Variable Time Scale Multimedia Streaming Over IP Networks

Enrico Masala, *Member, IEEE*, Davide Quaglia, *Member, IEEE*, and Juan Carlos De Martin, *Member, IEEE*

***Abstract*—This paper presents a comprehensive analysis of a variable time-scale streaming technique, VTSS, according to which rate changes are obtained by varying the inter-packet transmission interval, rather than altering, as in most cases, the source coding rate. Instead of constraining the transmitter to operate in real-time, the time scale of the packet scheduler can vary between zero, when the network is congested, to as faster than real-time as the channel bandwidth allows, when the network is lightly loaded. Although this approach is reportedly used in commercial streaming products, so far the technique has not yet been analyzed in a rigorous fashion, nor it has been compared to other state-of-the-art streaming techniques. This work first presents a theoretical analysis of the performance achievable by the VTSS approach, and it shows that, for the same channel conditions, VTSS yields a total distortion which is lower or, in the worst case, equal than the distortion of the standard real-time source-rate adaptive approach. A lower bound on receiver buffer size is also derived. Network simulations then analyze the performance of a TCP-friendly test implementation of VTSS compared with an *ideal* real-time source rate-adaptive technique, whose performance, being ideal, represents the upper bound of any transmission scheme based on source rate adaptation. The simulation results, also based on actual network traces, show that the VTSS approach delivers higher perceptual quality (up to 1.2 dB PSNR in the considered scenarios) and reduced video quality fluctuations (1.6 dB standard deviation PSNR, instead of 4.9 dB) for a wide range of standard video sequences. Perceptual quality evaluation by means of PVQM confirms such results. The gains, as expected, are even more pronounced (7.6 dB PSNR on average) if compared to real-time constant bit-rate video transmission.**

***Index Terms*—Rate adaptation, variable time-scale streaming, video communication.**

I. Introduction

MULTIMEDIA applications continue to be at the center of an extraordinary deal of attention. From voice over IP—which is radically changing telephony—to video streaming—which enables, among other things, hugely popular video sharing sites—multimedia communications are the focus of the efforts of engineers and researchers worldwide.

However, for this appealing class of applications to succeed, several major problems need to be solved. Among them, perhaps the most challenging issue is how to best deal with the strongly time-varying nature of IP networks, both wired [1] and, even more so, wireless [2]. Many proposals have been made to address the problem of time-varying bandwidth, loss rates and delays. They range from physical-layer solutions (e.g., adaptive modulation schemes [3]) to application-layer approaches (e.g., joint source-channel coding [4], [5]). Other studies suggest that end-to-end flow control should be adopted by multimedia flows to prevent congestion and unfair use of network resources [6]; thus, a large number of *rate-adaptive approaches* have been proposed, in which the transmission rate is typically adapted to match the estimated value of the instantaneous channel capacity.

Rate-adaptive techniques for multimedia communications are usually based on real-time *source rate* adaptation. The key idea behind this class of algorithms is that, in case of congestion, the sender reduces the amount of traffic sent to the network by lowering the source encoding rate. The reduced rate eases the network congestion, therefore reducing the likelihood of packet losses, with a final effect on quality that overcomes, in the ideal case, the higher distortion caused by the lower source rate [7]. This approach is, however, limited in two main ways. First, the encoding bitrate is restricted to a given rate range which depends on the compression algorithm. When the source rate cannot descend any lower, potentially heavy losses occur; when the source rate cannot go any higher, network resources are not fully exploited. The second limitation regards the multimedia encoding quality, which varies according to the fluctuations of network conditions. However, source rate adaptation algorithms remain the main viable option to ensure the highest possible quality for the crucial class of interactive multimedia communications, such as voice over IP and videoconferencing.

Changing class of applications, the noninteractive, delay-tolerant nature of multimedia streaming allow to consider other options, besides source rate adaptation, to match the rate of multimedia streams to the instantaneous channel capacity. This work analyzes the specific idea of decoupling the transmission rate from the playback rate. Streaming systems, in fact, usually transmit, for simplicity's sake, video or audio frames in real-time, that is, at approximately the same rate at which they will be decoded and presented to the user, just like interactive applications, such as voice over IP. For instance, many speech communication applications transmit a frame (i.e., a packet) every 20 ms, which is also precisely the schedule followed by the speech decoder.

According, instead, to the approach analyzed in this paper, hereafter referred to as variable time-scale streaming (VTSS), the source rate remains unchanged, but the packet scheduler is free to change its instantaneous transmission rate from zero (i.e.,

Manuscript received December 14, 2007; revised July 01, 2008. Current version published December 10, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Marco Roccetti.

E. Masala and J. C. De Martin are with the Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy (e-mail: masala@polito.it; demartin@polito.it).

D. Quaglia is with the Computer Science Department, University of Verona, Verona, Italy (e-mail: davide.quaglia@univr.it).

Digital Object Identifier 10.1109/TMM.2008.2007284

the transmission pauses) when the channel capacity is low, to faster than real-time as the channel bandwidth allows. The decoder playback rate remains, of course, unchanged. VTSS is independent of the specific source coding algorithm and thus it can be applied to pre-compressed multimedia content; furthermore it is certainly conceivable to combine both approaches, i.e., VTSS and source coding rate adaptation, to achieve even greater flexibility and performance.

