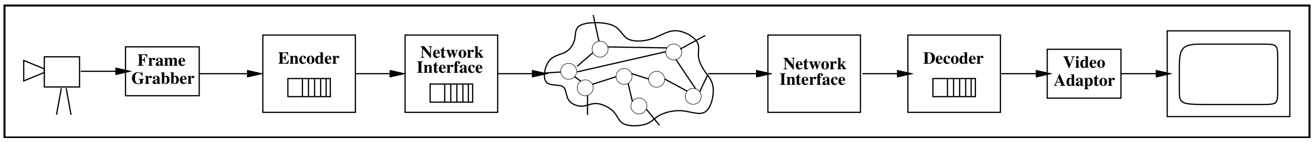

Figure 71: Architecture of a Videoconferencing System Exploiting a CBR MPEG Encoder and a Circuit Switched Network.

5.5 Circuit Switching

CBR MPEG encoding is suited for transmission over circuit switched networks: when a CBR MPEG encoder is used to build a videoconferencing system (Figure71), the most effective way to connect sender and receiver is through a circuit that has a bandwidth equal to the target rate of the encoder. The contribution of the network to the end-to-end delay is the switching delay Sw and the propagation delay P which depends on the distance between sender and receiver. The end-to-end delay of the system is

$$\Delta_{CBR}^{CS} = S_c + S_s + Sw + P + D + P_d \tag{29}$$

where D is the decoding delay, S_c is the coding shaping delay, and S_s is the startup shaping delay. This delay almost equals the one obtained with a dedicated link between sender and receiver: the only further contribution is the switching delay. Thus, it can be said that it represents as well a lower bound in the end-to-end delay obtainable from a videoconferencing system exploiting CBR MPEG encoding.

5.6 Packet Switching with Time Driven Priority

CBR MPEG encoding is best suited for transmission over circuit switched networks. Nevertheless, in the following two Sections we consider the end-to-end delay obtained when CBR encoded video is sent through a packet switched network. In this Section, time driven priority is considered to be exploited in the network. The network shaping delay introduced in the sender and receiver is analyzed first; then, the end-to-end delay is considered.

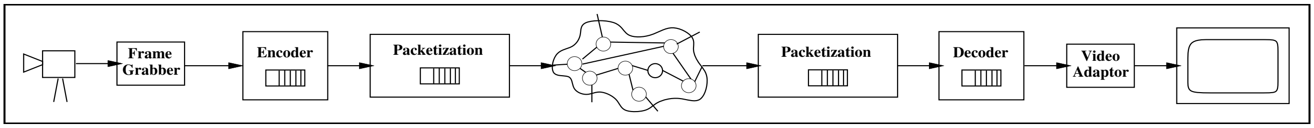

Figure 72: Architecture of a Videoconferencing System Exploiting a CBR MPEG Encoder and a Packet Switched Network with Time Driven Priority.

5.6.1 Network Shaping Delay

When time driven priority is used in the network, the time cycle is set to T_c, the video frame period, and $B \cdot T_c$ bits must be reserved in each time cycle. The bits exiting the CBR MPEG encoder are buffered in the buffer of the packetization function depicted in Figure 72, until they can be included in a packet. Each bit experiences a different delay, but due to Requirement 3 of continuous playing, a network resynchronization delay must be introduced on the receiver side so that each bit experiences the same whole delay S_n^{Pack} which is called the *packetization* component of the network shaping delay.

The packetization function on the receiver side introduces the network resynchronization delay; the packetization function knows the delay experienced in the buffer of the sender packetization function by the bits of each received packet and can compensate it. The compensation is done by buffering the bits and providing to the decoder a bit flow at the constant rate B. If the TFs reserved to the videoconference call are evenly distributed in the time cycle, no network resynchronization is required.

The dimension of the buffer of both packetization functions must be at least $B \cdot S_n^{Pack}$. P_n^{Pack} is the maximum time a bit spends in the sender buffer before being retrieved and inserted in a packet; the buffer, which is filled at constant rate B, does not overflow if it can store the $B \cdot S_n^{Pack}$ bits generated by the decoder in the meantime. S_n^{Pack} is as well the maximum time spent by a bit in the receiver buffer; the buffer cannot contain more than $B \cdot S_n^{Pack}$ bits otherwise the last one would take more than S_n^{Pack} to exit at the constant rate B.

The actual value of S_n^{Pack} depends on how the TFs have been reserved within the time cycle for the videoconference; thus, it is fixed at resource reservation time and is constant over the whole videoconference call (or until the bandwidth is reallocated). If, for example, a single TF is reserved, $S_n^{Pack} = T_c$ because the first bit exiting the CBR encoder after a packet has been assembled and sent has to wait a whole time cycle before being inserted in a packet. Intuitively, the smaller and the more uniformly distributed inside the time cycle are the packets, the shorter the network shaping delay S_n^{Pack}.

Exploitation of small and uniformly distributed packets is encouraged also by other arguments. In Section 4.3.5 it has been shown that, if the decoder starts decoding as soon as some of the bits encoding a picture enter its buffer, the decoding time is reduced by using small packets. In [14] is shown that reserving a small amount of bits in many TFs is convenient also from the point of view of the network because balancing the load along the time cycle provides lower blocking probability. Using small packets has a further advantage when time driven priority is exploited in the transmission of CBR MPEG encoded video.

To identify the schedule that minimizes the network shaping delay, the structure of the MPEG stream must be taken into account. The MPEG standard specifies that the bit stream produced by a CBR MPEG encoder be structured as a *transport stream* (see Section B.5) in order to make more robust its transmission over packet switched networks. The transport stream is composed of packets having fixed dimension of 188 bytes; each packet contains bits encoding a single picture. The network shaping delay can be minimized if the TFs are reserved so that as soon as an MPEG packet is produced, a reserved TF is scheduled and the packet is sent[16]. This requires to reserve 188 bytes in TFs equally spaced by $(188 \cdot 8)/B$ and have the encoder synchronized with the network interface so that the last bit of each MPEG packet is emitted just before a reserved TF is scheduled[17]. If the encoder is not synchronized with the network interface, there is an application synchronization component of the network shaping delay $S_n^{AS} \in [0, (188 \cdot 8)/B]$.

In the general case, the network shaping delay is given by the composition of the two components, i.e.,

$$S_n = S_n^{AS} + S_n^{Pack}$$

and it is experienced partly in the sender and partly in the network resynchronization function of the receiver.

5.6.2 Excess Resynchronization Delay

In what has been discussed so far, the network resynchronization delay is assumed to be introduced by an ad hoc function which can be implemented, for example, in the software layers below the application. Nevertheless, it can also be introduced in the decoder buffer; buffering is not required in the receiver software below the decoder[18] and the decoder buffer must be enlarged. If the decoder is synchronized with the network interface, the buffer must be enlarged by, at least, $B \cdot S_n^{Pack}$. This stems from the fact that when time driven priority is exploited in the network, the packetization component of the network shaping delay is the only variable component of the network delay.

[16]Breaking an MPEG packet and sending it during two TFs decreases robustness to losses and errors.

[17]Actually, even in this situation, the bits of each packet has to be buffered in the packetization function until all the 188 bytes have been sent out of the encoder. However, this time does not contribute to the end-to-end delay because it is already taken into account by the coding shaping delay: the bits would be anyway kept in the decoder buffer to introduce the processing resynchronization delay. The only impact on the end-to-end delay is a possible increase of the decoding time if the decoder has an access unit smaller than the MPEG packet.

