ELSEVIER

Signal Processing: *Image Communication* 15 (1999) 35–56

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Packet loss resilient MPEG-4 compliant video coding for the Internet

F. Le Léannec*, F. Toutain, C. Guillemot

INRIA/IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

Abstract

Targeting multimedia communications over the Internet, this paper describes a set of complementary techniques in the direction of both improved packet loss resiliency of video-compressed streams and efficient usage of available network resources. Aiming first at a best trade-off between compression efficiency and packet loss resiliency, a procedure for adapting the video coding modes to varying network characteristics is introduced. The coding mode selection is based on a rate-distortion procedure with global distortion metrics incorporating channel characteristics under the form of a two-state Markov model. This procedure has been incorporated in an MPEG-4 video encoder. It has been observed that, in error-free environments, the channel adaptive mode selection technique does not bring any penalty in terms of compression, with respect to the initial MPEG-4 encoder, while allowing a significant gain with respect to simple conditional replenishment. On the other hand, under the same loss conditions, it is shown that this procedure significantly improves the encoder's performance with respect to the original MPEG-4 encoder, to approach the robustness of conditional replenishment mechanisms. This intrinsic robustification of the encoder allows to minimize the effects of packet losses on the visual quality of the received video; however, it does not avoid losses. A rate-based flow control mechanism is then developed and introduced into the encoder, in order to match the bandwidth requirements of the source to the bandwidth available over the path of the connection, for both 'social' and 'individual' benefits. The control mechanism developed combines an RTT-based control loop allowing early reaction to congestion and a TCP-friendly rate prediction model getting into play under lossy conditions. This hybrid control mechanism allows full rate control (even in loss-free conditions) and smooth rate variations together with high responsiveness. The introduction of the rate control in the MPEG-4 compliant encoder allows to maintain a stable PSNR and visual quality while decreasing significantly the source throughput, hence reducing congestion and loss provoked by the same video source at a constant bit-rate.

1. Introduction

Multimedia communication within the current best-effort Internet faces well-known challenges with respect to quality of service, congestion management, and network friendliness. Due to the real-time nature of envisioned data streams, multimedia delivery usually makes use of the so-called unresponsive transport protocols, i.e. the User Datagram Protocol (UDP) and/or Real-time Transport Protocol (RTP). Both UDP and RTP offer no quality of service control mechanisms and can therefore not guarantee any level of QoS,

* Corresponding author. Tel.: + 33-299-842-543; fax: + 33-299-847-171.

E-mail address: fabrice.le_leannec@irisa.fr (F. Le Léannec)

0923-5965/99/$ - see front matter
PII: S0923-5965(99)00023-5

despite the companion protocol Real-time Transport Control Protocol (RTCP). RTP is indeed somehow an empty shell for multimedia data bits, with respect to traditional transport features, e.g. flow control or reliability. Hence, multimedia communication relying on a best-effort network service, as provided by today's Internet, has to face varying network QoS characteristics, in terms of delay and packet losses.

Traditional video compression algorithms, relying widely on differential, run-length and variable length coding, are very sensitive to packet losses. Losses can spread within a single picture up to a given resynchronization point, or even across several pictures when using temporal prediction, hence have different impacts on the quality of service, from decoder no-start, to a whole range of quality impairments. Various approaches aiming at improved resiliency against packet losses of video streams have emerged recently. Targeting intrinsically error resilient streams, robust variable length codes such as reversible VLC [33], or mechanisms for limiting loss propagations are introduced. Error propagation in the decoded stream is limited by incorporating in the stream syntactic descriptors like synchronization or data partitioning markers. Restrictions in terms of prediction window size are introduced, in order to confine all spatially predictively encoded information within a single video packet, delimited by resynchronization markers [10]. Similarly, in order to avoid temporal loss propagation, videoconferencing tools like nv and vic, currently used on the MBone, do not support temporal prediction but rely only on conditional replenishment [18]. Experiments reported here, and based on a simplified MPEG-4 encoder where temporal prediction is replaced by conditional replenishment, show the compression loss traded for the increased packet loss robustness. These experiments also show that error resilient mechanisms such as those supported by the MPEG-4 video verification model may not be sufficient under high loss condition. This paper proposes several solutions while remaining fully compliant to the MPEG-4 video syntax.

The first issue addressed here is therefore a better trade-off between error robustness and compression efficiency, while limiting both temporal and spatial propagation. First attempts for maintaining temporal prediction in an error-prone environment are considered in [9]. The channel is modelled by a Bernoulli process and the intra/inter coding modes selection relies on a Viterbi algorithm. However, best-effort Internet is better modelled by finite state erasure channels exemplified by the Elliott–Gilbert channel [1]. A new coding mode selection strategy aiming at tailoring intra/inter modes to channel characteristics is introduced. The procedure relies on global distortion metrics incorporating an Elliott–Gilbert process for channel modelling.

At the transport level, error control mechanisms, such as forward error correction (FEC), automatic repeat request (ARQ), or hybrid ARQ/FEC repeat request (ARQ), can be also considered. Error control mechanisms increase stream resiliency to packet loss, at the expense of increased bandwidth, but do not avoid packet loss. They are considered here for protecting high-priority information such as, for example, visual sequence or video object planes headers transporting decoder configuration parameters.

Complementary approaches, such as congestion and rate control, aim at minimizing the amount of packet loss by matching the video bandwidth requirement to the available network capacity. The basic principle behind unicast (or point-to-point) congestion control is an adaptive process involving a source and a receiver for controlling the source's throughput. By monitoring the network state, the source–receiver pair can detect incipient congestion and react by lowering the output rate. Conversely, an unloaded state triggers a rate increase so as to better use the available network resources. Quantities that are usually monitored include packet loss and round-trip time (RTT) delay. Schemes making use of additive increase/multiplicative decrease rate control are commonplace in the literature, the best known instance of this being the TCP protocol. The resulting aggregate behaviour of such schemes is ideally one in which the network utilization is kept high and the loss rate low. In addition, a new session coming into play may expect to get more or less fair a share of the network bandwidth.

However additive increase/multiplicative decrease schemes typically give rise to so-called sawtooth rate patterns. The QoS requirements of a multimedia stream may be in sharp contrast, as smooth rate variations are often a prerequisite for maintaining acceptable video quality. Furthermore, end-to-end delay variations resulting from network queues building up in time of congestion have a greater impact on continuous data streams than they have on traditional computer communications. It is therefore a core issue to adopt congestion control strategies dedicated to continuous streams, i.e. schemes targeting functional goals which reflect the QoS requirements of these communications.

A hybrid RTT/Bandwidth control algorithm, built upon the TCP-friendly approach, is introduced in the source encoder regulation procedure. Apart from the TCP-friendliness property, the design goals of the hybrid control approach, include full rate control (even when no packet loss occurs), smooth rate variations (no sawtooth pattern) together with high responsiveness, as well as the ability to make use of current RTP/RTCP features, and easy extension to multicast scenarios. This rate control mechanism allows to maintain a stable SNR and visual quality while significantly decreasing the source throughput. Encoding at a bit rate adapted to the link, leading to few losses, is more efficient than using a higher bit rate which yields higher loss rates. By adjusting the source throughput to the available bandwidth, this mechanism, reducing the channel congestion, also leads to a better share of the network resources.

The remainder of this article is organized as follows. Section 2 reviews the issues associated with video communications in the Internet and the mechanisms that have been proposed to deal with them. Sections 3–5 tackle the error control issue, with Section 3 briefly addressing error resilience within the MPEG-4 framework, Section 4 investigating conditional replenishment schemes, and Section 5 being devoted to the Coding Mode Selection scheme that we have designed, and introducing some experimental results. Next we turn to the rate and congestion control issue, and introduce in Section 6 a hybrid RTT- and TCP-friendly-based rate control prediction scheme, followed by experimentation results. Section 7 is devoted to the multicast scenario and reviews the various issues associated with video multicasting along with their possible answers within the framework of our proposals. Finally, concluding remarks and directions for future work are given in Section 8.

2. Video communication over the Internet

The success of the best-effort Internet is largely bound to widespread use of TCP, which maintains good network conditions through its congestion control mechanism. However, the use of this protocol is incompatible with strict delay requirements of real-time multimedia. Multimedia delivery relies on unresponsive protocols such as UDP or/and RTP. RTP offers no reliability mechanisms, has no notion of connection and is usually implemented as part of the application. Unresponsive use of these protocols gives rise to severe threats for the media QoS, as well as for the network QoS, as more and more multimedia streams are being deployed. As the number of unresponsible data flows increases, congestion inside the network with its devastative effects on multimedia delivery and interaction (large packet losses and end-to-end delays) becomes a major concern.

Therefore, it is of prime importance to design loss resilient coding strategies as well as rate control strategies dedicated to multimedia communication inside best-effort networks, i.e. implement mechanisms that are able to cope with the QoS-oriented needs of multimedia applications, yet at the same time ensure proper congestion avoidance or recovery. Widely retained approaches go from simplified and robust temporal redundancy exploitation techniques, often under the form of conditional replenishment, to error control and congestion control strategies.

2.1. Conditional replenishment

The compression efficiency of motion-compensated temporal prediction is often sacrificed for the simplicity, and error resilience of conditional replenishment [18]. Temporal redundancy is only exploited through a change or motion detection

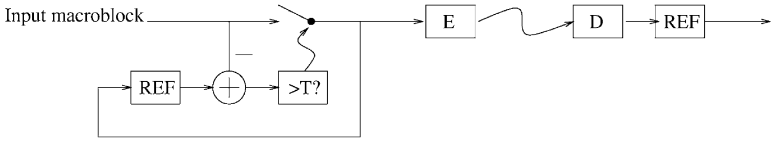

Fig. 1. Conditional replenishment scheme.

between blocks of adjacent frames. Fig. 1 depicts a block diagram for the conditional replenishment algorithm. For each block in a new frame, a distance between the reference block and the new block is computed. If the distance is above a threshold, the block is encoded and transmitted.