Several works (see [8] for a survey) propose to send frames ahead of schedule (with respect to their playback time), but not to maximize perceptual quality, rather to smooth a variable-bit-rate stream for transmission over a constant-bit-rate channel [9], e.g., to improve the statistical multiplexing gain [10]. Such smoothing approach is applied to streaming of both pre-compressed and live video [11], [12]. The rate adaptation protocol (RAP) is based, as VTSS, on the idea of varying the inter-packet gap to adjust transmission speed [13]; that work, however, mainly focuses on the end-to-end congestion control rather than on analyzing multimedia quality. In [14], a variable transmission rate technique that optimizes bandwidth usage by a streaming server, while controlling the buffer fullness of the clients, is studied; however, the case in which the transmission rate is zero is not considered, nor, more relevantly, the perceptual quality experienced by the end-users. Some industrial streaming solutions, such as the Windows Media and Real Networks systems, are also reported to vary the packet sending rate, particularly at the beginning of the transmission to fill the playout buffer; an analysis of their behavior is, however, not possible due to the confidentiality surrounding the algorithms used by such applications. Finally, according to Odlyzko, faster-than-realtime file transfers represents the most efficient way of delivering multimedia traffic [15].

This work studies the variable time-scale streaming approach, VTSS, first formulating the problem analytically, then providing results about the perceptual quality obtainable by both VTSS and other reference techniques. More specifically, a comparison has been made between VTSS and regular constant bit rate (CBR) transmission and between VTSS and an *ideal* implementation of the real-time source rate-adaptive approach, whose performance, being ideal, represents the upper bound of any transmission system based on source rate adaptation. The performance of the VTSS approach is experimentally assessed through a specific VTSS test implementation based on the same end-to-end control of transmission rate of the TCP protocol. Performance is studied by means of NS-2 network simulations, using H.264 test video sequences and objective measures of video quality. This paper substantially extends and completes the work presented in [16] with a theoretical analysis of the performance achievable by the VTSS approach and the receiver buffer size requirements. Moreover, the performance of the VTSS test implementation is assessed in realistic scenarios by means of actual network traces, and the impact of limiting the client buffer size is also investigated. A discussion of the state-of-the-art literature on this subject is also presented.

The paper is organized as follows. Section II reports on the literature in this field of research. Section III introduces the video streaming scenario and presents the VTSS approach. Section IV shows that the VTSS approach can achieve a distortion which is lower or equal than the distortion achieved by the best source rate-adaptive technique, under the same network conditions and analyzes the receiver buffer requirements. The simulation setup is described in Section V. Simulation results comparing the VTSS approach with other reference techniques are reported and discussed in Section VI. Finally, conclusions are drawn in Section VII.

II. Related Works

Most of the early approaches to real-time multimedia communications mainly focused on reserving network resources on the basis of peak bandwidth requirements [17]. Network congestion was avoided by using preventive call admission control mechanisms based on such peak bandwidth requirements [18]. This approach, however, sacrifices statistical multiplexing gain.

Currently, the problem of providing good perceptual performance in multimedia communications is better described in terms of transmitting only the data that fit network capacity at any given time. Thus, adaptation mechanisms have been developed to consider both network conditions and the specific characteristics of multimedia data. Networking issues are usually addressed by monitoring specific network parameters (e.g., bandwidth, delay, jitter) and then estimating an optimal transmission rate on the basis of such parameters. Regarding multimedia data, source rate shaping techniques may be used to adapt the source coding rate to the available network capacity. Section II-A presents a survey of rate estimation algorithms, whereas Section II-B focuses on rate adaptation algorithms.

A. Rate Estimation Algorithms

Rate estimation algorithms strive to determine the optimal transmission rate as a function of the capacity of the channel; channel capacity depends on concurrent traffic load and, in case of wireless links, also on physical conditions such as interference, path loss, and fading. Since channel capacity is not known in advance and it can rapidly change, its value should be estimated at the transmitter by means of feedback information from the network [7], [19] or the receiver.

Receiver-based feedbacks are the basis of the so-called end-to-end rate estimation algorithms which match the Internet's philosophy at concentrating complexity into the end nodes, keeping internal nodes as simple as possible. TCP can be considered the most important end-to-end rate-adaptive Internet protocol. Each source maintains a counter (named congestion window) indicating the number of bytes that can be transmitted; the size of the congestion window is changed according to experienced packet losses and round-trip time. The end-to-end congestion control of TCP has two purposes [20], i.e., *1)* recovery of packet losses and *2)* the fair allocation of available bandwidth among various sources. TCP's behavior is a reference for the design of multimedia transmission systems, since it is widely regarded as desirable that multimedia flows use resources fairly with respect to traditional TCP flows (TCP-friendliness) [21]–[23]. However, it is widely accepted that TCP is not suitable for transmitting real-time multimedia

[24], [25]. Therefore, several proposals that modify TCP for multimedia applications have been advanced in recent years.

In [26] and [27], TCP-like protocols without data retransmission are implemented; each application PDU is sent as an independent segment or a set of segments. Mukherjee and Brecht propose a TCP modification for continuous media, called Timelined TCP, in which a deadline is specified for each packet, to ensure timely delivery [28]. However, all cited works do not address either perceptual quality or playout buffer behavior.