[18]Actually, a buffer is needed only to buffer each incoming packet while the header is processed, but this is not relevant to this work.

If the decoder is not synchronized with the network interface, when the first packet is received the decoder does not know how much time the bits have spent in the buffer of the sender packetization function. Thus, the decoder uses the conservative assumption that the bits have experienced the minimum delay and the decoding of pictures must be delayed accordingly[19]. If the assumption is true, the network shaping delay experienced by each bit partly in the sender packetization function, partly in the decoder buffer is actually S_n^{Pack} and the decoder buffer must be enlarged by, at least, $B \cdot S_n^{Pack}$. If the assumption is not true, a excess resynchronization delay E_r is introduced, i.e., bits are buffered longer than necessary; the decoder buffer must be dimensioned accordingly so that it doesn't overflow due to the excess buffering time E_r. In the following, the maximum value of E_r and the minimum dimension of the buffer are calculated.

The largest excess resynchronization delay is introduced when the first bit in the first received packet has already experienced a delay S_n^{Pack} in the buffer of the sender packetization function. The decoder should not introduce any network resynchronization delay; instead the decoding is delayed because it is assumed that the bits experienced minimum delay in the packetization function. In order to calculate the time spent by the first packet in the decoder buffer, it is necessary to compute the minimum delay experienced in the sender packetization function.

Bits experience minimum delay due to packetization, if a packet of dimension P_f is sent as soon as the CBR encoder emits the bit number P_f. In this case, the first bit in the packet had been in the buffer of the sender packetization function for the time the CBR encoder takes to produce P_f bits, i.e., P_f/B. Since decoding of the first packet must start after S_n^{Pack} from when the first bit was produced by the CBR encoder, the network resynchronization delay experienced in the decoder buffer (which is also the upper bound on the excess resynchronization delay E_r) is

$$S_n^{Pack} - \frac{P_f}{B} \tag{30}$$

The overall time spent by bits in the decoder buffer due to network resynchronization delay and excess resynchronization delay is thus given by

$$S_n^{Pack} + E_r$$

where

$$E_r \in \left[0, S_n^{Pack} - \frac{P_f}{B}\right]$$

[19]As explained in Section 3.4.1, the replay buffer shows the same behavior when introducing the network resynchronization delay to compensate the variation of the queueing delay in asynchronous packet switched networks.

The decoder buffer must be oversized according to the worst case, i.e., in order to introduce the network resynchronization delay and be robust against the excess resynchronization delay, it must be enlarged by

$$\left(2 \cdot S_n^{Pack} - \frac{P_f}{B}\right) \cdot B$$

The actual value of $S_n^{Pack} \in [0, T_c]$ depends on the TFs reserved for the videoconference call. Thus, S_n^{Pack} is known at reservation time by the network interface; this can be provided with a mechanism to communicate it to the decoder. If such a mechanism is not available, the decoder does not know the actual value of S_n^{Pack} and delays decoding according to the worst case $S_n^{Pack} = T_c$; i.e., decoding is started after

$$T_c - \frac{P_f}{B} \tag{31}$$

from the reception of the first packet. If the actual packetization component of the network shaping delay is null (because TFs had been allocated optimally) and the first packet has experienced S_n^{Pack} in the sender, it is unuseful to delay decoding by (31), i.e., this value is the excess resynchronization delay. In particular (31) is the maximum E_r value; the minimum excess delay is introduced when the bits in the first packet has experienced minimum delay and decoding should be delayed by (30):

$$E_r \in \left[T_c - S_n^{Pack}, T_c - \frac{P_f}{B}\right]$$

The decoder buffer must thus be enlarged by

$$\left(S_n^{Pack} + T_c - \frac{P_f}{B}\right) \cdot B$$

This shows that network resynchronization can be performed inside the decoder; this eliminates the need for an ad hoc buffer. Nevertheless, if the decoder is not tightly connected to the network interface (i.e., it does not have information about the schedule of the reserved TFs and the TF in which each packet is received), the excess resynchronization delay is larger.

5.6.3 End-to-end Delay

The general expression for the end-to-end delay is

$$\Delta_{CBR}^{TDP-SH} = S_c + S_s + L \cdot T_f + S_n + E_r + D + P_d \tag{32}$$

where

- S_c is the coding shaping delay introduced by the encoder/decoder system to produce the encoded stream at the constant target rate;
- S_s is the startup shaping delay introduced by the encoder to avoid the encoder buffer to overrun when the natural rate is smaller than the target rate;
- $L \cdot T_f$ is the time taken by a packet to travel through the network from source from destination,;
- $S_n = S_n^{Pack} + S_n^{AS}$:
  - $S_n^{Pack} \in [0, T_c]$ depends on the number, dimension, and position of the TFs reserved to the videoconference call and is constant over the duration of the call,
  - $S_n^{AS} \in [0, T_c]$ is due to the lack of synchronization between encoder and network interface; because of the relative drift of the two timing references, this delay can vary inside the given interval during the videoconference call;
- $E_r \in [0, T_c - P_f/B]$ is not null when the network resynchronization delay is introduced by the decoder and it is not synchronized with the network interface.

If resource reservation is optimal ($S_n^{Pack} = P_f/B$), the encoder is synchronized with the network interface ($S_n^{AS} = 0$), and network resynchronization is performed with information from the network interface ($E_r = 0$), (32) is very similar to (29). The terms $Sw + P$ appearing in the latter are substituted by the term $L \cdot T_f$ in the former; all of these terms depend on the propagation delay and the number of nodes on the path between sender and receiver. When the sender is topologically close to the receiver, the coding shaping delay S_c is the main contribution in both Δ_{CBR}^{CS} and Δ_{CBR}^{TDP}. For example, if S_c is around $3 \cdot T$ at 30 frames per second, and the propagation delay P and switching delay Sw are negligible, $\Delta_{CBR}^{CS} \simeq 100\ ms$. If a packet switched network with time driven priority is exploited in the same scenario, it is likely that packets are forwarded by one router, i.e., $L = 3$. Moreover, if small packets are used ($P_f = 188$ bytes) and the rate of the CBR encoder is 1.5 Mb/s,

$$\Delta_{CBR}^{TDP} = \Delta_{CBR}^{CS} + S_n^{Pack} + 3 \cdot T_f \simeq 100 + 1 + 0.375 \simeq \Delta_{CBR}^{CS}$$

Thus, the short distance transmission of CBR MPEG video over a packet switched network with time driven priority yields almost the same end-to-end delay as the transmission over a circuit switched network.

If sender and receiver are not close (i.e., the propagation delay is not negligible), the difference between Δ_{CBR}^{TDP} and Δ_{CBR}^{CS} depends on the speed and the number of intermediate nodes on the path. The faster the packet switches, the smaller the difference.

A suboptimal schedule of the TFs reserved for the videoconference call increases the end-to-end delay; nevertheless, suboptimal schedules can be accepted in order to reduce the probability that calls be blocked.