2.2. Error control mechanisms

The objectives of error control are to provide loss recovery facilities. To recover from loss, two well-known techniques exist, ARQ and FEC, under the form of so-called redundant data or of parity codes. ARQ consists in re-transmitting the original packets that are lost. Therefore, the sender needs to know the sequence number of the lost packet. This information may, for instance, be provided by the receiver by using the RTCP report packets. The principle of redundant data consists in re-transmitting into packets, information bits that have already been transmitted in the previous packet, under the same form or encoded at a lower bit rate. This mechanism is widely used for vital information such as picture headers in [8], and also for audio data [4]. FEC strategies for the Internet are often based on parity codes and block codes, such as the Reed–Solomon codes. One or more parity blocks over a group of k packets are generated by linearly independent combinations of data blocks, often by bit-wise exclusive-ORing of the k packets. The particular combination is called a parity code. After the parity operation, there is a total of n data plus parity blocks (i.e., $n-k$ parity blocks). This mechanism can recover from k losses in a n packet message. It increases the rate by a factor of k/n and adds latency. Reed–Solomon codes offer better protection than parity codes but at the expense of increased processing. A RS code takes a group of k data blocks and generates $n-k$ FEC blocks. Comparative to ARQ, using parity data, the sender needs only to know the packet loss probability (or maximum number of packets lost) but does not need to know their sequence number. This presents some advantages for evolutions towards multicast, the feedback being reduced from per-packet feedback to per-group of packets feedback.

Besides the error control strategy to retain, the control of the amount of redundant information added at the source is a major concern. It can be based on feedback information about the loss process measured at the destination, i.e. using QoS reporting mechanisms of RTCP.

2.3. Rate and congestion control mechanisms

Unicast congestion control schemes are typically designed as a feedback loop between a source and a receiver. In such a scheme the receiver is responsible for monitoring the network state, and sending periodic feedback information to the source. The latter uses this indication to compute a proper sending rate or to trigger some parameter tuning actions within its data encoding software (e.g. compression of raw video frames). The network state observation is to be smoothed, using either a sliding window or exponentially weighted moving averaging (EWMA), then coarsely quantized into two or three "network states", e.g. increase/decrease, or unloaded/loaded/overloaded. Coarse quantization requires one or two thresholds to be defined. Three-state schemes actually implement a "deadzone" feature which allows a more static steady-state behaviour. The network state characterization is then sent to the source via the feedback loop.

Some schemes have been proposed which require the receiver to monitor the round-trip time (RTT) on the path between sender and receiver (e.g. [25]). The goal of focusing on a delay quantity is to trigger early reaction from the source, in order to

avoid any packet loss. Since network queues building up in time of rising congestion increase the RTT, such schemes are supposed to react sufficiently early to withdraw the congestion before it actually induces packet losses. There are however three drawbacks associated to RTT monitoring. First, asymmetry within the network may make the RTT measure somewhat inefficient. Next, delay thresholds have to be chosen to coarsely quantize the monitored RTT, and it is a difficult task to accurately select "mean delay" values, that would result from standard network load. Finally, it has been proven that RTT-based schemes cannot easily interfere with loss-based ones (e.g. TCP sessions) on a fair basis, because the former have a much more conservative behaviour than the latter [1].

An alternative is to monitor packet losses, as TCP does. Several design choices may be made: the receiver may be asked to acknowledge incoming packets, either positively or negatively, or to compute a loss rate over some time interval. The latter may be calculated at the packet level or at a higher level (e.g. frames for video streams). Using the RTP protocol and its companion, RTCP makes it easy to feed the source with packet loss event reports.

A new trend has emerged, which emphasizes on the "network citizen" behaviour of the congestion control scheme. A property named TCP-friendliness captures the characteristics of a "good" session, with respect to TCP connections, that is a behaviour which allows conforming sessions to fairly share the network. However, the underlying principle of additive increase/multiplicative decrease typically gives rise to sawtooth rate patterns, which may be in contrast with QoS requirements of a video stream. Hence, broadly speaking, rate control schemes should avoid sawtooth rate patterns and rather aim at feeding the source with smooth rate indications, yet at the same time be almost as reactive as a typical TCP implementation is, in order to maintain some fairness between traditional data exchanges and multimedia sessions.

3. MPEG-4 error resilient modes

The MPEG-4 video syntax provides support for a set of specific error resilient modes. When the error resilience mode is "on" ("error-resilience-disabled" flag set to "0"), then a resync-marker is inserted by the encoder before the first macro-block after the number of bits output since the last resync-marker field exceeds a predetermined value. The marker spacing value is dependent on the anticipated error conditions of the transmission channel and compressed data rate. The compressed data included between two resync-markers is called a video packet. In order to make each video packet independently decodable, all predictively encoded information must be confined within a video packet so as to prevent the propagation of errors. However, depending on the initial setting of the resync-markers and of the transmission channels variations, the video packet may be larger than the size of one RTP packet, and may then be fragmented. This may adversely impact the loss resilience efficiency along with the overall network usage, since losing any fragment of a video packet renders the whole packet useless, yet unlost fragments are carried and processed by the network. Depending on the loss pattern, very high video packet loss rates may result from a moderate fragment loss rate.

As shown in Fig. 2, header information is also provided at the start of a video packet. This header contains information needed to restart the decoding process and includes the macroblock address (number) of the first macroblock contained in this packet and the quantization parameter (quant-scale) necessary to decode that first macroblock. The macroblock number provides the necessary

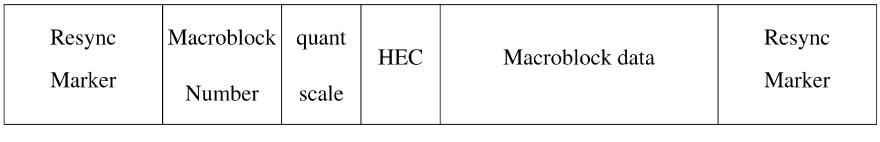

Fig. 2. MPEG-4 video packet structure.

spatial resynchronization while the quantization parameter allows the differential decoding process to be resynchronized. Following the quant-scale is the Header Extension Code (HEC). HEC is a single bit used to indicate whether additional information will be available in this header. If the HEC bit is set then the following additional information is available in this packet header: modulo time base, VOP-time-increment, VOP-coding-type, intra-dc-vlc-thr, VOP-fcode-forward, VOP-fcode-backward. When the Header Extension Code is set to "1", each video packet (VP) can be decoded independently. The information needed for decoding the VP is then included in the header extension code field. If the VOP header information is corrupted by the transmission error, it can be corrected by the HEC information. However, the above mechanisms turn out not to be sufficient under high loss conditions.

4. Conditional replenishment in an MPEG-4 compliant encoder

As a first step, the temporal prediction modes of an MPEG-4 compliant encoder are abandoned and replaced by a very simple conditional replenishment (CR) mechanism (Fig. 1). Note that the bitstream delivered is fully compliant with the VP structure (Fig. 2) and the MPEG-4 video syntax. The decoder is therefore strictly the same.

4.1. Motion detection

The motion detection aims at selecting 16×16 pixels macroblocks to be refreshed in intra-mode. Similarly to [18], macroblocks are divided into 4×4 blocks. Let $B_{t-1} = (r_1, \ldots, r_n)$ be a reference block of pixels in the reference frame buffer, and T the motion detection threshold. The macroblock containing the block of pixels $(x_1, \ldots, x_n)$ in the frame t will be refreshed in intra-mode if and only if

$$\left| \sum_{i=1}^{n} (r_i - x_i) \right| > T. \quad (1)$$

In order to reduce blocking artifacts, replenishment is also applied to the neighbouring macroblocks adjacent to the selected block. The threshold T can be adjusted according to the motion in the scene. In our experiments, a threshold $T = 100$ turned out to be well adapted to high motion sequences.

4.2. Results

The MPEG-4 Verification Model (without B frames) and the CR-based MPEG-4 compliant encoders have been used for encoding the CIF "coastguard" sequence at a constant bit-rate of 384 kbit/s. The frame rate of the source sequence is 10 frames/s. The MPEG-4 encoder is used in the rectangular mode, with INTRA refreshment periods of respectively 15 and 30 frames. The error resilient modes (described in Section 3) are enabled. To keep the bit-rate at a constant average value of 384 kbit/s, the VM5.1 SRC (scalable rate control) rate control algorithm [11] is used.

4.2.1. Experiment on error-free channels

Both encoders are first tested in an error-free environment, in order to compare the respective compression efficiency. Fig. 3 shows the PSNR ratio of decoded sequences for both codecs, as a function of the frame number. We observe that the PSNR curve of the CR system lies below the MPEG-4's one by an average of 2.65 dB, at the same bit-rate. This emphasizes the poor compression efficiency of CR, in comparison with a temporal prediction algorithm.

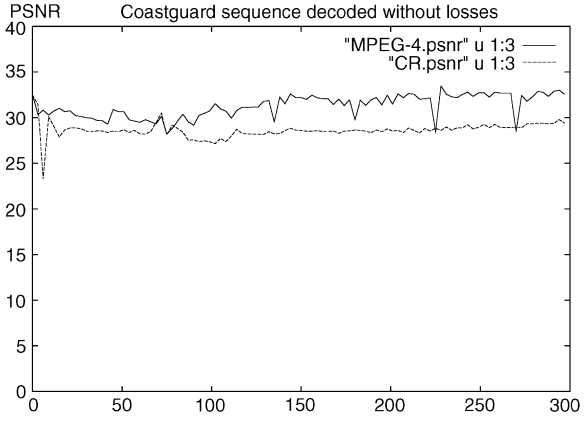

Fig. 3. Performances (PSNR) on error-free channels.