The *additive increase multiplicative decrease* (AIMD) approach provides good TCP-friendliness since it is the basis of the TCP congestion control mechanism: transmission rate is increased by an additive value when the channel is good and it is decreased by a multiplicative factor in case of packet loss. Bolot and Turletti [29] describe a feedback control mechanism for the transmission of variable bitrate video in which the output rate of the coder is adjusted in response to periodic reports on packet losses at the receiver; source rate is halved if the median loss rate exceeds a given threshold, otherwise it is increased by a fixed fraction of its current value. Also Busse, Deffner, and Schulzrinne describe an end-to-end control system in which transmission rate is decreased by a multiplicative factor if the short-term loss average exceeds a given threshold, otherwise it is increased by an additive term [30]. Variants of the AIMD approach are also proposed to avoid drastic reductions in transmission rate, which are generally problematic for multimedia applications, while still maintaining a certain degree of TCP-friendliness [31], [32]. In [33] and [34], the delay is also monitored to foresee the beginning of congestion.

Model-based rate estimation algorithms use a model of TCP which gives the throughput of an ideal TCP sender as a function of packet loss rate and round trip time. If the output rate of the multimedia transmission follows this throughput profile then fairness and TCP-friendliness are achieved [35], [24], [36]–[39]. In [40] the loss-delay based adjustment algorithm (LDA) relies on the enhanced version of the end-to-end real-time transport protocol (RTP) to determine the bottleneck bandwidth of a connection; the bottleneck bandwidth is then used for dynamically determining the adaptation parameters.

Feedback information can be in the form of either periodic RTCP receiver reports [30], [40], [33], [34] or packet-level acknowledgments [13].

At the network standardization level, the Internet Engineering Task Force has recently standardized the Datagram Congestion Control Protocol to transport continuous media over the Internet [41]; the new protocol supports several of the end-to-end flow control algorithms cited above.

B. Source Rate Shaping Algorithms

Rate-shaping algorithms adjust the transmission rate of the multimedia source to match the rate determined during the rate estimation process. Rate shaping algorithms strongly depend on the kind of source, i.e., speech, audio, or video, due to the fundamental differences among the respective compression techniques. Early works addressed adaptive transmission of speech [42], [33]. Video coding rate was first varied by adjusting the quantization stepsize [29], [7], [19], [34], [43], [44], [38]; it is well known, in fact, that the higher is the quantization stepsize, the lower are both output bitrate and quality [45]. Since the quantization stepsize is adjusted at the encoding time, this method can only be applied to the streaming of live video. A possible solution for pre-encoded video is to produce different versions of the same sequence at different bitrates [46], [34]. The resulting bitstreams, however, can be switched only at key frames, otherwise the prediction loop may break. To provide efficient switching points in the sequence, H.264 video coding standard has introduced a new type of picture, called switching predictive (SP), to provide efficient switching points in the sequence [47]. Also re-encoding with a different quantization stepsize can be performed to reduce the bitrate of pre-compressed streams, at the price, however, of a considerable increase in terms of CPU usage, encoding time and coding distortion [48].

In case of live streams or re-encoding, another way to reduce bit rate is by avoiding to encode some parts of the video as in case of macroblock skipping, that is, frame areas with low motion activity are not coded [29]. Other techniques in this context are low-pass filtering, color reduction, and change of coding format; for an overview, refer to, e.g., [48].

Another way to vary the source coding rate is to use layered-encoded video. The compressed input signal is organized in a set of layers arranged in a hierarchy that provides progressive refinements (from the points of view of spatial/temporal resolution or quality) [49]–[52]. Adaptation can be performed by dropping layers both at the transmitter side and in proxy caches [53], [38]. In case of multicast transmission each layer can be transmitted over a different multicast group and the rate is chosen by each receiver by subscribing to a different number of multicast groups [54]. When two or more independent transmission channels are available, the source signal can also be profitably encoded and transmitted using multiple description coding (MDC) techniques [55]–[57]. Unlike layered coding, in fact, for which the core layer must be received in all cases, in the case of MDC, missing descriptions do not impair the ability of the receiver to decode the other, correctly received descriptions. It is worth noting that the benefits of scalable coding (be that layered or MD) are paid in terms of reduced coding efficiency with respect to non-scalable multimedia encoding.

Rate shaping of pre-compressed streams can also be performed by dropping some portions of the compressed stream such as frames [29], [58], [48], [59]–[61] or high-frequency DCT coefficients [27]. However, in these cases, considerable distortion might enter the prediction loop.

III. Variable Time Scale Streaming (VTSS)

The first step of the streaming process is to encode the source signal. In case of video, compression is performed on frames, that is, segments of input data, which we will call, for the sake of generality, *presentation units* (PU). All the data belonging to the same PU are played back at the same time during the decoding process. Thus each PU is associated with a unique presentation time. Each PU may be encapsulated in one or more *transmission units* (TU), i.e., one or more packets, that share the same presentation time. Each TU is then transmitted and buffered at the receiver where it will be reassembled into PU's,

decoded and presented to the user. All the TU's that constitute a PU must be received before the PU's presentation time.

Let $t_{i,j}$ be the time instant at which the j-th TU of the i-th PU is sent, and let τ_i be the presentation time of the i-th PU. Let $\delta_{i,j}$ and $s_{i,j}$ be the transmission delay and the TU size, respectively. To allow error-free decoding at the receiver, the following straightforward timing condition must hold for each TU:

$$t_{i,j} + \delta_{i,j} \leq \tau_i. \tag{1}$$

If the instantaneous available channel capacity along the path between the source and the destination is indicated by $C(t)$, each TU can be successfully transmitted only if

$$s_{i,j} \leq \int_{t_{i,j}}^{t'_{i,j}} C(u)\mathrm{d}u \tag{2}$$

where $t'_{i,j} = t_{i,j+1}$ or $t'_{i,j} = t_{i+1,1}$ depending on the number of TUs in the i-th PU.