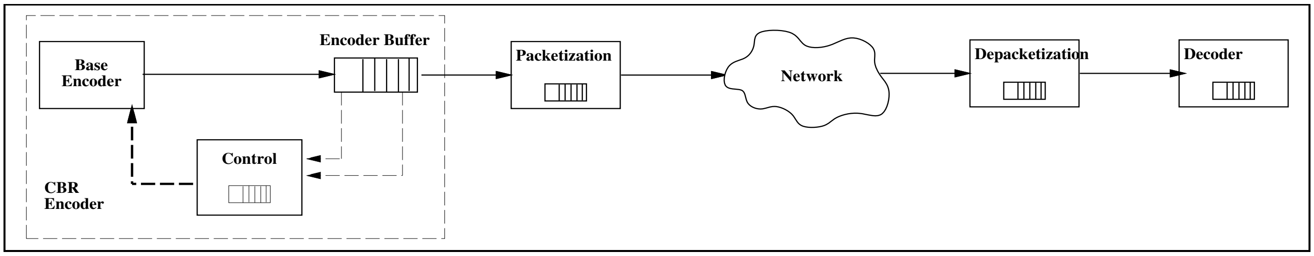

Figure 73: MPEG Encoder and Packetization Function.

5.7 Asynchronous Packet Switching

Transmission of a CBR MPEG stream over a packet switched network requires the stream to be segmented in packets and then reconstructed in the receiver. In order to better focus on the issues related to packetization, in this Section a configuration in which sender and receiver are connected by a direct link is first considered. Then, a multi-hop packet switched network between sender and receiver is taken into account.

5.7.1 Single-Hop Packet Network

With the system configuration depicted in Figure 73, sender and receiver are connected through a direct dedicated link on which they send data in packets, as shown by the packetization function in Figure 73. This configuration is not necessarily used in real applications, but allows us to focus on the delay due to packetization.

Assumption 5 *Transmission of a packet cannot start until all the bits that are to be put in its payload has been produced by the encoder.*

This Assumption conforms with reality since many existing protocols need to insert in the header of the packet a check sequence that can be calculated only from the whole packet payload. Nevertheless, due to this Assumption the packetization function introduces a delay in the system. The bits produced by the encoder are stored in the buffer of the sender packetization function. Given a packet size P_s, when the buffer contains P_s bits it is emptied (as shown in Figure 74) and a packet is sent over the network.

Since bits are produced by the CBR MPEG encoder at constant rate B, packets leave the sender at regular intervals P_s/B, as shown in Figure 74. The generated traffic has the same characteristics of data exiting a leaky bucket with token generation rate B equal to the target rate of the encoder and token bucket size $A = P_s$. Packet size must be chosen according to the structure of the MPEG transport stream so that a single MPEG packet is not sent into two packets. Moreover, we assume that a packet is sent each time the last bit encoding a picture exits the encoder, independently from its dimension:

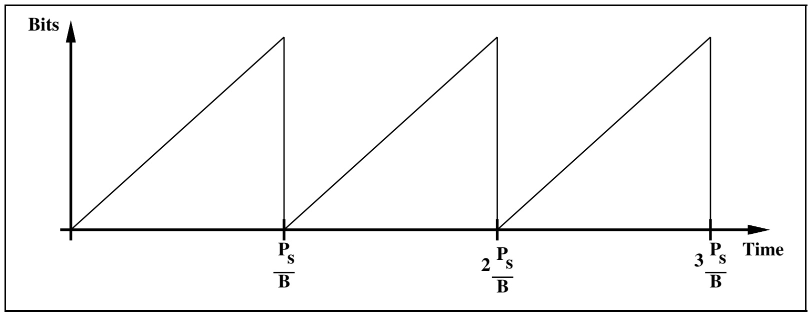

Figure 74: Fullness of the Buffer of the Packetization Function.

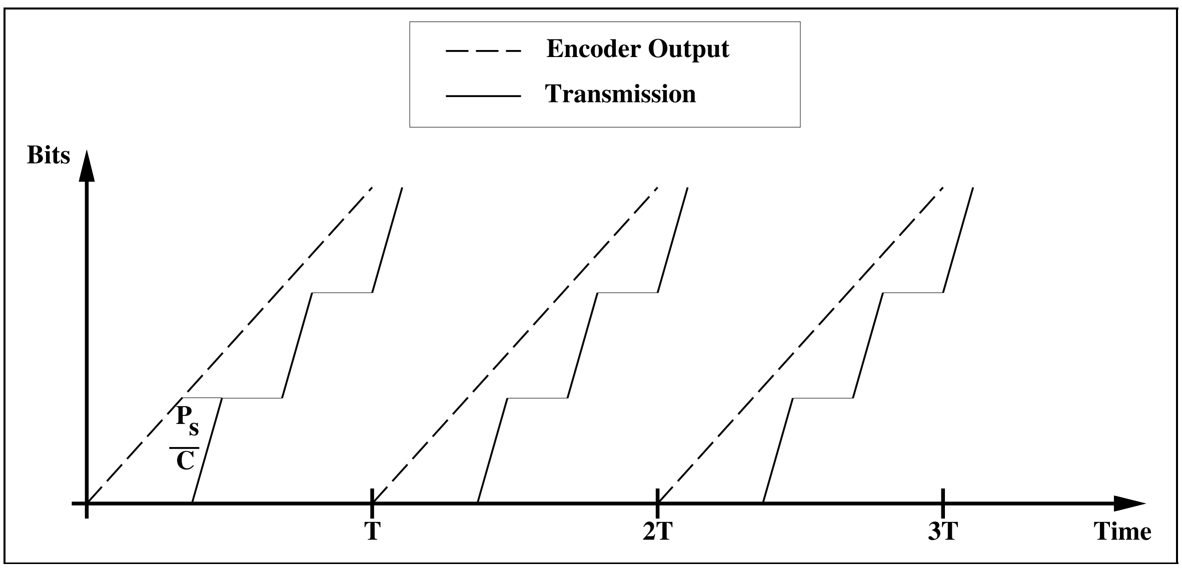

Figure 75: Generation and Transmission of Bits; Fullness Level of the Decoder Buffer.

Assumption 6 *A packet contains bits encoding only one picture.*

Thus, the real behavior of the packetization buffer can be different from the one depicted in Figure 74, but the basic principles discussed in this Section still hold.

As shown by Figure 75, packets are sent at the full speed C of the link connected to the sender. Each bit of a packet experiences a different delay from when it is generated by the CBR encoder to when they it is put on the link.

No Buffering in the Depacketization Function We first assume that packets get directly to the decoder buffer upon arrival at the receiver, as shown by Figure 76; the depacke-

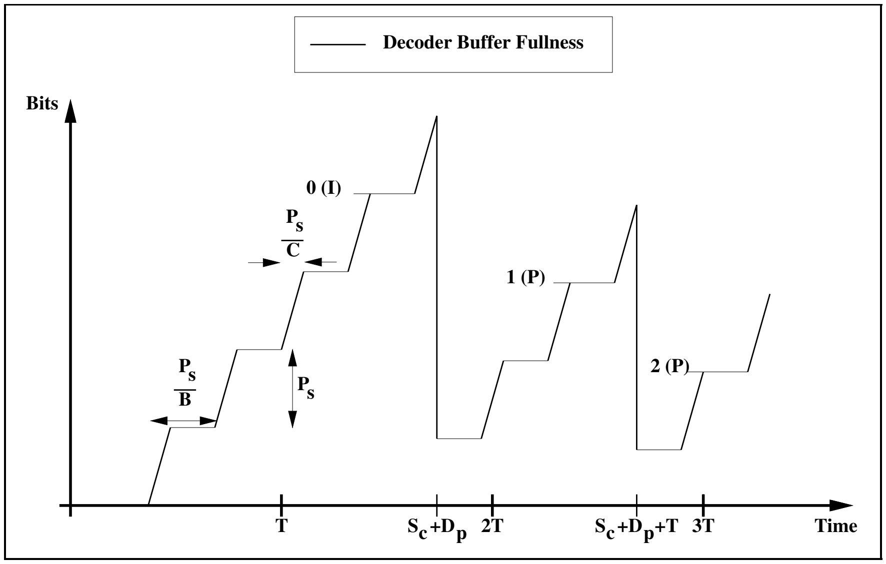

Figure 76: Fullness Level of the Decoder Buffer.

tization function has the only task of stripping off the packet header[20].