4.2.2. Experiments on finite state erasure channels

The current Internet is often modelled by a two-state Markov process, characterized by a "received" and a "lost" state. The transition probabilities from "received" to the "loss" state and from "loss" to the "received" state are denoted respectively as p and q. Measures collected between INRIA in France and University College London in the UK, reported in [2], led to Elliott–Gilbert parameters $p = 0.08$ and $q = 0.76$. In our experiments, the p and q parameters have been set to 0.08 and 0.60 in order to simulate a channel with high packet loss rates. Fig. 4 depicts comparative PSNR results of the Conditional Replenishment and the MPEG-4 encoders. An Intra refreshment period of 15 frames is used in the MPEG-4 encoder. The major result of this experiment is that CR performs much better than differential coding in the presence of packet losses. These results are also noticeable on the decoded images. In conclusion, despite its low compression efficiency, CR appears to be more adapted to Internet video coding than a temporal predictive scheme. In the next section, a new coding method is presented, which tries to gather the advantages of both CR and MPEG-like coding systems.

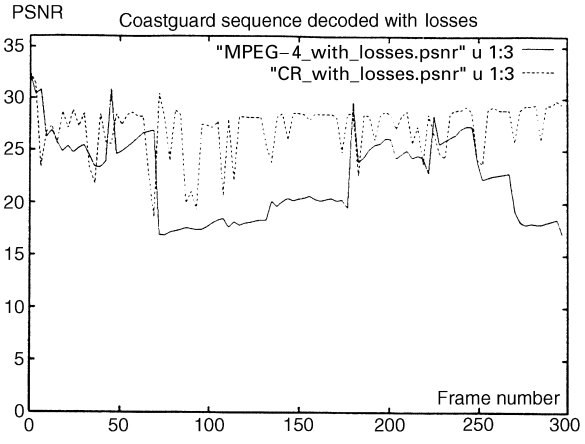

Fig. 4. Performances (PSNR) on finite state erasure channels. ($p = 0.08$, $q = 0.60$).

5. Coding mode selection

As shown above, CR-based encoders provide a higher packet loss resilience, but at the expense of poor compression efficiency. The purpose of this paragraph is to find a coding strategy that would optimize the trade-off between error robustness and compression efficiency. A solution, proposed in [35], consists in creating a dependence graph between macroblocks of consecutive frames. An algorithm provides an importance measure to each macroblock of each frame, and selects a set of macroblocks to be encoded in intra mode, according to a given bit budget. However, this method is not adapted to real-time video coding, because it needs multiple pass compression.

On the other hand, a coding mode selection is proposed in [34], for real-time video encoding on wireless channels. The method jointly optimizes macroblock coding modes and associated parameters (quantization parameters for instance), under a bit budget constraint. The purpose of this method is to exploit the numerous modes provided by the H.263 standard, in order to improve the rate-distortion performance of the source coding process. However, no channel characteristic is taken into account.

This section describes a mode selection algorithm, based on a distortion measure which exploits the knowledge of channel characteristics. In addition, the coding modes optimization is preceded by a motion detection process (as used in CR), in order to reduce the amount of macroblocks to be encoded, hence the encoder complexity.

5.1. Principle

The strategy retained consists in combining Conditional Replenishment with an intra/inter-coding mode decision mechanism, as shown in Fig. 5. The decision process is based on a rate-constrained optimization procedure that takes into account the channel characteristics.

The syntax of the MPEG-4 video encoder used supports, for the P-frames, the following modes:

- *INTRA*: intra-coded,
- *INTER*16: inter-coded with one motion vector per macroblock,

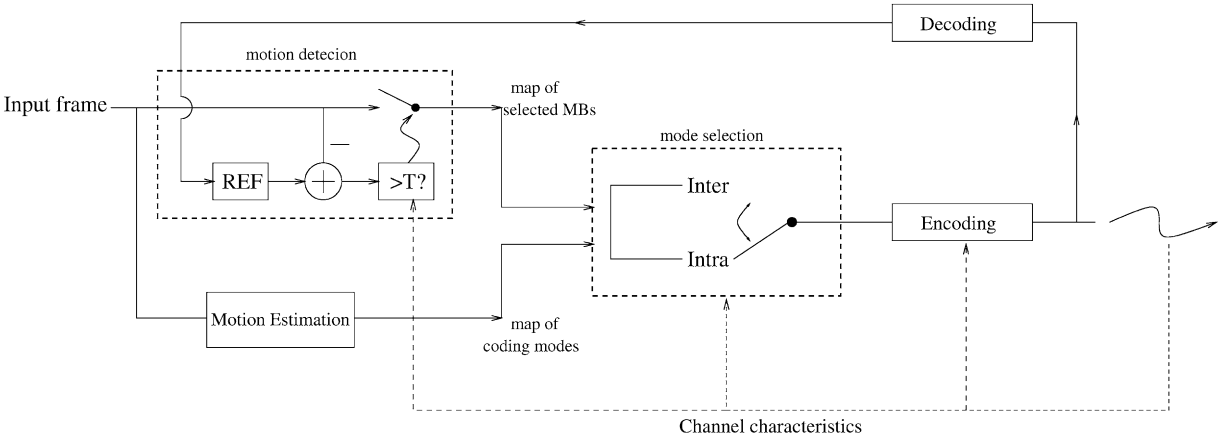

Fig. 5. Coding mode selection mechanism.

- *INTER*4*V*: inter-coded with four motion vectors per macroblock,
- *U-mode*: uncoded.

The *INTRA*/*INTER* mode decision is based on a comparison between the variance of the luminance of the original MB, and that of the prediction error. The coding mode in $\{INTER16, INTER4V\}$ yielding the smallest sum of absolute difference (SAD) between the original and the motion-compensated macroblocks is chosen. *U*-mode is chosen when the motion vector found and the associated quantized prediction error are equal to zero.

Building upon the conditional replenishment scheme, the macroblocks to be encoded are clustered into groups of N macroblocks (MB). For each macroblock X_i to be refreshed, the mode selection algorithm chooses between: *INTER* mode, i.e. *INTER*16 or *INTER*4*V*, following the strategy of the MPEG-4 encoder as described above, and the *INTRA* mode. Let $\mathscr{I} = \{INTRA, INTER\}$ be the set of possible coding modes for each MB of a group of MB $\mathscr{X} = \{X_1, \ldots, X_N\}$. A combination of coding modes for the GOB χ is an element $\mathscr{M} = \{M_1, \ldots, M_N\} \in \mathscr{I}^N$. The mode selection process aims at providing a best combination of coding modes, in the rate-distortion sense.

5.2. Distortion metrics

The distortion metric often used in mode selection, as in [34], measures the distance between a macroblock and its reconstructed version after inverse quantization. This problem is re-formulated here by defining a global distortion measure, taking also into account the channel distortion.

5.2.1. Channel models

In [9], the channel is modelled by a Bernoulli process. Considering the loss or receiving states of consecutive packets as independent events, the packet loss rate is expressed by an average probability P_e. However, best-effort Internet is often better modelled by finite state erasure channels and especially the Elliott–Gilbert channel [2]. The Elliott–Gilbert model is a two-state Markov process, as depicted in Fig. 6. The process is in state R if packet n at step n has been received and in state L otherwise. p and q are the transition probabilities between the two states. The average loss probability is $\bar{P}_e = p/(p+q)$. The state propagation across packets is modelled by the transition matrix

$$P = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}.$$

Note that the Elliott–Gilbert process is equivalent to a Bernoulli process if and only if $p + q = 1$. In addition, it can be shown that

$$\forall k \geqslant 1,\ \exists p_k, q_k \in]0,1[\ / P^k = \begin{pmatrix} 1-p_k & p_k \\ q_k & 1-q_k \end{pmatrix}.$$

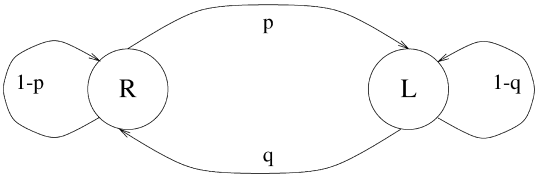

Fig. 6. Elliott–Gilbert model (R: received, L: lost).

So, given a number k of successive transitions, the packet loss process will converge towards a Bernoulli process if $p_k + q_k \sim 1$. Therefore, according to the parameters k, p and q, the mode selection algorithm developed here will alternately adopt a Bernoulli process or an Elliott–Gilbert process for modelling the channel. Indeed, the value $\det(P^k) = (1-(p+q))^k$ is compared to a threshold $\varepsilon > 0$. If $(1-(p+q))^k > \varepsilon$, then the packet loss model chosen is a Gilbert process. If $(1-(p+q))^k < \varepsilon$ it is a Bernoulli process.

5.2.2. Channel as a Bernoulli process

Let P_e be the packet loss probability. Let X_i^t be the original macroblock at spatial location i and frame number t. We call $\hat{X}_i^t$ the result of inverse quantization of X_i^t and $\tilde{X}_i^t$ the concealed version of $\hat{X}_i^t$ at the decoder side, when a packet loss occurs. The concealment method considered consists in replacing the current MB by the MB at same spatial location in the previous frame. By adopting a Bernoulli process for modelling the channel, the distortion for an intra-coded macroblock can be expressed as

$$D(X_i^t, \text{Intra}) = (1-P_e)|X_i^t - \hat{X}_i^t| + P_e|X_i^t - \hat{X}_i^{t-1}|, \tag{2}$$

where $|.|$ denotes the mean-square error of a macroblock. Let us now consider the inter-coding mode. The distortion metric introduced covers the general case where a frame can be fragmented across several packets and reciprocally the case where several frames can be grouped into one packet. This depends on both the source rate and the maximum transfert unit (MTU) of the network. Let k_i be the number of packets sent on the network and containing at least one macroblock at spatial location i, since the last intra-macroblock (see Fig. 7). Let $\phi_i(l)$ be the number of occurrences of a MB at spatial location i in the packet $l \in [0, k_i - 1]$. For example, in Fig. 7, we have $k_i = 2$, $\phi_i(0) = 1$ and $\phi_i(1) = 3$. Let Φ_i be the function defined by

$$\forall l \in [0, k_i - 1], \quad \Phi_i(l) = \sum_{j=0}^{l} \phi_i(j).$$

The function Φ_i captures the number of macroblocks at the spatial location i contained in the last l packets transmitted on the network, among the k_i packets considered here. To simplify the expression of the distortion metric, we only consider prediction modes with motion vectors equal to zero. Let ξ_{t-1}^t be the prediction error between the macroblocks $\hat{X}_i^{t-1}$ and X_i^t, at spatial location i in frames of numbers $t-1$ and t.