A scheduling strategy determines the transmission instants $t_{i,j}$. A simple choice is to uniformly distribute the TU's between the PU's creation instants. A different approach consists in sending all the TUs of a certain PU consecutively right after the PU creation. In the first case, the rate offered to the network is more uniform, whereas in the latter case the probability to receive the TU's in time is higher, at the price of a spiky rate profile.

The encoding and packetization strategy determines TU's number and sizes, the latter indicated by the numbers $s_{i,j}$. The $s_{i,j}$ values are generally determined at encoding time, but can also be adjusted at later times, as explained in Section II-B.

Regardless of the encoding strategy, PU's are generally transmitted on the network at approximately the same rate at which they will be decoded and presented to the user. For instance, a video streaming application offering 20-frame-per-second video, will transmit with the same scheduling, i.e., packets corresponding to 20 frames per second. We refer to this approach as *real-time transmission*. Assuming that the first TU of each PU is transmitted at the beginning of each PU time frame, real-time transmission is described by the following timing condition:

$$t_{i+1,1} - t_{i,1} = \tau_{i+1} - \tau_i. \tag{3}$$

The VTSS approach, instead, is based on the concept of varying the transmission rate of the TU's, i.e., the $t_{i,j}$ instants, according to the instantaneous network conditions. Hence, the TU's transmission rate may range from zero (i.e., the transmission pauses) to as high as the available channel capacity allows, in which case the equality holds in (2).

Consequently, PU's may be *locally* more closely set in time, or more apart from each other, than in the real-time case. Fig. 1 shows examples of TU transmission schedules for both real-time and VTSS streaming. Fig. 1(a) represents TU transmission times in the case of standard real-time streaming: TU's may be distributed as desired along their PU time frame, but the PU's timing is constant. Fig. 1(b) and (c) show VTSS transmission examples, where the PU's instants are not uniformly distributed. Fig. 1(b) depicts a VTSS transmission that ends earlier than real-time, whereas in Fig. 1(c) the VTSS transmission happens to end, as it may, at the same time as a real-time transmission, but during the first part of the stream PU's were sent faster than real-time, whereas the second part registers a strong slow-down around PU 6.

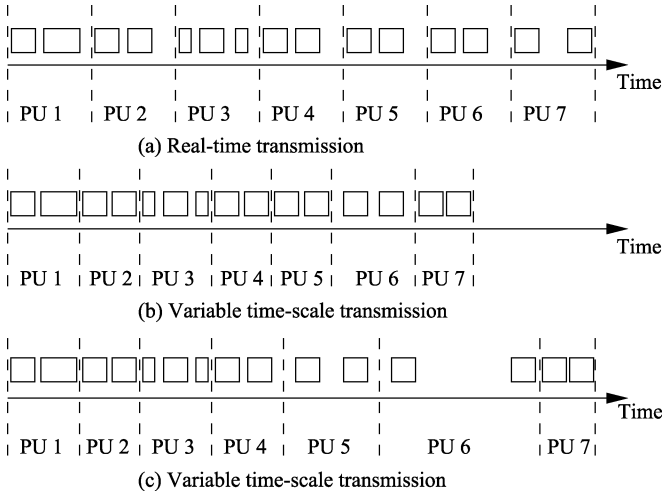

Fig. 1. Examples of TU's transmission schedule for (a) standard real-time streaming (PUs uniformly spaced in time), (b) VTSS with faster than real-time transmission, and (c) VTSS with overall real-time behavior [the transmission ends at the same time as (a)].

We introduce the *instantaneous time-scale coefficient* ρ_i, defined as follows:

$$\rho_i = \frac{\tau_{i+1} - \tau_i}{t_{i+1,1} - t_{i,1}}. \tag{4}$$

When condition (3) holds, then the transmission is carried out in *real-time*, and $\rho_i = 1$. If $\rho_i > 1$, the transmitter is sending packets at a rate higher than the receiver's consumption rate, i.e., the transmission is happening faster than real-time. If $\rho_i < 1$, the receiver is consuming data faster than they are sent, down to the case of $\rho_i \to 0$ when no data are sent and playback continues until the receiver buffer is empty.

Fig. 2 shows the block diagram of a video streaming system using the VTSS approach. In the remainder of this paper, for simplicity's sake, the terms "TU" and "packet" will be used interchangeably, as well as "PU" and "frame". With the VTSS approach the pacing of the packets is determined by the packet scheduler to achieve the time-scale coefficient ρ_i chosen by a time-scale selection algorithm on the basis of channel state information. At the receiver, data packets are retained into the playout buffer until they are decoded and presented to the user. The channel monitor determines the instantaneous channel state using one of the methods presented in Section II-A. The status information is sent back to the time-scale selection algorithm which modifies the time-scale coefficient according to the estimated instantaneous channel capacity.