Each bit of a packet experiences a different delay due to packetization; nevertheless we are interested in the delay experienced by the whole picture. The system target decoder specified by the MPEG standard, the bits encoding a picture are removed from the decoder buffer all together when decoding starts. According to Assumption 6, the last bit of a picture is the last bit in a packet; in the worst case a time P_s/C elapses from when this last bit exits the CBR MPEG encoder to when it enters the decoder buffer. Thus, decoding is delayed by the *packetization delay* D_p that is the transmission delay of a packet, i.e.:

$$D_p = \frac{P_s}{C} \tag{33}$$

Maximum dimension packets experience the packetization delay entirely in the sender packetization function; packets of dimension $P < P_s$ spend $(P_s - P)/C$ in the decoder buffer. Since no further data arrive during this time and after it the whole picture is retrieved, the decoder buffer must not be oversized. Considering also the delay introduced by the encoder/decoder system (S_c, S_s, and D) and the propagation delay, the end-to-end delay is given by

$$\Delta_{CBR}^{Async} = S_c + S_s + D_p + P + D + P_d \tag{34}$$

The packet size shall be chosen as small as possible (possibly equal to the dimension of

[20]This assumption is not realistic as the packetization function usually needs to buffer the whole packet in order to possibly check the error control code. This is not relevant to what is explained in the following: it means that Figure 76 should be modified by having each packet enter the decoder buffer all at once.

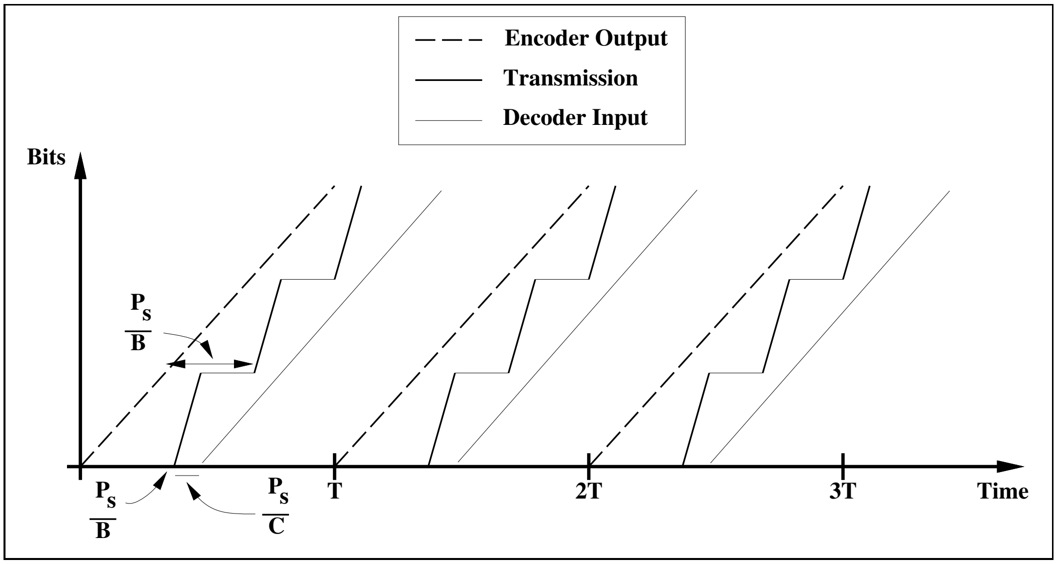

Figure 77: Generation and Transmission of Bits and Output of the Receiver Packetization Function.

MPEG packets) in order to decrease the packetization delay and possibly the decoding time if the decoder processes bits as soon as they enter its buffer.

Buffering in the Depacketization Function We assume now that the depacketization function in the receiver reconstructs the CBR MPEG stream, as shown in Figure 77, by buffering incoming packets and feeding the decoder buffer at the constant rate B. Received bits are feeded to the decoder only after the whole packet has been received. Thus, bits exiting the CBR MPEG encoder are buffered in the buffer of the packetization (sender) and depacketization (receiver) functions for an overall time

$$D_p = \frac{P_s}{B} + \frac{P_s}{C} \tag{35}$$

The packetization delay introduced in this system configuration is P_s/B larger than the one given by Equation (33). Thus, if the depacketization function reconstructs a CBR MPEG stream, the end-to-end delay is increased. The P_s/B increment is necessary if Assumption 6 is dropped and the last bit of a picture must not be the last of the packet in which it is sent. In fact, the sender packetization function delays the first bit of each packet by P_s/B, as shown in Figure 77; if decoding is not delayed by the same amount of time, the decoder buffer underflows when, for example, the last bit of a picture happens to be the first in a packet.

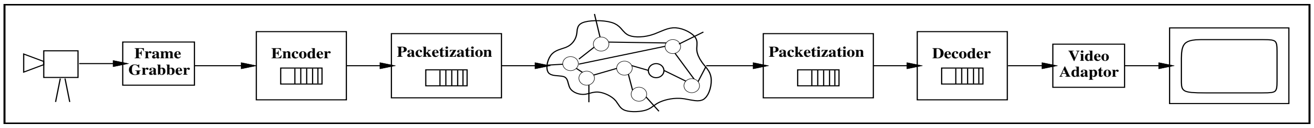

Figure 78: Architecture of a Videoconferencing System Exploiting a CBR MPEG Encoder and an Asynchronous Packet Switched Network.

5.7.2 Multi-Hop Configuration

Figure 78 shows the architecture of a videoconferencing system that exploits a CBR MPEG encoder and an asynchronous packet switched network. This configuration adds the queueing delay, the network resynchronization delay, and the excess resynchronization delay E_r to the end-to-end delay given by Equation (34):

$$\Delta_{CBR}^{Async} = S_c + S_s + D_p + P + Q_M + E_r + D + P_d \tag{36}$$

where P is the propagation delay on the links on the path from source to destination, Q_M is the maximum queueing delay experienced by packets in the network, $E_r \in [0, \Delta Q]$ is the excess resynchronization delay introduced by the replay buffer (being it in the packetization function or in the decoder) due to lack of synchronization between sender and receiver network interfaces (see Section 3.4.1), ΔQ is the variation of the queueing delay. The packetization delay D_p is given by Equation (33) or Equation (35) depending on the behavior of the de-packetization function (see Section 5.7.1).

Since bits are produced at the constant rate B and packets are sent at constant pace, the dimension of the replay buffer must be, at least,

$$2 \cdot \Delta Q \cdot B$$

This expression provides also the increment required in the dimension of the decoder buffer in case it is used to compensate the network delay variation.