We suppose that the last intra-coded macroblock at spatial location i, denoted MBI in Fig. 7, has been received. If we consider Fig. 7, the distortion metric for an inter-coded macroblock is given by

$$\begin{aligned} &D(X_i^t, \text{Inter}) \\ &= (1-P_e)^2|X_i^t - \hat{X}_i^t| + (1-P_e)P_e|X_i^t - \hat{X}_i^{t-\Phi_i(0)}| \\ &\quad + P_e\Bigg[P_e|X_i^t - \hat{X}_i^{t-\Phi_i(1)}| + (1-P_e) \\ &\qquad \left|X_i^t - \hat{X}_i^{t-\Phi_i(1)} - \sum_{j=t-\Phi_i(0)}^{t-1} \xi_j^{j+1}\right|\Bigg]. \end{aligned}$$

Implementing an accurate distortion measure which would take into account all the possible loss cases, for each macroblock i in the image, would require to store all the macroblocks containing the prediction error signals ξ_j^{j+1} transmitted since last intra-coded macroblock at the same spatial location i. In order to reduce the implementation complexity, an approximate distortion measure, based on the assumption that the difference between two macroblocks at the same spatial location i increases with their temporal distance. The distortion measure is indeed approximated by the upper bound $|X_i^t - \hat{X}_{i'}^{t'}|$, where $\hat{X}_{i'}^{t'}$ is the inverse quantized version of the last intra-MB at location i. This assumption leads to

$$\begin{aligned} D(X_i^t, \text{Inter}) \leqslant{}& (1-P_e)^{k_i}|X_i^t - \hat{X}_i^t| \\ &+ (1-(1-P_e)^{k_i})|X_i^t - \hat{X}_{i'}^{t'}|. \end{aligned} \tag{3}$$

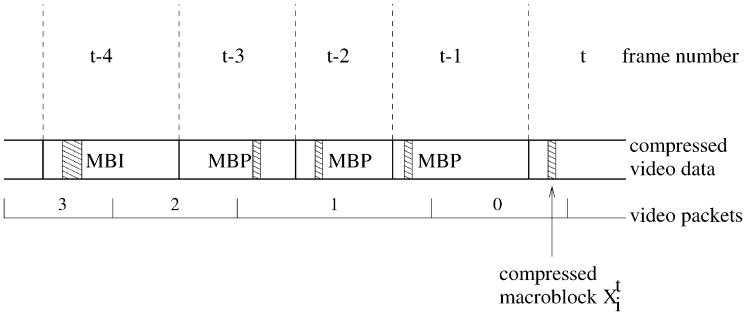

Fig. 7. Correspondence between video compressed stream and packet numbers.

The implementation of this distortion metric is quite simple: each time a MB is intra-coded, we store its value $\hat{X}_i^{t_l}$ in a reference frame buffer. The value k_i of relation (3) for each macroblock is updated each time a video packet is formed.

5.2.3. Channel as an Elliott–Gilbert process

If we take into account the memory of the Elliott–Gilbert model, the distortion expression is the same as in Eq. (2) for the intracoding mode. P_e is replaced by the average loss probability $\overline{P_e} = p/(p+q)$. In inter-mode, the distortion measure can be developed by adopting a similar approach as for the Bernoulli model.

The difference is that k_i is redefined as the total number of packets sent on the network since last intra-macroblock at spatial location i, and we use the transition probabilities of the Elliott–Gilbert process. In the same way as with the Bernoulli model and with the same approximation, we have

$$D(X_i^t, \text{Inter}) \leqslant (1-p)^{k_i} |X_i^t - \hat{X}_i^t| + (1-(1-p)^{k_i}) |X_i^t - \hat{X}_i^{t_l}|. \quad (4)$$

The value k_i for each macroblock is maintained almost in the same way as before (Section 5.2.2), except that it accounts for the total number of packets since the last intra-coded macroblock.

5.3. Mode selection

Given the definitions in Section 5.1, if the rate allocated to the current frame is R_{frame}, then the rate allocated to each GOB is proportional to R_{frame} and to the length of the GOB:

$$R_c = \alpha \times R_{\text{frame}} \times \frac{\text{Nb}_{\text{GOB}}}{\text{Nb}_{\text{frame}}}, \quad (5)$$

where Nb_{GOB} and Nb_{frame} represent respectively the number of macroblocks in the considered GOB and the total number of macroblocks to refresh in the current frame. The choice of parameter α is discussed in Section 5.4. The problem of finding the best coding modes combination for the subset of macroblocks $\mathscr{X}$ consists in finding

$$\min_{\mathscr{M}} \quad D(\mathscr{X}, \mathscr{M})$$
$$\text{s.t.} \quad R(\mathscr{X}, \mathscr{M}) \leqslant R_c,$$

where D, R are the distortion and rate measures and R_c the rate constraint for the GOB $\mathscr{X}$. The distortion measure is given respectively by expressions (3) or (4), according to the test made on the channel model as described in Section 5.2.1. The above rate constrained optimization problem is rewritten as an unconstrained Lagrangian formulation. The distortion measure being additive, the Lagrangian cost function becomes

$$J(\mathscr{X}, \mathscr{M}) = \sum_{i=1}^{N} J(X_i, \mathscr{M}) = \sum_{i=1}^{N} D(X_i, \mathscr{M}) + \lambda R(X_i, \mathscr{M}).$$

The algorithm then searches for the set $\mathscr{M}$ of coding modes minimizing the above objective cost

function,

$$\min_{\mathscr{M}} (J(\mathscr{X}, \mathscr{M})). \tag{6}$$

The minimization of this functional can be carried out with the following steps [21,28]:

Initialization: Two values of λ, λ_l and λ_u are chosen such as $\lambda_l \leqslant \lambda_u$. The minimization of J_{λ_l} gives the rate R_l and distortion D_l parameters. Similarly, the minimization of J_{λ_u} gives the parameters R_u and D_u. R_u and R_l must verify

$$R_u \leqslant R_c \leqslant R_l. \tag{7}$$

This condition requires a careful choice of λ_l and λ_u. In practice, we take $\lambda_l = 0$ (minimizing the distortion without rate constraint) and $\lambda_u = \infty$ (minimizing the bit-rate). If the constraint (7) is met with equality for one of the two values, we have an exact solution. Otherwise the algorithm proceeds with the following steps:

1. Minimize J_λ where $\lambda = (D_l - D_u)/(R_u - R_l) + \varepsilon$, ε being a given real number. This provides a rate R and a distortion D.
2. If $R = R_u$, the algorithm is over. If $R > R_u$, do $\lambda_l := \lambda$ and goto step (1). Otherwise do $\lambda_u := \lambda$ and goto step (1).

5.4. Rate control mechanisms

5.4.1. Rate control in the MPEG-4 verification model

The rate control algorithm used in our video encoder is the VM5.1 SRC presented in [10]. This rate control process uses an output buffer and assumes that the encoder rate distortion function can be modelled by

$$R = X_1 S Q^{-1} + X_2 S Q^{-2}, \tag{8}$$

where R is the encoding bit count and S the sum of absolute differences between original current macroblock and previous reconstructed macroblock for a P-macroblock. The variable Q, X_1 and X_2 represent respectively quantization and rate control modelling parameters. The rate control mechanism consists of four main steps:

1. Initialization of parameters X_1 and X_2.
2. Computation of the target bit-rate before encoding, based on the available and last encoded frame bits and on the buffer status.
3. Computation of the quantization parameter Q before encoding, based on the rate distortion function (8) of the source encoder.
4. Updating of the rate distortion function given by Eq. (8).

5.4.2. Choice of parameter α

The rate control presented in the previous section attempts to achieve optimal quality for a given target bit-rate. Hence, it chooses quantization parameters as fine as possible, according to the available bit-rate.

As a result, if $\alpha = 1$ in Eq. (5), then the amount of intra-selected mode is very low because of the bit-rate cost of intra-coding when the quantizer parameters are fine.

A choice of $\alpha > 1$ in Eq. (5) allows to counterbalance the VM5.1 SRC rate control process. The parameter α turns out to have a high influence on the amount of intra-coded macroblocks. Hence, it allows to get a trade-off between the number of intra-MBs selected by the mode selection process and the fineness of the quantization parameters, adjusted by the VM5.1 SRC rate control. This trade-off can be found by tuning α according to the channel characteristics, for instance the average loss probability. We then define α as a function of P_e, and following the next relation:

$$\alpha = \frac{1}{1 - P_e}. \tag{9}$$

As a result, the whole rate control process is made of a two separable steps that counterbalance each other.

Some work is driven in order to jointly optimize both quantization parameters and coding modes in the rate-distortion sense as in [12,13,34].

5.5. Results

The MPEG-4 compliant encoder described above has been used for encoding "coastguard" and "news" sequences, with the same coding configuration as in Section 4.2. The mode selection algorithm uses the same Elliott–Gilbert transition probabilities as in Section 4.2, in presence of packet losses, i.e. $p = 0.08$ and $q = 0.60$. Note that no I

frame is used in the mode selection encoder, except for the first frame of the sequence. Only P frames with intra-macroblocks are encoded.