The playout buffer should handle the potentially large rate variations that characterize the VTSS technique. The maximum allowed rate variation is, in fact, determined by the size of the playout buffer. Therefore, an important aspect of the VTSS design is to choose the time-scale coefficient according to the instantaneous conditions of the receiver-side playout buffer. If the time-scale coefficient has been less than one for a long time, the

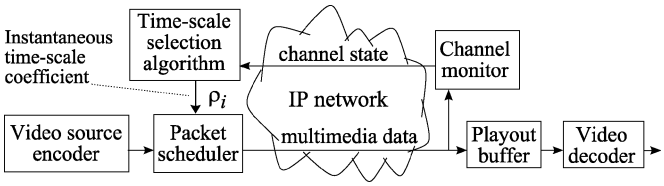

Fig. 2. Block diagram of a VTSS video transmission system.

playout buffer may soon empty, causing a potentially disrupting playback gap. To deal with this case, however, the VTSS transmission strategy could be coupled with a standard source rate adaptation mechanism so that the amount of transmitted data is reduced when the playout buffer is nearly empty and channel bandwidth is scarce.

The VTSS approach can be applied to any multimedia coding standard and may be used in conjunction with any kind of source coder. Source coding can be performed as desired, e.g., at constant playback quality, as is the case of most entertainment-quality applications. In real-time source rate adaptive approaches, instead, the source rate is dictated by the instantaneous channel quality. Therefore, it may happen that, in certain time intervals, the video signal requires a high encoding bit rate to ensure good video quality, but the available bandwidth is low: in those cases, the users experience poor video quality. If channel bandwidth is abundant, instead, and the video signal characteristics allow good encoding quality even at relatively low bit rates, channel capacity cannot be efficiently exploited by the real-time source rate adaptive approach since it is useless to increase the source encoding rate over a given threshold due to saturation effects.

Finally, differently from the real-time source rate adaptive approaches, the VTSS approach can even suspend the transmission if network is heavily congested, whereas real-time source rate adaptive techniques are restricted to minimum rates which depend on the specific compression algorithm. Therefore, the source rate cannot be reduced under a certain threshold, which may however be still too high, thereby causing heavy losses and prolonged network congestion.

A VTSS system may, therefore, easily implement the desired trade-offs between several factors, including peak network occupancy, receiver-side buffer size, maximum tolerable playout delay, and TCP-friendliness. Moreover, compared to real-time source rate adaptive approaches, it also avoids fluctuations of multimedia encoding quality in case of varying network conditions.

IV. VTSS Performance Analysis

Let us assume that the available channel bandwidth as a function of time, $C(t)$, is known. We will demonstrate that a video sequence transmitted with the VTSS technique can achieve a distortion which is lower or, at worst, equal to the distortion achieved by the best source rate-adaptive technique, under the same network conditions and under the assumption for simplicity's sake that no packet losses occur.

In the following, we use s_i to indicate the size of the i-th frame. For simplicity's sake, we assume that the transmission starts at time zero. The encoding optimization problem for the real-time source rate adaptive approach can be stated as follows:

$$\min_{\{s_i\}} D = \sum_{i=0}^{N-1} d_i(s_i)$$
$$\text{s. t.} : s_i \le \int_{iT}^{(i+1)T} C(t)\mathrm{d}t, \quad \forall i \in \{0,1,\ldots,N-1\} \quad (5)$$

where $d_i(s_i)$ is the encoding distortion for the i-th frame (for simplicity's sake, a function of the frame size s_i only), N is the number of frames in the sequence and T is the inverse of the frame rate. The constraints ensure that the channel bandwidth is sufficient to transmit each frame i. Using the same notation, the VTSS approach may be formulated as

$$\min_{\{s_i\}} D = \sum_{i=0}^{N-1} d_i(s_i)$$
$$\text{s. t.} : \sum_{i=0}^{k} s_i \le \int_{0}^{(k+1)T} C(t)\mathrm{d}t, \ \ \forall k \in \{0,1,\ldots,N-1\}. \quad (6)$$

Except for $k = 0$, the constraints in (6) do not directly impose a maximum size for a single frame. However, at time instants which are multiples of T, all past frames must have been transmitted, hence their cumulative size is limited.

To clarify the implications of the constraints in (6) on the transmission in terms of the instantaneous time-scale coefficient ρ_i defined in (4), we express such constraints as a function of ρ_i. First, note that

$$t_{i+1,1} - t_{i,1} = \frac{\tau_{i+1} - \tau_i}{\rho_i} = \frac{T}{\rho_i}. \quad (7)$$

Then, $t_{i,1}$ and $t_{i+1,1}$ are chosen such that

$$s_i = \int_{t_{i,1}}^{t_{i+1,1}} C(t)\mathrm{d}t \quad (8)$$

to completely use channel bandwidth. Therefore, by using (7)

$$\sum_{i=0}^{k} s_i = \sum_{i=0}^{k} \int_{t_{i,1}}^{t_{i+1,1}} C(t)\mathrm{d}t = \int_{t_{0,1}}^{t_{0,1}+\sum_{i=0}^{k} T/\rho_i} C(t)\mathrm{d}t \quad (9)$$

since integration intervals are adjacent. Assuming that the transmission starts at time zero, i.e., $t_{0,1} = 0$, the constraints in (6) can be expressed as

$$\int_0^{\sum_{i=0}^{k} T/\rho_i} C(t)\mathrm{d}t \le \int_0^{(k+1)T} C(t)\mathrm{d}t, \ \forall k \in \{0,1,\ldots,N-1\} \quad (10)$$

since $C(t)$ is a non-negative function. Therefore:

$$\sum_{i=0}^{k} \frac{T}{\rho_i} \le (k+1)T, \quad \forall k \in \{0,1,\ldots,N-1\}. \quad (11)$$