5.7.3 Packetization and Startup Shaping Delay

The packetization function can impact the startup shaping delay. In many practical cases the mean bit production rate is higher than the target rate; thus the startup shaping delay is essential only while encoding the first MBs of a picture in order to compensate a possible instantaneous low bit production rate. If the packetization function is included in the closed control loop of the encoder, the startup shaping delay can be reduced or even eliminated. The control function takes into account that bits are not removed from the encoder buffer for a time $\frac{P_s}{B}$. Thus, even if the initial bit production rate is low, the average over this time interval can be sufficient to produce a packet worth of bits.

	Dedicated Link	Circuit Switching	Time Driven Priority	Asynchronous Packet Switching
Raw Video	$\Delta_{Raw}^{Ded} = \frac{F_r}{C} + P + P_d$	$\Delta_{Raw}^{CS} = S_n + P + Sw + P_d$	$\Delta_{Raw}^{TDP} = S_n + L \cdot T_f + P_d$ $S_n = S_n^{AS} + (N_r - 1) \cdot T_f$ $S_n^{AS} \in [0, T]$	$\Delta_{Raw}^{Async} = \frac{F_r}{C} + P + Q_M + E_r + P_d$ $E_r \in [0, \Delta Q]$ $\Delta_{Raw}^{Async-Sh} = S_n + \frac{P_s}{C} + P + Q_M + E_r + P_d$
VBR MPEG	$\Delta_{VBR}^{Ded} = C_M + P + D + P_d$	$\Delta_{VBR}^{CS} = C_M + Sw + P + D + P_d$ $\Delta_{VBR}^{CS-TS} = C_M + S_n^{CS} + P + Sw + D + P_d$	$\Delta_{VBR-I}^{TDP} = C_M + S_n^{AS} + L \cdot T_f + D + P_d$ $S_n^{AS} \in [0, T]$ $\Delta_{VBR}^{TDP-CxSc} = C_M + S_n^{Sched} + L \cdot T_f + D + P_d$ $S_n^{Sched} \in [0, N \cdot T]$	$\Delta_{VBR}^{Async} = C_M + \frac{P_s}{C} + P + Q_M + E_r + D + P_d$ $\Delta_{VBR}^{Async-TS} = C_M + S_n^{TS} + \frac{P_s}{C} + P + Q_M + E_r + D + P_d$
CBR MPEG	$\Delta_{CBR}^{Ded} = S_c + S_s + P + D + P_d$	$\Delta_{CBR}^{CS} = S_c + S_s + Sw + P + D + P_d$	$\Delta_{CBR}^{TDP} = S_c + S_s + L \cdot T_f + S_n + E_r + D + P_d$ $S_n = S_n^{Pack} + S_n^{AS}$ $S_n^{Pack} \in [0, T_c]$ $S_n^{AS} \in [0, T_c]$ $E_r \in [0, T_c - P_f / B]$	$\Delta_{CBR}^{Async} = S_c + S_s + D_p + P + Q_M + E_r + D + P_d$ $D_p = \frac{P_s}{B} + \frac{P_s}{C}$

Table 2: Summary of the Configurations Considered in this Work.

6 Summary

In this work we analyzed the end-to-end delay of videoconferencing over packet switched networks. Our key findings are summarized in Table 2. The main design objective is to keep the end-to-end delay below 100 ms. This requirement comes from the human hearing sensitivity for delays larger than 100 ms and the requirement for lip-synchronization, i.e., the need for the audio and video to be synchronized.

Controlling the delay also requires to control the amount of buffers used at the (i) sender, (ii) network and (iii) receiver. The control of the buffer sizes can have adverse consequences:

1. Decreasing the buffers inside the network can increase the packet loss inside the network and degrade the quality of the received video.

2. Limiting the MPEG encoder buffer size will limit the maximum I-frame size and degrade the compressed MPEG video quality.

Thus, what we found and formulated, in this study, are some tradeoff between the perception quality due to delay and the received picture quality due to loss and compression.

We found interesting results some of which are counter intuitive. This in turn illustrates the importance of this sort of study.

1. Transmission of raw video does not necessarily provide the shorter delay. This is because of the transmission time needed for large number of bits of high definition pictures.

2. MPEG CBR encoding of a fixed scene introduces long delay. This is because the inter-frame coding of P-frames requires few bits, and therefore, a single I-frame can use all the CBR capacity allocated to the group of picture (GOP). As a result, the I-frame transmission will last the entire GOP period.

3. Using asynchronous packet switching with statistical multiplexing is challenging. This is because the distribution of the delay variation or jitter inside the network can be large and with heavy tail. Thus, conservative design with large replay buffer would result in large delay, and small replay buffer would result in occasional underflow/overflow of the replay buffer and distortion of the video viewed by the user at the receiver.

4. If the capture card and the display buffer are using the same reference clock the delay can be decreased by one video frame period. For 10 frames per second this means delay reduction of 100 ms and for 30 frames per second the delay reduction is 33 ms.

5. MPEG VBR video over asynchronous packet switched network requires high equivalent capacity or effective bandwidth, which can be too much to be wasted over wide area links. Therefore, in some case it may be better to use only I-frames, i.e., to use for example Motion-JPEG.

In conclusion, MPEG-based videoconferencing is possible, however, in order to keep the end-to-end delay below human perception, there are four requirements:

1. The capture (frame grabber) card and display (video) adapter should have a common time reference clock.
2. The pictures should be sent right after the compression is completed as a variable bit rate (VBR) stream.
3. The network jitter should be controlled with a well defined bound.

We also showed that with time driven priority and complex scheduling it is possible to have the following properties for VBR MPEG:

1. Bound of 250 μseconds on the jitter.
2. No loss even if the link is fully utilized.
3. The end-to-end delay is dominated by the propagation delay plus $L \cdot 125\mu$seconds (L depends on the number of hops). This delay is shorter than the delay that can be obtained over a circuit switched network. On a circuit switched connection a CBR encoder must be deployed which introduces a coding shaping delay larger than the video frame period. If the scene is slow, this delay can be as large as the duration of a group of pictures.

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.

A List of Achronyms and Symbols

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

b_{mb} macroblock level rate control factor of macroblock mb.

B target rate for MPEG CBR encoder or rate of circuit switched connections used to send raw and encoded video.

C capacity of physical links.

C_M maximum coding delay.

CBR Constant Bit Rate.

$c_{i,j}^{mb}$ DCT coefficient (i, j) of MB mb.

$\hat{c}_{i,j}^{mb}$ quantized DCT coefficient (i, j) of MB mb.

D decoding time or decoder delay.

DCT Discrete Cosine Transform.

D_i^T delay experienced in the network by frame i

D_M^T maximum delay experienced by a frame in the network.

D_m^T minimum delay experienced by a frame in the network.

D_i^R delay experienced in the replay buffer or transmission synchronization delay experienced by video frame i.

D_M^R maximum delay experienced in the replay buffer or maximum transmission synchronization delay.

E_r Excess resynchronization delay

F number of bits encoding a frame;

F_m lower bound on picture dimension,

F_r number of bits encoding a raw picture,

F^I dimension of an I-frame,

F^P dimension of a P-frame.

G global distortion level that is chosen on a picture by picture basis

GOP Group Of Pictures.

J_n network delay jitter.

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

M number of video frame periods in the time cycle.