5.5.1. Experiments on error-free channels

The three encoders, namely MPEG-4 Verification Model, the CR-based encoder, and the MPEG-4 compliant encoder incorporating the Channel Adaptive Mode Selection mechanism, are first tested in an Error-Free channel. From now on, the three encoders will be respectively referred as MPEG-4, CR and the CAMS encoders.

Fig. 8 depicts the PSNR ratio as a function of the frame number for each encoder. We observe that, on average, the CAMS encoder performs better than CR when no packet loss occurs. Its rate distortion performance remains a bit lower than the rate distortion performance of the MPEG-4 VM. This is due to the use of the threshold T of relation (1) which is at the moment not adapted to the channel characteristics. The lower PSNR values are due to macroblocks that are not selected by the motion detection process.

5.5.2. Experiments on finite state erasure channels

Considering the same channel characteristics as in Section 4.2, Fig. 9 depicts the PSNR values obtained. When losses occur, the MPEG curve falls below the two others for both sequences. The CR encoder's curve is the most stable one in "Coastguard" sequence, in presence of packet losses and high motion in the sequence (first part of "Coastguard" sequence). However, when the

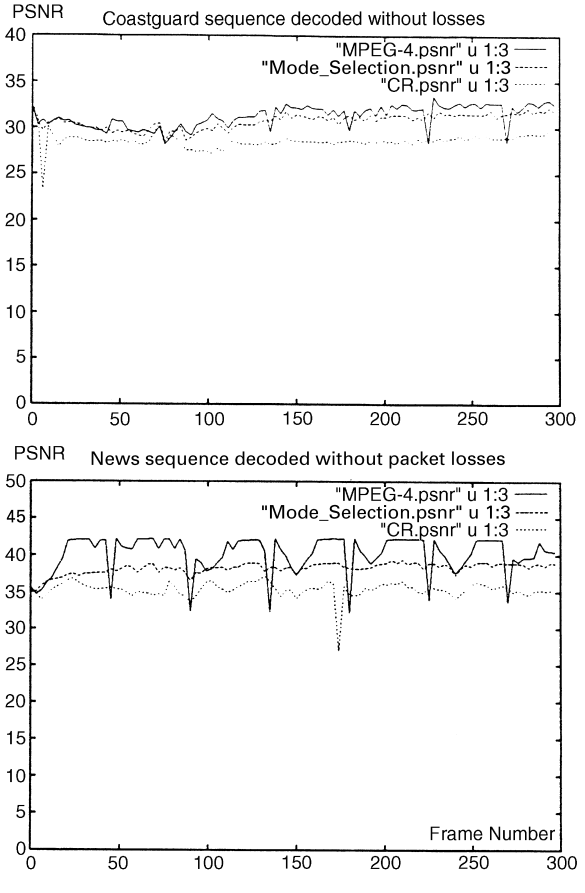

Fig. 8. Performances (PSNR) on error-free channels.

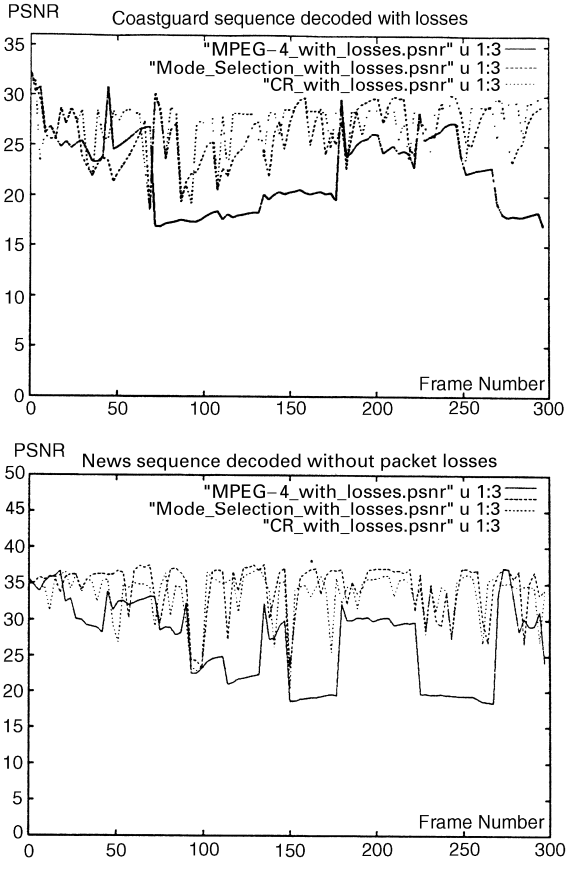

Fig. 9. Performances (PSNR) on finite state erasure channels.

amount of motion is reasonable, the CAMS encoder provides better PSNR values than CR when no channel error occurs, and remains comparable to it in the presence of packet losses.

As a matter of fact, the mode selection scheme seems to provide a good trade-off between compression efficiency and packet loss resilience. Therefore, taking into account the channel statistical model and the scene activity through the optimization process allows to improve the efficiency of video transmission across a finite state erasure channel.

Fig. 10 depicts the PSNR values obtained as a function of the frame number, when considering distortion metrics based exclusively on the Bernoulli or the Elliott–Gilbert models, for both an Error-Free and a Finite State Erasure channel. The usage of the Elliott–Gilbert model leads to a higher amount of intra-coded macroblocks. This explains the higher stability in presence of packet losses, of the PSNR obtained with the Elliott–Gilbert model. The Gilbert model's curve recovers faster a packet loss than the other one. Hence, the use of the Elliott–Gilbert in the distortion measure in the coding mode selection process allows a higher resilience to packet losses.

6. Rate and congestion control

The adaptive mode selection mechanism described above increases the trade-off between packet loss robustness and compression efficiency but does not avoid losses. A rate control mechanism based on congestion control and rate-predictive models is developed here in order to minimize the amount of losses by matching the video source bandwidth requirement to the available network capacity.

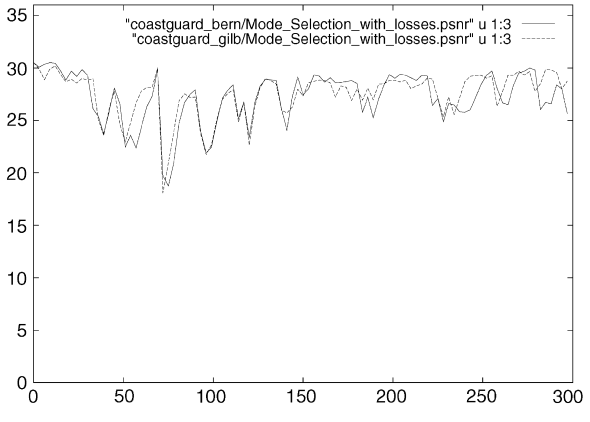

Fig. 10. Performances (PSNR) obtained when using exclusively the Bernoulli or Elliott–Gilbert model.

Given our loss resilient coding scheme, we now turn to the congestion control issue, and propose a rate control scheme dedicated to video communication, the behaviour of which is compatible with standard (i.e. TCP) communications.

6.1. TCP throughput models

Early work in this area has shown that the stationary throughput of a saturated TCP sender (i.e. one with an infinite amount of bytes to send) is on the inverse order of the square root of the loss rate observed by the connection [16]. In the following we consider the so-called MF model, due to Madhavi and Floyd [15]. Another model of interest is the PFTK one, which was proposed by Padhye et al. [19]. Mahdavi and Floyd propose in [15] the equation

$$\mathrm{BW} = 1.22 \times \frac{\mathrm{MTU}}{\mathrm{RTT} \times \sqrt{\mathrm{Loss}}} \tag{10}$$

to compute the bandwidth that a TCP connection receives, provided it has MTU bytes per packet, and incurs a roundtrip time of RTT seconds and a loss rate of Loss. This equation comes from a steady-state analysis of the TCP congestion avoidance mechanism. Such a model is known to be valid for loss rates up to about 15%. In the following, we consider using this model in a rate prediction process designed to feed the source's rate control mechanism.

6.1.1. Parameter estimation

Implementing a throughput model requires knowledge of the connection's MTU, RTT and loss rate. The MTU can be either set to a fixed value, for instance the standard minimum value of 576 bytes defined for TCP, or determined using an MTU discovery algorithm [32]. In our experiments, we

used fixed length packets which allow for a constant MTU, set to 576 bytes.

The RTT estimation is performed by keeping a recent average value. Following TCP's RTT smoothing strategy, we implemented an exponential filter with a constant 0.9 smoothing factor [29]. This filter is fed with RTT measures conducted by a very simple and robust (stateless) request/reply protocol, which takes place as a background process. This protocol sends RTT requests carrying the host's local time periodically, and delivers a new RTT measure upon reception of the RTT reply which carries the same, unchanged time value. These requests are sent once per RTT. The smoothed RTT estimation is hereafter denoted by S_{RTT}.

The loss rate estimation is performed by observing packet losses, detected using sequence numbers, and again keeping a recent average value. The averaging operation is somewhat problematic in this case, because we may think of network conditions where the loss rate reduces to zero at a given time. Using an exponential filter, whatever the smoothing factor is set to, implies that losses are taken into account for quite a time. This generates a "residual" loss rate which may trigger extremely high rate predictions from the throughput model as it tends to zero. Instead we use a time-based sliding window, the width of which is set to a 30 RTT time interval, as determined through some simulations in [31].

6.2. Architecture

Our purpose is to make use of the TCP-throughput model to regulate the output rate of our video codec. The envisioned architecture is as follows. In addition to the forward data stream, a backward control stream is set up between the source and the receiver. The control stream is made of periodic feedback packets, which carry explicit rate information used by the source application to regulate its data output. In this architecture the throughput prediction model is located at the receiver side, and the results of its computations are periodically sent back to the source. An alternate architecture could be designed, where the parameters required by the throughput model would be carried inside feedback packets to the source, the latter implementing the throughput model computations. An important question relates to the feedback frequency, as too frequent estimations may well overload the network with control packets, whereas too few feedback indications may render the regulation scheme inefficient [22]. According to [15], a time interval of at least one RTT is required between two consecutive actions, so as to take into account the impact of the former one. In order not to overload the network when the RTT is small, we choose in the rest of this section to set the feedback period to the maximum between one second and one RTT.