Equation (11) implies that

$$\begin{cases} \rho_0 \geq 1 \\ \rho_k \geq \frac{1}{1+k-\sum_{i=0}^{k-1}\frac{1}{\rho_i}}, \ \forall k \in \{1,\ldots,N-1\}. \end{cases} \quad (12)$$

The previous equation expresses the constraints on the instantaneous time-scale coefficient ρ_k as a function of the values of the past instantaneous time-scale coefficients. Except for the transmission of the first frame which, of course, requires no less than real-time transmission, the minimum value of the instantaneous time-scale coefficient which ensures the correct transmission of a given frame k is bounded and it depends on the values of the instantaneous time-scale coefficient used for previous frames: the greater the instantaneous time-scale coefficients for previous frames, the lower the allowed instantaneous time-scale coefficient for the current one. Note also that (12) never imposes an instantaneous time-scale coefficient ρ_k greater than one, provided that the constraint for $k-1$ is satisfied. This can easily be proven by substituting that constraint into (12). This implies that at any time the instantaneous time-scale coefficient can be reduced to one, i.e., a faster than real-time transmission can always be slowed down and carried out in real-time. This observation is important for the case of a limited receiver buffer size, as shown at the end of this section.

In the following part we focus on determining the s_i values which solve the problems posed by (5) and (6). Note that the $d_i(\cdot)$ functions are the same in both cases if the same video sequence is considered.

Lemma 1: For the same channel conditions, any set $\{s_i\}$ that satisfies the constraints of (5) also satisfies the constraints of (6).

Proof: Let $\{s_i\}$ be a set that satisfies (5). We sum each constraint expressed by (5) for which $i \leq k$

$$\sum_{i=0}^{k} s_i \leq \sum_{i=0}^{k} \int_{iT}^{(i+1)T} C(t)\mathrm{d}t. \quad (13)$$

The integrals can be easily summed because the integration intervals are adjacent, yielding

$$\sum_{i=0}^{k} s_i \leq \int_{0}^{(k+1)T} C(t)\mathrm{d}t. \quad (14)$$

Since k is arbitrarily chosen, those sets $\{s_i\}$ that satisfy the constraints of the real-time source rate adaptive optimization problem (5) also satisfies the constraints of the VTSS optimization problem (6). □

Lemma 2: For the same video sequence and the same channel conditions, there exists at least one optimization problem in the form of (5) such that its optimal solution D is suboptimal if the constraints are replaced with the constraints of (6).

Proof: The proof will show how to construct such problem. Since the problem can be arbitrarily chosen, we assume that $d_i(\cdot)$ are monotonically decreasing functions of s_i. We choose the set $\{s_i\}$ such that

$$s_i = \int_{iT}^{(i+1)T} C(t)\mathrm{d}t, \quad \forall i \in \{0,1,\ldots,N-1\}. \quad (15)$$

This is the optimal solution of problem (5) because each term of the summation is positive, each term only depends on a single s_i value and all constraints are satisfied at equality. Let $D_A = \sum_{i=0}^{N-1} d_i(s_i)$ be the corresponding minimum distortion. Now we construct the set $\{s'_i\}$ as follows:

$$s'_i = \begin{cases} s_i, & \text{if } i \neq m \text{ and } i \neq m+1 \\ s_i - \epsilon, & \text{if } i = m \\ s_i + \epsilon, & \text{if } i = m+1 \end{cases} \quad (16)$$

where $m \in \{0,1,\ldots,N-2\}$ and $0 < \epsilon < \min(s_m, s_{m+1})$ is an arbitrary value. Hence the constraints of (5) are violated if $i = m+1$, but the constraints of (6) continue to hold because

$$\begin{aligned} &\sum_{i=0}^{k} s'_i = \sum_{i=0}^{k} s_i, \quad \forall k \in \{0,1,\ldots,N-1\}, k \neq m \\ &\sum_{i=0}^{k} s'_i < \sum_{i=0}^{k} s_i, \quad \text{if } k = m. \end{aligned} \quad (17)$$

Then we further assume that the chosen $d_m(\cdot)$ and $d_{m+1}(\cdot)$ satisfy the condition

$$d_m(s_m - \epsilon) - d_m(s_m) < d_{m+1}(s_{m+1}) - d_{m+1}(s_{m+1} + \epsilon). \quad (18)$$

It is easy to show that

$$d_m(s_m - \epsilon) + d_{m+1}(s_{m+1} + \epsilon) < d_m(s_m) + d_{m+1}(s_{m+1}) \quad (19)$$

therefore

$$d_m(s'_m) + d_{m+1}(s'_{m+1}) < d_m(s_m) + d_{m+1}(s_{m+1}) \quad (20)$$

which implies

$$\sum_{i=0}^{N-1} d_i(s'_i) < \sum_{i=0}^{N-1} d_i(s_i). \quad (21)$$

Hence, the set $\{s'_i\}$ satisfies the constraints of (6) while yielding a total distortion which is strictly lower than D_A. □

Theorem 1: For the same sequence and the same channel conditions, the solution of the VTSS problem leads to a total distortion D which is lower or, at worst, equal to the one of the rate-adaptive problem.