MB MacroBlock.

N number of video frames in a group of pictures.

N_s number of time frames in which bits have been allocated to send a raw or encoded video frame with time driven priority.

N_r number of time frames between the first and the last TF reserved for the transmission of a picture.

P propagation delay.

p_{mb} MB local activity factor of MB mb.

P_s size of a packet.

Q queueing delay in the network;

- Q_m minimum queueing delay,
- Q_M maximum queueing delay.

Q_{mb} quantization parameter of macroblock mb.

QoS Quality Of Service.

$q_{i,j}^{\{I|P\}}$ element (i,j) of quantization matrix for I-frames or P-frames.

S_c coding shaping delay.

S_n network shaping delay.

S_n^{CS} network shaping delay due to lack of synchronization between application and network interface.

S_n^{CS} network shaping delay due to circuit bandwidth.

S_n^{TS} network shaping delay due to traffic shaping at the boundaries of asynchronous packet switched networks.

S_s startup shaping delay;

- S_s^m minimum startup shaping delay.

Sw switching delay.

T video frame period.

T_c time cycle duration.

TF Time Frame.

T_f duration of a time frame.

VBV Video Buffer Verifier.

V_b maximum backlog in the encoder buffer.

V_s size of the VBV.

B The MPEG Encoding Standard

A video sequence is digitalized by converting into digital form pictures (or video frames) at a fixed pace. Digital video is replayed by displaying pictures at the same fixed pace at which they had been captured. For example, good quality video is digitalized and displayed at 30 frames per second in the US and 25 frames per second in Europe.

Each picture is divided in pixels; the number of pixels in each picture and the number of bits used to encode each pixel determines the picture resolution. Pictures are described by three components which are either the intensity of the red, green, and blue (RGB format) or the intensity of luminance and chrominance (YUV format) of the pixels.

This coding format requires a huge amount of bits to encode sequences. Many compression schemes have been proposed which exploit the redundancy inside (*spatial redundancy*) and among (*temporal redundancy*) pictures in order to reduce the number of bits needed to encode a video sequence. The Moving Pictures Expert Group coding standard [11, 25, 8] specifies how to digitally encode and compress video and related audio mainly for two application fields: (1) storage and subsequent retrieval and display (e.g., on digital video disks), and (2) one way broadcasting (e.g., TV broadcasting and video on demand).

Nevertheless, the MPEG standard has been proposed for being applied also for videoconferencing purposes, even though the encoding algorithm is computationally intensive and it was not designed for real-time operation. The MPEG standard supports various resolutions and output bit rates through the same coding principles.

This Appendix describes the basic principles of MPEG standard with the aim of highlighting and explaining the characteristics which are particularly relevant to this work.

B.1 General Principles

Compression schemes are based on reduction of temporal and spatial redundancy and quantization of values used to code images. While redundancy reduction is a lossless process, quantization degrades image quality.

Temporal redundancy is eliminated by coding the difference between the picture and a reference one. This introduces strong dependency among pictures and pictures coded by difference cannot be decoded and displayed prior to decoding the reference image. This turns

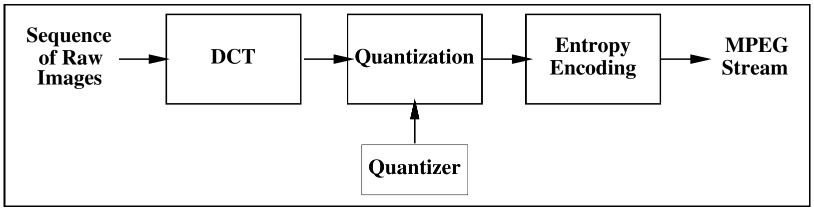

Figure 79: Block Diagram of the MPEG Encoding Algorithm for I-Frames.

out to be a problem when transmitting a coded sequence because the loss of (part of) a reference picture causes all the referencing images to be corrupted.

The MPEG standard defines three categories of pictures according to whether spatial or temporal redundancy is eliminated:

- I-frames (*Intra-coded* frames) are coded by eliminating spatial redundancy only. Hence they are decoded by themselves, without reference to any other picture.
- P-frames (*Predictive-coded* frames) are encoded with reference to the previous image.
- B-frames (*Bidirectionally predictive-coded* frames) are coded with reference to the previous and following I-frame or P-frame. This allows similarities with a future image (possibly not present within a previous picture) to be exploited to reduce temporal redundancy. This coding yields very good compression, but introduces a large delay because a B-frame cannot be coded until the future reference picture has been captured; the lower the frame rate, the larger the delay introduced. Since in this work we are considering interactive applications which require low delays, bidirectional predictive coding is not exploited and further explained.

When a video sequence is encoded, images are gathered in *Groups of Pictures* (GOPs): the first image is intra-coded and the following pictures in the same GOP are predictive-coded[21].

B.2 Intra-coded Pictures

Intra-frame coding yields compression by eliminating spatial redundancy. In this Section the basic steps of intra-coding are briefly described.

B.2.1 Discrete Cosine Transformation

Raw pictures which are to be encoded with the MPEG standard are digitalized in format similar to YUV. Each image consists of a luminance component and two chrominance components. The former has twice as many pixels as the latter two components.

[21] In general, the GOP contains a pattern of B-frames interleaved by P-frames, e.g., IBBPBBPBBPBB.

On each of the three components of a picture 8x8 blocks of pixels (named simply *blocks* in the following) are identified. The Discrete Cosine Transformation (DCT) is performed independently on each block and the representation of the block in the frequency domain is obtained. The DCT takes as input 64 8-bit integer values $s_{x,y}$ representing the value of each pixel in the block and produces 64 12-bit integers $c_{i,j}$ which are the representation of the block in the frequency domain [8]:

$$c_{i,j} = DCT(s_{x,y}) \qquad i, j, x, y \in [0, 7]$$

The $c_{0,0}$ coefficient is called DC-coefficient and represents the fundamental color in the block. The other coefficients, AC-coefficients, represent the separation of lines in either or both directions. E.g., $c_{7,0}$ is the highest frequency appearing in the vertical direction and represents the closest separation of vertical lines in the block. $c_{i,i}$ is proportional to the amount of squares in the block because they are characterized by the same spatial frequency in both directions. E.g., the larger the number of isolated pixels, the highest the value of $c_{7,7}$.

The DCT provides *energy compaction* because the information contained in a block is grouped according to the level of detail in the block. Low spatial frequency coefficients contain the information on the overall aspect of the block; high frequency coefficients contain the information concerning the details of the block. Low frequency coefficients usually have larger values than high frequency ones. The encoding step (see Section B.2.3) provides better compression as the values to be encoded are not uniformly distributed.

Moreover, frequency domain coefficients can be coded with different precision according to how sensitive the human eye is to each of them. Low frequency ones are more important in determining the perceived quality of an image, i.e., they must be encoded with minimal information loss (fine quantization). High frequency components are less critical for the image quality, and thus they can be encoded with less accuracy and higher compression ratio (coarser quantization).

B.2.2 Quantization

Each DCT coefficient is quantized with a different quantizer because each one of them weighs differently on the image quality. The quantization is driven by an 8x8 matrix of 8-bit integers called *quantization matrix*. Each element of the matrix indicates the aggressiveness of quantization on the corresponding DCT coefficient: quantization is performed by dividing the elements of the block by the corresponding element of the quantization matrix. Thus, the higher is the value of quantization elements, the coarser is the quantization, i.e., the greater is the yielded compression. The result of this step is a 10-bit integer value [8].