6.3. New rate prediction model

An issue when investigating an MF-based regulation is its so-called non-self-limiting behaviour. As opposed to TCP's congestion control mechanisms which obey an upper rate bound as a consequence of the maximum congestion window size, the MF model may potentially compute infinite rate predictions, typically when the loss rate is null. The non-self-limiting behaviour of most rate-based congestion control schemes is classically answered by implementing timers, which expire in the absence of feedback indication, thus forcing the source to lower its rate [22]. In the present case however, unlimited rate may occur as a result of a valid feedback indication, hence another solution is to be designed.

Hereafter we use a hybrid rate control principle designed to retain the desirable properties of TCP-friendly based regulation and to answer these non-self-limiting problems. Its basic idea is to perform lossless rate adaptation whenever possible, using an RTT-based control loop, yet to embed a TCP-friendly rate prediction model which gets into play upon lossy conditions. The usual purpose of RTT-based rate control is to allow early reaction to congestion, thereby avoiding packet losses. A classical scheme is to compare RTT observations against a given RTT threshold value, and to vary the source rate by performing additive increase when the last RTT measure is under the threshold, and multiplicative decrease when the observed RTT exceeds the threshold. Our approach retains the threshold mechanism, but differs from the previous in the way the source rate is generated. The

RTT-based control mechanism uses the smoothed RTT measure already computed for the TCP-throughput model as a basis to compute the RTT threshold. More precisely, the RTT threshold T_{RTT} is set to $S_{RTT} + k \times$ standard deviation (S_{RTT}). The receiver maintains an averaged measure R of its received rate. If the last RTT observation is above the threshold, then it generates an RTT-predicted rate P_{RTT} equal to $R - K$, where K is a constant rate increment. If the observed RTT is below the threshold, then the RTT-predicted rate P_{RTT} is set to $R + K$. In the subsequent experiments we take $k = 0.9$ and $K = 1$ kb/s. The received rate R is smoothed by a time-based sliding window, the width of which is tuned to 30 RTT, according to the window size computation method proposed in [31]. We denote by P_{TCP} the rate predicted by the TCP throughput model. The actual mixed prediction P_{MIX} is computed as follows:

- if P_{TCP} is valid (that is, not infinite), let $P_{MIX} = (P_{TCP} + P_{RTT})/2$;
- otherwise, let $P_{MIX} = P_{RTT}$.

The computation is performed once during a regulation round, the frequency of which is set to the last smoothed RTT measure.

In this scheme, the TCP throughput-based prediction acts as a "master controller" with respect to the RTT-based prediction. This behaviour comes from the use of the actually received rate as a basis for RTT-based prediction. Since the rate actually received relates to the previous feedback indications, the RTT-based prediction is not supposed to impact it by a large amount, but rather to closely follow it. Hence, under lossy conditions, the TCP throughput-based prediction takes a major role in the overall rate prediction.

If we indeed consider that P_{RTT} at a given regulation round is equal to P_{MIX} as computed during the previous round (not taking into account packet losses and the constant K), then, assuming a constant P_{TCP}, we have a geometric series $\{P_{MIX}\}$ which converges to P_{TCP}.

6.4. Experimentation results

In [30], a TCP-friendly rate control, based on a pure MF model, is used to regulate a video source. Although the authors avoid the typical sawtooth rate variations encountered with a pure MF regulation [31], they do so by estimating the model parameters on a long term basis, and thus cannot expect early reaction to network variations. Our purpose is to improve the bandwidth usage and video quality stability at the receiver, through a decreased amount of packet losses, and yet have a highly reactive scheme in order to be actually "TCP-friendly", with respect to classical data transmissions.

6.4.1. Rate control within the video encoder

The rate control algorithm in the video encoder has been modified so as to adapt the source to the feedback information received. This information corresponds to the new network bandwidth available. Some parameters of the rate control [11] process are recalculated:

- the bit budget available to encode the current GOP,
- the size of the buffer,
- the amount of bits to remove from the buffer at each frame.

Quantization parameters of the encoder are then adjusted using these new values, in the same way as in [11].

6.4.2. Network topology

Due to the non real-time nature of our experimental video codec, we performed rate control experiments over the Internet using a simulated codec in order to get trace files, and then we used these files off-line as an input to the codec so as to assess the impact of packet losses and rate control over the quality of the video sequence.

The topology we experimented over is of wide-area network (WAN) kind, with the video source located at INRIA Sophia and the receiver located at INRIA Rennes (Fig. 11). The path between source and receiver thus crosses two regional networks (Ouest-Recherche and R3T2) and the French nation-wide R.N.I. interconnection network. It consists of a dozen routers. In addition, a packet source was used at ENST Bretagne in order to generate cross-traffic within the regional network, thus adding to congestion.

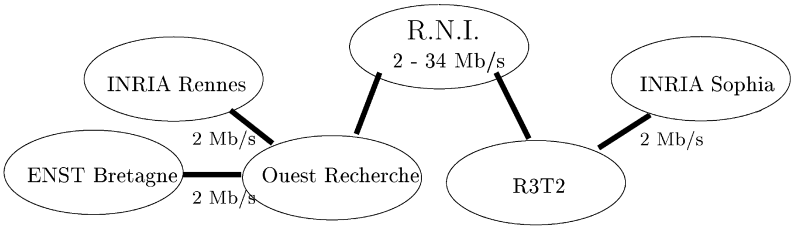

Fig. 11. WAN topology.

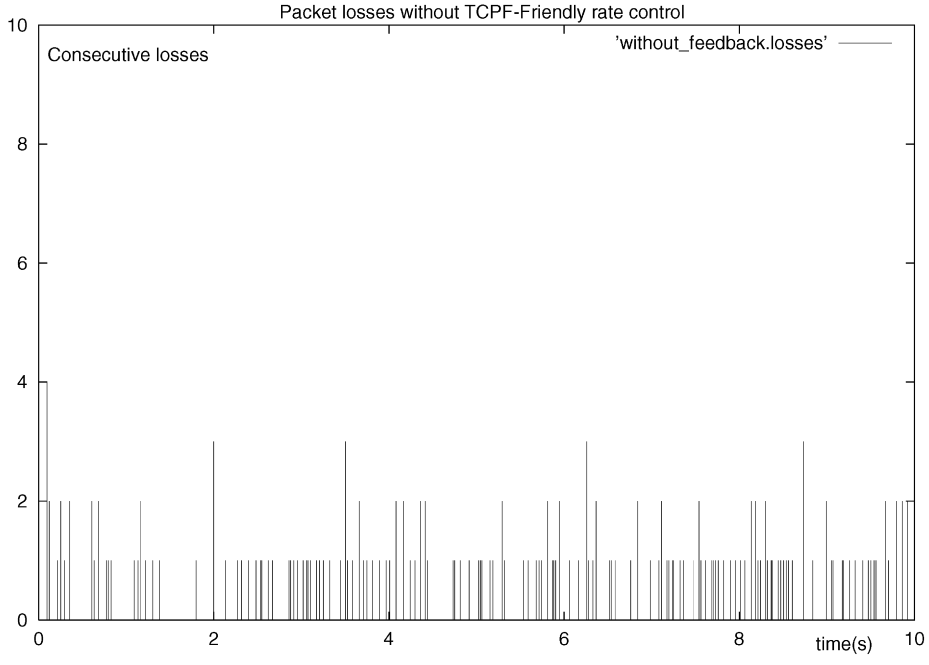

Fig. 12. Consecutive packet losses without TCP-friendly rate control.

6.4.3. Rate control results

Figs. 12 and 13 depict the packet loss patterns we obtained from a typical experimentation round on the aforementioned topology. The initial rate was set to 384 kbits/s, we traced the packet losses with the TCP-friendly rate control process disabled (Fig. 12) or enabled (Fig. 13). In the latter case, CBR rate control is performed. As can be easily seen in the figure, using the TCP-friendly rate control feature allows to drastically reduce the occurrence of packet losses. This is however at the expense of the transmission rate, as the TCP-friendly rate control mechanism triggers a large decrease in the allowed rate through the feedback indications, depicted by Fig. 14.

The feedback indications were then used to drive our video source, while the packet loss traces were used to simulate the real channel. Fig. 15 depicts the PSNR of the resulting decoded sequence as a function of the frame number. The major result we can observe is that the mean PSNR values obtained are comparable for the two versions (with and without TCP-friendly rate control), despite the large difference in allowed rate. TCP-friendly rate

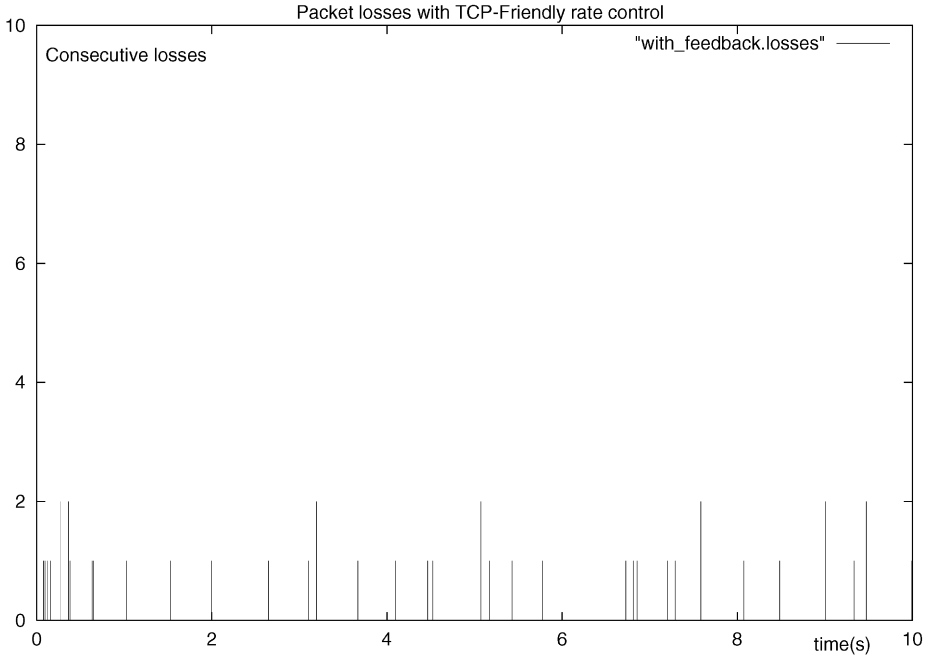

Fig. 13. Consecutive packet losses with TCP-friendly rate control.