Proof: Let D_A be the total distortion achieved by the optimal solution of the rate-adaptive problem. This solution satisfies the constraints of the VTSS problem by Lemma 1, hence a solution of the VTSS problem with total distortion equal to D_A exists. The solution, however, is not guaranteed to be optimal for the VTSS problem by Lemma 2, since there exists at least one rate-adaptive problem whose optimal solution D_A is not the optimal one for the corresponding VTSS problem. □

Therefore, under the hypothesis of no losses, we showed that the minimum total distortion achieved by the real-time source rate adaptive strategy operating in ideal conditions can be lowered, or, at worst, left unchanged, by using the VTSS transmission strategy. Moreover, note that the assumptions about the frame distortion function in Lemma 2, i.e., monotonic decrease as a function of the frame size as well as the conditions of (18), are realistic since they express the typical behavior of the rate-distortion characteristic of a video encoder. Thus, in realistic cases, the solution of the VTSS transmission problem leads

to a distortion which is *strictly* lower that the distortion produced by the solution of the real-time source rate adaptive problem.

In the following, the requirements of the two transmission strategies on the receiver buffer size are investigated. In case of real-time source rate adaptive transmission the receiver buffer size can be determined by considering the expected network jitter only. In case of VTSS transmission, instead, the receiver buffer is also used to store multimedia data when the instantaneous time-scale coefficient is greater than one, i.e., the transmission proceeds faster than real-time. If we assume that the VTSS transmission technique is transmitting the k-th frame, i.e., $t_{k-1} < t < t_k$, the amount of transmitted data up to time t can be expressed as

$$\sum_{i=0}^{k-1} s_i + s_k \frac{t - t_{k,1}}{t_{k+1,1} - t_{k,1}} \leq \int_{t_0}^{t} C(t')\mathrm{d}t', \quad t_{k-1} < t < t_k \tag{22}$$

while the amount of decoded data at the receiver is

$$\sum_{i:\tau_i \leq t} s_i = \sum_{i \leq (t-(t_0+\delta))/T} s_i = \sum_{i=0}^{\lfloor (t-(t_0+\delta))/T \rfloor} s_i \tag{23}$$

where δ is the sum of the transmission and the initial buffering delays at the receiver. Thus, the receiver buffer size B_R needed at a given time instant t can be computed as

$$B_R(t) \geq \sum_{i=0}^{k-1} s_i + s_k \frac{\rho_k (t - t_{k,1})}{T} - \sum_{i=0}^{\lfloor (t-(t_0+\delta))/T \rfloor} s_i, \quad t_{k-1} < t < t_k. \tag{24}$$

Equation (24) imposes a lower bound on the buffer size at the receiver for the VTSS technique, which depends both on the amount of available channel bandwidth and the time elapsed from the beginning of the playback. Note that buffer level can be monitored at the transmitter by reproducing its behavior, i.e., by adding the size of all transmitted frames and by subtracting, as transmission proceeds, the size of each already decoded frame. In case of packet losses and uncertain transmission delay, the transmitter might overestimate but never underestimates the buffer requirement at the receiver. Therefore, the transmitter, if the estimated level of the receiver buffer is too close to the upper limit, could reduce the instantaneous time-scale coefficient to one, i.e., transmit in real-time, to avoid receiver buffer overflow.

V. Simulation Setup

A. VTSS Test Implementation

To test the VTSS approach with actual video material over an IP network, and to validate the theoretical analysis presented above, we designed a specific VTSS implementation in which the rate of the VTSS source has been set to be less than or equal to the rate of a generic TCP transmission in presence of the same interfering traffic. Thus, the specific VTSS implementation used for the tests described in the following is TCP-friendly by definition. Packet scheduling is performed as follows. For each frame, a current frame interval is determined by multiplying the current time-scale coefficient by the real-time duration of one frame. Then, packets are uniformly spaced within that current frame interval and sent according to that schedule. When the last packet of the frame has been sent, the previous steps are repeated for the next video frame using the updated value of the time-scale coefficient.

The implementation of the VTSS approach described above was tested with the *ns* network simulator [62] and a set of widely used video sequences. We used 239 frames of the standard video sequences known as *tempete*, *mobile*, *foreman*, *silent* and *mother & daughter*. The spatial resolution is 352×288 pixel (CIF resolution). The encoding distortion values for the five sequences are reported in Table I.

TABLE I
Encoding PSNR

Video sequence	Avg. PSNR (dB)	PSNR std. dev. (dB)
tempete	34.81	0.70
mobile	33.91	0.69
foreman	37.03	0.93
silent	36.36	0.18
mother & daughter	38.47	0.36

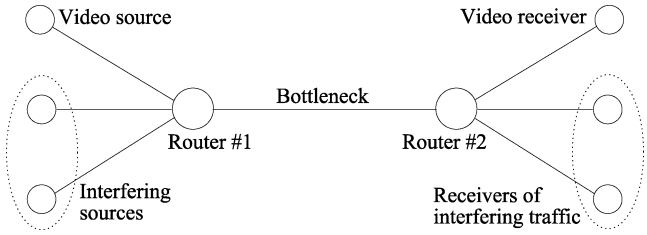

Fig. 3. Topology of the simulated network scenario. The video source competes with other concurrent interfering sources for bottleneck usage.