The quantization matrix is chosen according to the nature of the scene to be encoded and is included at the beginning of the MPEG video stream.

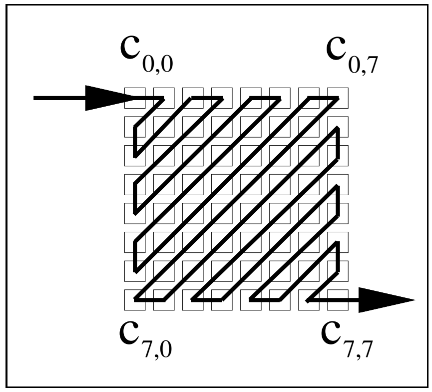

Figure 80: Encoding Order for Quantized DCT Coefficients.

B.2.3 Entropy Encoding

This step produces a compressed stream of bits through two sub steps [7, 27]:

1. *Run-length.* The coefficients obtained by the quantization are run-length encoded with a sequence of couples of symbols:

   - the first symbol, is divided in two parts:
     (a) `run-length` (4 bits) represents the number of coefficients with value zero,
     (b) `size` (4 bits) is the number of bits (between 0 and 10) used to encode the value of the next non zero coefficient;
   - the second symbol is the coefficient value expressed on `size` bits; the value zero does not have to be coded as it is run-length encoded.

   As a consequence, for each non-zero coefficient a number of bits between 8 to 18 is used.

   The DC-coefficient $c_{0,0}$ is encoded using a predictive coding technique, i.e., the difference between the value of the DC-coefficient of the current block and the one of the previous one is encoded. As the two quantized coefficients are both represented on 10 bits, the result is encoded on a maximum of 11 bits.

   The AC-coefficients are encoded according to the zig-zag order shown in Figure 1, thus being considered in increasing frequency order. Low frequency coefficients typically have similar big values, while high frequency ones have smaller values; the probability of zero valued coefficients is higher among the high frequency ones and having them grouped together allows run-length encoding to deliver better compression.

2. *Huffman encoding.* The intermediate symbol sequence produced by the run-length encoder is transformed by replacing the first symbol in each couple of symbols with a

variable length code obtained from a Huffman table that has been specified for each component of the picture.

B.2.4 Controlling I-frame Dimension

As shown in Section 4.1 (see Figure 20) dimension of encoded pictures is highly variable. Application of the MPEG encoding scheme often requires picture dimension to be controlled: e.g., CBR MPEG encoding (Section 5) and transmission of VBR MPEG encoding through packet switched networks with time driven priority Section 4.3). The number of bits obtained when intra-coding a picture (I-frame dimension) depends on the following factors:

- *Resolution* of the digitalized image: it is defined for each conference and it is not usually changed on the fly during the session.
- *Content* of the digitalized image: if the image has few details, many high frequency DCT coefficients have small values. Quantization reduces small values to zero and run-length encoding delivers high compression.
- *Quantizer*: the higher the quantization stepsize, the smaller the values of the quantized coefficients. Smaller quantized values are Huffman encoded with fewer bits; sequences of zero values yield even higher compression thanks to run-length encoding.
- *Variance of the first symbols* produced by run-length encoding: Huffman encoding is as more effective as the number of different first symbols in the sequence is low and each symbol has a high occurrence probability.

Among the foregoing factors, only the quantizer can be changed on purpose and dynamically to control the compression ratio and hence I-frame dimension. The quantization stepsize can be possibly modified on a block by block basis. Since a quantization matrix is exploited, quantization is regulated separately for each coefficient in a block, thus allowing a fine tuning over a wide range of compression ratios (and image qualities).

One of the most debated topics in controlling I-frames dimension is the choice of a quantizer that allows the intended number of bits to be produced, while keeping image quality uniform. The problem is sometimes addressed with reiterative approaches that quantize a picture more than once with different quantizers in order to identify the most suitable quantization stepsize. At each iteration the picture is encoded; if the yielded amount of bits is too big (small), all the coefficients are re-quantized with a coarser (finer) step and re-encoded. After a number of iterations the intended dimension is obtained for the I-frame.

This approach is computationally hard and it does not bound the encoding time, thus not being suited to real-time encoding. Some heuristics can be used to predict the amount of bits that are going to be produced by compressing an I-frame without having to perform the coding process up to the end.

1. The DCT is performed on the whole picture before starting quantizing and encoding the coefficients. By a swift analysis of the yielded coefficients the compression factor provided by Huffman encoding can be roughly estimated and a sensible quantization stepsize can be chosen.

2. During quantization some very simple and computationally light algorithm is run in order to get an estimate of the amount of bits expected as outcome of the encoding step. If the I-frame is expected to grow too large, the coefficients are quantized again with a larger quantization stepsize before encoding the outcome of quantization. If the system has more than one processor, more than one quantization processes, each with different stepsize, can be run in parallel.

3. Blocks are grouped in sets and an equal share of the target I-frame dimension is assigned to each set. After all the blocks in a set have been processed, the yielded number of bits is compared against the assigned share. If it is significantly grater (smaller) than the target, the coefficients are re-quantized with a larger (smaller) quantizer and re-encoded.

   Block grouping must be pseudo-random because an image can have very simple parts whose blocks are encoded with a small amount of bits; more complex parts are characterized by blocks requiring a huge amount of bits. Due to the pseudo-random order with which blocks are encoded, the image sending cannot start before the whole picture has been encoded.

The proposed heuristics can be used singularly or in composition to get better results. Constraints must be imposed on the variability of the quantization stepsize within each image in order to prevent non-uniform quality.

B.3 Predictive-coded Pictures

Predictive-coding reduces both spatial and temporal redundancy. Pictures are encoded by reference to the previous one (having it been intra- or predictive-coded) which is called the *reference picture* in the following. This Section describes the basic steps of predictive-coding; the first two allow temporal redundancy to be reduced before going through the very steps of intra-coding.

B.3.1 Motion Estimation

On the luminance component of the digital image, 16x16-pixel squares are identified, each composed of 4 blocks; one block on each of the chrominance components corresponds to the 16x16-pixel square on the luminance component. Motion estimation operates on *macroblocks* (MBs) which consist of 6 corresponding blocks (4 on the luminance component and one on the two chrominance components). The reference picture is searched for a MB "similar" to the one being encoded; the possibly found MB is called a *predictor*. The difference between the given MB and the predictor is encoded as a representation of the former.

An algorithm, not specified by the MPEG standard, is run in order to identify a predictor in the area of the reference picture around the location corresponding to the MB being encoded. The algorithm behavior is controlled by two parameters which determine its performance in terms of running time and contribution to the overall picture compression.

1. The *search range* identifies the area around the current MB position in which the reference picture is searched for the predictor. The larger the search area, the higher the probability of finding a very well matching MB, thus yielding good compression. On the other hand, the larger is the search area, the longer it takes to the algorithm to complete the search. The search range can be changed on a picture by picture basis.

2. The *similarity criteria* aims at choosing as predictor the MB which is going to provide the highest compression when the difference from the actual MB is encoded. The more complex the similarity criteria, the better the compression obtained, but the harder the computation.

Motion estimation is the most computationally intensive and time consuming step of the whole encoding process.