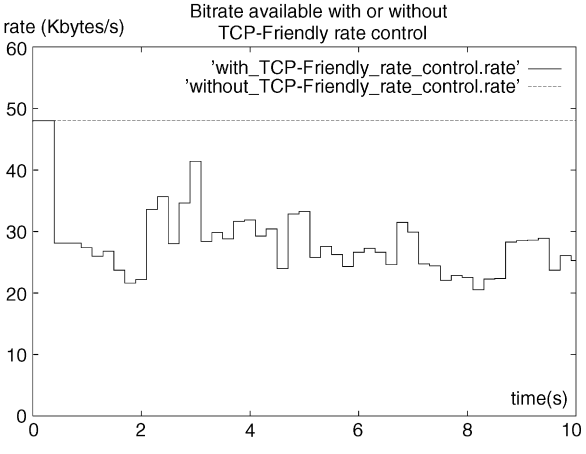

Fig. 14. Rate constraint provided by the receiver to the source.

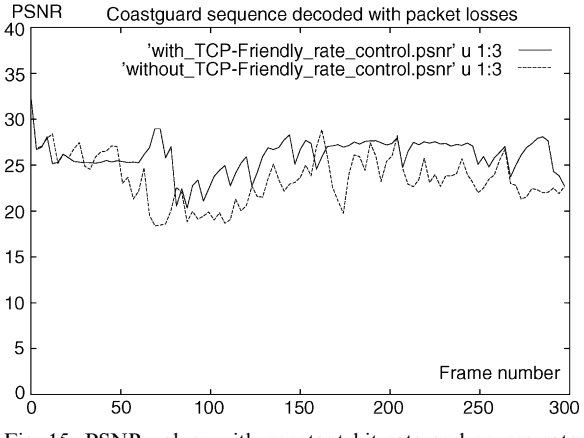

Fig. 15. PSNR values with constant bit rate and source rate adapted to TCP-friendly prediction.

control also allows a more stable PSNR curve (starting from frame number 75). This is explained by the sensitivity of temporal prediction coding to packet losses. Encoding at a bit rate adapted to the link, thus leading to few losses, turns out to be more efficient than using a higher bit rate which yields a higher loss rate.

7. Perspectives

7.1. Multicast communications

Since 1992, when the Mbone (Multicast Backbone) was first introduced, multicast communications across the Internet have become the focus of

a great amount of studies. Video multicasting across a best-effort Internet is a specially interesting topic as it faces heterogeneity issues. In a multicast topology (multicast delivery tree in the $1 \rightarrow N$ case, acyclic graph in the $M \rightarrow N$ case), network conditions such as loss rate and queuing delays are not homogeneous in the general case. Rather, there may be local congestions affecting downstream delivery of the video stream in some branches of the topology. Furthermore, receiver's heterogeneity may also be considered as real-time video decoding and display is somewhat tied to the particular receiver's hardware and software performances. Recent work in this area has shed the light on the benefits of subband coding and multichannel multicasting in the framework of heterogeneous bandwidth availability: by delivering subbands through distinct network channels (e.g. RTP sessions) and relying on standard group management protocols to join/leave these channels, it is possible for a given receiver to dynamically adapt the amount of video data it receives to the available bandwidth (see for instance the *Receiver-driven Layered Multicast* proposal [17]).

The issue of packet loss is also the subject of intense study. On the one hand, reliable multicasting has been proposed with a variety of mechanisms making use of selective and hierarchical acknowledgement and retransmissions schemes [14,20]. On the other hand, packet losses within a real-time flow (e.g. video) can be dealt with using FEC strategies tailored to the multicast framework. It is, for instance, possible for a video source to send separate FEC channels that can be joined by receivers experiencing a high loss rate [23].

7.2. Feedback in multicast

The use of feedback schemes in a multicast scenario faces two major issues. The first one deals with the so-called feedback implosion which results from straightforward re-use of a unicast feedback scheme in a multicast framework. As the number of participants in the multicast session increases, so does the number of receiver reports that must be carried by the network and processed by the source. Moreover, the arrivals of receiver reports may well be synchronized, raising the source's burden to an unbearable level [3]. A number of schemes have been devised to alleviate this problem. Probabilistic approaches require each receiver to observe some random delay before sending a receiver report [36]. This makes it possible to assign unequal "importance" to receivers by changing some weights in the random process, but the delays before a congestion indication is taken into account by the source may be prohibitive. Another approach is for the source to perform progressive, topological polling of the receivers using, for instance, *time to live* (TTL) mechanisms to control the number of responses. The main drawback of this method lies in its inaccuracy, as a modest TTL increase may yield a drastic increase of the number of answers, thus causing a feedback implosion if the source is overwhelmed. A third alternative was proposed [3]. Their scheme is also based on progressive polling but relies on random keys computed by each receiver. Progressive polling is done by varying the number of significative bits, thus allowing the source to quite accurately poll the number of receivers it wants. This number is further reduced by filtering the answers: a new poll embeds the maximum congestion state already known by the source, so that only receivers incurring worse conditions have to reply. The RTP/RTCP standard also tackles the feedback implosion problem. RFC 1889 suggests that the fraction of traffic dedicated to control packets (i.e. reports) does not exceed 5% of the bandwidth assigned to the multicast session [26]. Moreover the inter-departure time of RTCP packets is recommended to be larger than a minimum of 5 s (a lower minimum may be allowed, particularly for unicast sessions [27]), and is subject to a random variation factor taken over the range $[0.5; 1.5]$. The inter-departure time interval is calculated with the help of an estimate of the session size (i.e. number of participants), so that they collectively attain the 5% bandwidth (the interval scales linearly with the session size). Further studies have shed the light on scalability troubles occurring when the number of participants varies rapidly, as the session size estimate may not reflect the increase or decrease in a timely manner, and appropriate mechanisms have been proposed [24,27].

The other issue associated with multicast feedback is that of aggregating heterogeneous reports into a consistent view of the communication state. In some cases this considerably restricts the usefulness of multicast feedback schemes. Consider for instance an adaptation process where receivers send back to the source the bandwidth availability they infer. Then the source faces a problematic trade-off as it has to determine a single transmission rate that is bearable to all (or at least to a majority of) receivers. In most cases this translates into selecting the lowest common rate between all indicated values, or a rate low enough that a vast majority of receivers will not experience congestion. The corollary is reduced quality for all receivers, whether they actually experience congestion or not [3,6,36]. When the feedback scheme is devoted to error protection, as in our mode selection approach, then the source has to take into account the worse error conditions encountered by the different receivers. The corollary in this case is suboptimal rate/distorsion ratio, since most forward error correcting data is useless to all but a few receivers.

Layered coding and transmission, as discussed above, alleviate this problem by making it possible to adapt the rate or amount of error control data on a subband basis. A variety of multicast schemes making use of layered coding for audio and video communications have been proposed, some of which rely on a multicast feedback scheme. The *Destination Set Grouping* scheme is presented in [7], where a source produces multiple versions of a video sequence and lets the receivers individually choose which version to join. In addition, a feedback mechanism makes it possible for the receivers to alter the rate of the subband they chose. The *CafeMocha* approach is proposed in [5], where receivers are supposed to leave an enhancement layer as soon as the loss rate passes a given threshold. An indirect feedback mechanism is built by having the source monitor the session size of each layer, and adapt the rate of a layer that is too often abandoned by congested receivers.

7.3. Video encoder adaptation

The mechanisms we have proposed in the previous sections, namely coding mode selection and rate control using a TCP-friendly rate prediction model, may be adapted to the multicast framework and exploit the scalable coding features which have been designed in MPEG-4. Object scalability consists in coding each region of interest (ROI) in different video object layers (VOL). The motion detection process can be easily adapted to objects of arbitrary shape, taking into account their position in a reference window (see [11]). Temporal scalability consists in increasing the temporal resolution of the base VOL, or of a partial region of it. TCP-friendly rate control and coding mode selection can be performed in a straightforward way on the different layers. Spatial scalability aims at increasing spatial resolution of the base layer VOPs. The mode selection process can be applied to P VOPs of the enhancement VOL by taking into account additive coding modes based on prediction from the corresponding VOP in the base layer. Similarly, for B VOPs, prediction and interpolation from the base layer have to be taken into account. Since coding mode selection relies on the Elliott–Gilbert model of the channel, in a multicast scenario we have to consider having a distinct channel model for each receiver, as loss conditions in different locations of the multicast distribution tree are unlikely to be identical. It is therefore the responsibility of each receiver to calculate the Elliott–Gilbert model for the path leading to it, and to communicate the transition probabilities p and q to the source. Next the source may apply the "worst-case" loss model to the mode selection process, thus providing disproportionate error protection to those receivers which experience few losses, but with moderate impact on the resulting quality.

Alternately, it is possible to further refine the loss model on a layer basis, by requiring the receivers to provide the set of layers they have joined to along with the transition probabilities. This allows the source to select a worst-case model for a given layer between only the loss models provided by receivers actually receiving this layer, and may potentially improve the signal-to-noise ratio of higher layers as it is unlikely that heavily congested receivers have joined them.