B.3.2 Motion Compensation

When a predictor is found for the MB, the difference between each pixel and the corresponding one in the predictor is computed. Since two 8-bit integers are subtracted, the result is a 9-bit integer. If a MB similar enough is not found, each block of the MB is encoded as an I-frame block, i.e., the motion compensation step is skipped.

The complete encoding of a motion compensated MB encompasses also a motion representation which allows the predictor to be identified. Motion is represented through a *motion vector*, i.e., the bidimensional offset of the predictor from the position in the picture of the MB being encoded. The motion vector is encoded as difference from the motion vector of the previous MB.

B.3.3 DCT

The DCT is performed on the output of the previous step. If a predictor has been found, the DCT is performed on the differences between the pixels of the given MB and those of the predictor. If no predictor has been found, the DCT is applied on the pixels of the MB itself.

B.3.4 Quantization

The result of the DCT is quantized using different quantizers for motion compensated MBs and intra-coded MBs. The motion compensated MBs likely have small coefficients and thus the quantizer must have no deadzone, i.e., the range of values that are quantized to zero must be smaller than the quantization stepsize.

If the coefficients of a motion compensated MB are all zero (i.e., a predictor identical to the MB has been found) the MB is no further processed and it is represented in the MPEG stream by a special 6-bit code.

B.3.5 Entropy Encoding

All the coefficients of motion compensated MBs are encoded using run-length encoding and Huffman encoding (i.e., the DC-coefficient is not treated differently). Blocks of non motion compensated MBs are encoded according to the intra-coding process.

B.3.6 Controlling P-frame Dimension

The number of bits obtained when predictive-coding a picture (P-frame dimension) depends on the following factors:

- *Dynamics of scenes*: if scenes are static, subsequent pictures do not change too much, motion compensation is highly effective, and P-frames are small.
- *Resolution of images*: see Section B.2.4.
- *Search range* and *similarity criteria* used to find the predictor. Both parameters could be used to control the compression ratio, but they are employed too early in the compression process. Thus, on one side, reiterations with different values would be impracticable due to the large amount of computation required by motion estimation. On the other side, the relationship between changing these parameters and the number of generated bits is not trivial.
- *Quantizer*: see Section B.2.4.
- *Variance of the first symbols* produced by run-length encoding: see section B.2.4.

The same actions proposed in Section B.2.4 for controlling the size of I-frames, can be taken for P-frames. In addition, before encoding a P-frame, the percentage of MBs which can be motion compensated can be roughly estimated by estimating the overall similarity of the two images. Having some knowledge or statistics about the number of bits needed to encode motion compensated MBs, it is possible to roughly estimate the amount of bits that are going to be produced and consequently choose the search range and the quantizers to be used.

B.4 Signal to Noise Ratio

The loss due to the encoding process can be quantitatively measured by the *peak-to-peak Signal to Noise Ratio* (SNR) which is calculated as

$$SNR = 10 \cdot \log_{10} \frac{255^2}{\sum_i (s_i - \hat{s}_i)^2} \tag{37}$$

where s_i is the value of the i^{th} pixel before encoding, while $\hat{s}_i$ is the value of the same pixel reconstructed from the MPEG stream. The signal to noise ratio is scaled according to the maximum value of the pixel (peak-to-peak) which is 255.

B.5 Packetization and Streaming

The MPEG standard [11] defines an *elementary stream* for carrying each media: video, audio, or data. A *system stream* is designed to combine a number of elementary streams.

Each elementary stream is packetized; each *packet* has a header identifying the elementary stream to which the packet belongs. Moreover, the header can contain a *presentation time stamp* and a *decoding time stamp* which indicate the time at which the first access unit[22] in the packet is intended to be presented and decoded, respectively. The compliance of an elementary stream with the MPEG standard is checked by feeding a *System Target Decoder* (STD) with the stream and verifying that its buffer does not overflow or underflow.

The STD buffer is filled by the packets of the corresponding elementary stream as soon as they get to the decoder. The buffer is emptied by retrieving a whole access unit at the time specified by the decoding time stamp contained in the packet header. The STD buffer underflows if the complete access unit is not yet in the buffer before its retrieval is scheduled. The buffer overflows if there is not enough space to store a whole packet when it is extracted from the system stream.

Packets, possibly belonging to different streams, are grouped in *packs*, each preceded by a pack header which contains information about the rate of the stream. Moreover, the pack header contains a time stamp, called the *system clock reference*, which indicates the intended time of arrival of the last byte of the time stamp itself. It is used to recover the encoder clock at the decoder.

The first pack of the system stream contains a system header that can be optionally repeated in some other packs. The system header carries information that is relevant to the whole stream and contains the maximum size of the STD buffer to be used with each elementary stream. The current size of the STD buffer optionally included in packet headers must be smaller than the maximum value present in the system header.

The MPEG-2 standard provides two kinds of system stream:

[22] For example, the the access unit in a video stream is a picture.

1. The *Program Stream* is designed for error-free environments, e.g., storage and retrieval of video scenes from digital video disks. It uses variable length packets.

2. The *Transport Stream* is intended for exploitation in lossy environments (e.g., transmission over computer networks) and packets have fixed length (188 bytes).

References

[1] M. Baldi, Y. Ofek, and B. Yener. Adaptive real time group multicast. Submitted to IEEE INFOCOM '97.

[2] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the 'leaky bucket' policing mechanism in ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):355 – 342, April 1991.

[3] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *ACM Computer Communication Review (SIGCOMM'89)*, pages 3–12, 1989.

[4] W. Ding and B. Liu. Joint encoder and channel rate control of VBR video over ATM networks. In *Visual Communications and Image Processing*, volume 2668, pages 392 – 407, 1996.

[5] 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*, 6(1):12 – 20, February 1996.

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

[7] B. Fufht. A survey of multimedia compression techniques and standards. Part I: JPEG standard. *Real Time Imaging*, (1):49 – 67, 1995.

[8] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):47 – 58, April 1991.

[9] Moving Pictures Experts Group. *MPEG-2 test model 5.* ISO/IEC JTC1/SC29/WG11/N0400, April 1993.

[10] C. Horne and A. Puri. Video coding with adaptive quantization and rate control. In *Visual Communications and Image Processing*, volume 1818, pages 798 – 806, 1992.

[11] 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.

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

[13] G. Karlsson. Asynchronous transfer of video. *IEEE Communications Magazine*, pages 118 – 126, August 1996.

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

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

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

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

[18] W. Luo and M. El Zarki. Adaptive data partitioning for MPEG-2 video transmission over ATM based networks. In *International Conference on Image Processing*, volume 1, pages 17 – 20, October 1995.

[19] P. Pancha and M. El Zarki. MPEG coding for variable bit rate video transmission. *IEEE Communications Magazine*, pages 54 – 66, May 1994.

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

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

[22] 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*, 9(3):305 – 313, April 1991.

[23] 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*, volume 1903, pages 246 – 254, 1993.

[24] 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*, pages 1920–1924, 1995.

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

[26] 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*, volume 2668, pages 181 – 188, 1996.

[27] G. K. Wallace. The JPEG still pictures compression standard. *Communications of the ACM*, 34(4):30 – 44, April 1991.

[28] L. Wang. Rate control for MPEG video coding. In *Visual Communications and Image Processing*, volume 2501, pages 53 – 63, 1995.