As for the rate adaptation process, our rate prediction model can be implemented inside each

receiver. This provides for individual congestion monitoring and allows each receiver to determine its bottleneck rate. With the source advertising the layers it generates along with their current rates on a multicast control channel, each receiver is free to join to the layers that collectively fulfill its bandwidth capacity, similar to the approach taken in [32]. Rate adaptation may further be implemented in several ways.

For those applications which involve a limited number of participants (for instance up to ten receivers), we envision that the number of layers equates the number of participants, so that each one gets a dedicated enhancement layer. Considering that receivers $R_1, R_2, \ldots, R_n$ are ordered by increasing rate demand, the least demanding receiver R_1 only joins to the base layer, R_2 joins to the base layer and the first enhancement layer, R_3 joins to the base, first and second enhancement layers and so on. This makes it possible to implement fine grain rate control as each receiver triggers the rate variation of an enhancement layer. Moreover, as the session size is kept small, the feedback frequency can be set quite high without the overall control traffic being too large. In addition, this scenario makes it possible to tune the mode selection coding process of a given layer on the basis of the Elliott–Gilbert models from those receivers which actually joined the layer.

In the case of multicast applications involving a high number of participants, it is no longer possible to apply the above scheme. We then envision to classify the receivers according to the rate limitation they calculate using the rate prediction model, into a small number of classes which correspond to different enhancement layers. The rate control of each layer would then be performed by aggregating the various rate indications inside the corresponding class, for instance using the mean rate, and so would be the level of error protection provided by the mode selection process.

8. Conclusion

Internet communications traditionally face congestion artifacts. With respect to real-time video transmission, congestion must be addressed in two ways: packet loss resilience and packet loss avoidance (which translates into congestion and rate control). Moreover, the current Internet (with best-effort network service) requires data sources to behave fairly, that is, to harmoniously share network resources, as TCP does. Future improvements of the Internet service model (i.e. differentiated service and/or integrated service) may well not remove this requirement, for instance when individual communications are aggregated into service classes which do not control resource usage competition. The proposed coding mode selection mechanism, based on a channel characterization using an Elliott–Gilbert model, contributes to increase intrinsic robustness of the compressed video stream. Jointly exploiting the knowledge of scene activity and channel characteristics leads to better trade-off between coding efficiency and resilience to packet loss. This paper also describes a rate control scheme which embeds a TCP-friendly rate prediction model. Experimentation results show that the coding mode selection algorithm improves conditional replenishment in terms of compression efficiency, yet is more robust to packet losses than an MPEG-4 video encoder. As for the congestion control scheme, our experimental results shed the light on the benefits of being TCP-friendly, as a better bandwidth usage is achieved when using a CBR, not congestion-responsive video source. Coupling the two strategies developed in this work allows to globally refine a video transmission scenario in terms of provided service and social behaviour of the video source.

Some complementary work may allow an extension of the developed techniques to a multicast scheme, with time-varying channel characteristics.

Acknowledgements

The authors would like to thank Jean-Chrysostome Bolot from INRIA Sophia in France, for helpful discussions, and Franck Galpin for providing an efficient software MPEG-4 decoder.

References

[1] J.C. Bolot, T. Turletti, A rate control mechanism for packet video in the internet, in: IEEE Infocom'94, Vol. 3, Toronto, Canada, June 1994, pp. 1216–1223.

[2] J.C. Bolot, T. Turletti, Adaptive error control for packet video in the internet, in: Proceedings IEEE ICIP'96, Vol. 1, Lausanne, September 1996, pp. 25–28.

[3] J.C. Bolot, T. Turletti, I. Wakeman, Scalable feedback control for multicast video distribution in the internet, in: ACM SIGCOMM'94, London, UK, September 1994, pp. 58–67.

[4] J.C. Bolot, A. Vega-Garcia, Control mechanisms for packet audio in the Internet, in: Proceedings IEEE Infocom'96, Vol. 1, San Francisco, CA, April 1996, pp. 232–239.

[5] T.B. Brown, P.E. Cantrell, J.D. Gibson, Multicast layered video teleconferencing: Overcoming bandwidth heterogeneity, in: First Annual Telecommunication Conference, Austin, TX, 1996, pp. 145–152.

[6] I. Busse, B. Deffner, H. Schultzrinne, Dynamic QoS control of multimedia applications based on RTP, Research Report, GMD-Fokus, Hardenbergplatz 2, D-10623 Berlin, May 1995.

[7] S.Y. Cheung, M.H. Ammar, X. Li, On the use of destination set grouping to improve fairness in multicast video distribution, in: IEEE Infocom'96, Vol. 2, San Francisco, CA, March 1996, pp. 553–560.

[8] G. Côté, B. Erol, M. Gallant, F. Kossentini, H.263 + : video coding at low bit rates, IEEE Trans. Circuit Systems Video Technol. 8 (6) (November 1998) 849–866.

[9] R.O. Hinds, T.N. Pappas, J.S. Lim, Joint block-based video source/channel coding for packet-switched networks, in: Proceedings SPIE Visual Communication and Image Processing, Vol. 3309, 1997, pp. 124–133.

[10] ISO/IEC 14496-2, Coding of audio visual objects: Visual, ISO/IEC JTC1/SC29/WG11, Tokyo, March 1998.

[11] ISO/IEC JTC1/SC29/WG11, MPEG-4 video verification model 10.0, Technical Report, ISO/IEC JTC1/SC29/WG11, February 1998.

[12] W. Kwok, H. Sun, L. Ju, Obtaining an upper bound in MPEG coding performance from jointly optimizing coding mode decisions and rate control, in: Proceedings SPIE VCIP, Vol. 2501, 1995.

[13] J. Lee, B.W. Dickinson, Joint optimization of frame type selection and bit allocation for MPEG video encoders, in: Proceedings ICIP, Vol, 2, 1994, pp. 962–966.

[14] C.G. Liu, D. Estrin, S. Shenker, L. Zhang, Local error recovery in scalable reliable multicast: comparison of two approaches, Technical Report 97-648, USC, January 1997.

[15] J. Mahdavi, S. Floyd, TCP-friendly unicast rate-based flow control, Technical note sent to the end2end-interest mailing list, January 8, 1997.

[16] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The macroscopic behaviour of the TCP congestion avoidance algorithm, Comput. Comm. Rev. 27 (3) (July 1997) 67–82.

[17] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, in: ACM SIGCOMM'96, Stanford, CA, August 1996, pp. 117–130.

[18] S. McCanne, M. Vetterli, V. Jacobson, Low-complexity video coding for receiver-driven layered multicast, IEEE J. Selected Areas Comm. 15 (6) (August 1997) 983–1001.

[19] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, Technical Report TR98-008, UMASS CMPSCI, February 1998.

[20] S. Pingali, D. Towsley, J. Kurose, A comparison of sender-initiated and receiver-initiated reliable multicast protocols, Performance Evaluation Rev. 22 (May 1994) 221–230.

[21] K. Ramchandran, Joint optimization techniques in image and video coding with applications to multiresolution digital broadcast, Ph.D. Thesis, Columbia University, 1993.

[22] R. Rejaie, M. Handley, D. Estrin, RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the Internet, Technical Report 98-681, Computer Science Department, USC, August 1998.

[23] J. Rosenberg, H. Schulzrinne, An RTP payload format for generic forward error correction, IETF Internet Draft draft-ietf-avt-fec-03, July 1998, work in progress.

[24] J. Rosenberg, H. Schultzrinne, Timer reconsideration for enhanced RTP scalability, in: IEEE Infocom'98, Vol. 1, San Francisco, CA, 1998, pp. 233–241.

[25] J. Sandvoss, J. Winckler, H. Wittig, Network Layer Scaling: congestion control in multimedia communication with heterogeneous networks, Technical Report 43.9401, IBM European Networking Center, Heidelberg, 1994.

[26] H. Schultzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, Request for Comments 1889, IETF Network Working Group, January 1996.

[27] H. Schultzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, IETF Internet Draft draft-avt-rtp-new-01, August 1998, work in progress.

[28] Y. Shoham, A. Gersho, Efficient bit allocation for an arbitrary set of quantizers, IEEE Trans. Acoust. Speech Signal Process. 38 (9) (September 1989) 1445–1453.

[29] W.R. Stevens, TCP/IP Illustrated, Vol. 1 – The Protocols, Addison-Wesley Professional Computing Series, Addison-Wesley, Reading, MA, 1994.

[30] W.T. Tan, A. Zakhor, Internet video using error resilient scalable compression and cooperative transport protocol, in: Proceedings ICIP, Vol. 3, October 1998, pp. 458–462.

[31] F. Toutain, TCP-friendly point-to-point video-like source rate control, In: Packet Video'99, March 1999, pp. 1–10.

[32] T. Turletti, S. Fosse-Parisis, J.C. Bolot, Experiments with a layered transmission scheme over the internet, Technical Report RR-3296, INRIA Sophia-Antipolis, 1998.

[33] J. Wen, J. Villasenor, Reversible variable length codes for efficient and robust image and video coding, in: Proc. IEEE Data Compression Conf., Snowbird, UT, March 1998, pp. 471–480.

[34] T. Wiegand, M. Lightstone, D. Mukerjee, T.G. Campbell, S.K. Mitra, Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard, IEEE Trans. Circuits Systems Video Technol. 6 (2) (April 1996) 182–190.

[35] M.H. Willebeek-LeMair, Z.Y. Shae, Robust H.263 video coding for transmission over the internet, in: Proceedings IEEE Infocom'98, Vol. 1, April 1998, pp. 225–232.

[36] R. Yavatkar, L. Manoj, Optimistic strategies for large-scale dissemination of multimedia information, in: ACM Multimedia'93, Anaheim, CA, August 1993.