To achieve statistical significance, the simulations were performed on the sequences concatenated with themselves 30 times, for a total of 7170 frames. At 25 frames per second, the duration of the resulting sequence is 286.8 s. The sequence was encoded using the H.264 reference software version JM 8.0 [63]. The encoding pattern is IPBPB... with an I-picture inserted every 12 pictures; consequently, when the video is transmitted in real-time, a full refresh takes place every 480 ms. The Network Abstraction Layer of H.264 was instructed to place two rows of macroblocks in each packet, with a maximum transmission unit size of 1500 bytes.

B. Network Setup and Test Cases

The network topology features a simple two-node bottleneck, shown in Fig. 3, with a 5-ms source-to-destination propagation delay. The bottleneck link capacity varies. The other links are oversized in bandwidth not to impact on the results.

We consider two different scenarios. In Scenario 1, CBR traffic is injected in the network to vary the amount of available bottleneck bandwidth, whereas in Scenario 2 an actual network trace is used as interfering traffic. In both cases, the channel capacity available for video transmission over the whole simulation time is approximately equal to the average bitrate of the video sequence. For the first scenario, the minimum channel

capacity never drops below the minimum rate of the source encoder for the real-time ideal source rate-adaptive technique, whereas in the second scenario the interfering traffic trace may occupy, for certain time intervals, the whole bottleneck capacity.

The performance of our VTSS sample implementation is compared with two other techniques. The first technique is the ideal real-time source rate-adaptive algorithm, that in our experiments varies the encoding rate to exactly match the available rate on the bottleneck. The technique is ideal since: 1) the available channel rate is assumed to be *exactly* known; 2) encoding rate changes are assumed to happen *instantaneously*; 3) the available channel bandwidth never drops below the *minimum rate* achievable by the source coder (in Scenario 1); 4) the maximum rate achievable by the source coder exactly matches the maximum available bandwidth on the bottleneck so that bandwidth is not wasted; and 5) *no losses* are experienced by the ideal real-time source rate-adaptive flow. Such technique models any packets scheduling technique which sends, in the time interval corresponding to each video frame, a number of bytes for that frame which fully exploits channel capacity. The second reference technique implements a nonadaptive approach that keeps its transmission rate constant (CBR), regardless of network conditions. In this case, the fixed source rate is the same as the VTSS technique. For each frame, packets are uniformly spaced between frames. Due to implementation constraints in the H.264 software, it was not possible to fine-tune the encoding rate for the ideal rate-adaptive technique to exactly match the average encoding rate of the other two techniques. However, to ensure fairness in the comparisons, the ideal rate-adaptive technique has been allowed a slightly higher average bitrate (about 10% higher) compared to the other two techniques; for the *tempete* sequence, for instance, this yields a non-negligible advantage of about 0.66 dB PSNR. Finally, note that the playout buffer, for all techniques, is assumed to be unlimited, except in Section VI-C which investigates the effect of limiting the buffer size on the performance. However, in this case the VTSS algorithm has been modified to keep track of the receiver buffer level, thus avoiding overflow. Therefore, in all the presented results data loss is only caused by router drops.

VI. Results

In this section, the quality performance of the various techniques is compared by means of two quality measures. First, the peak signal-to-noise ratio (PSNR) between the received video sequence and the original uncompressed one is shown, together with the PSNR standard deviation, which can be interpreted as another index of video quality [64]. Moreover, to partially overcome the limitations of PSNR in evaluating video quality as perceived by human users, we also use the perceptual video quality evaluation (PVQM) algorithm proposed in [65], which provides a value that has been shown to have a good correlation with subjective quality evaluation scores. The PVQM algorithm provides an estimate of the results that would be obtained using the double stimulus continuous quality scale (DSCQS) quality evaluation method [66], in which subjects are requested to evaluate both the reference and the test video sequences using a 0–100 scale, without knowing which sequence is the reference. The analysis is based on the difference in rating for each pair, i.e., the differential mean opinion score (DMOS) value. Therefore, the lower is the DMOS value, the highest is the quality of the tested video sequence. The DSCQS method is quite sensitive to small differences in quality [67]. Since single-rating methods such as DSCQS are not suited to the evaluation of long video sequences due to the recency effect, that is, a bias in the rating towards the final 10–20 s due to limitations of human working memory [68], we estimate the PVQM on each repetition of the video sequence (which is 9.56 s long), then we average the results over the whole sequence.

A. Scenario 1

In this scenario, one video source transmits packets to its destination. Network conditions change during the simulation because a concurrent on/off UDP source is activated. The UDP source generates a constant bit-rate traffic from time 144.54 s to 287.94 s with a bandwidth variation at time 240.14 s.

Table II shows the performance of the VTSS implementation with respect to the ideal real-time source rate adaptive and CBR techniques in the network conditions previously described. The VTSS technique shows consistently higher PSNR values, in the order of 1 dB, than the ideal rate-adaptive transmission, for all the tested video sequences. The result is confirmed by the DMOS value: the average gain is about three, and it reaches 6.3 for the *tempete* sequence. As to be expected, the gain is even more pronounced with respect to the constant-bit-rate transmission technique (on average, a gain of 7.6 dB and of about 30 for the DMOS value).

Besides providing higher PSNR absolute values the VTSS technique also delivers considerably lower PSNR fluctuations. On average, the PSNR standard deviation is 1.6 dB instead of 4.9 dB. The VTSS technique, in fact, is able to keep the video quality very close to the encoding quality even considering small variations due to occasional losses, over the whole duration of the simulation, whereas the quality delivered by the ideal real-time source rate-adaptive technique, instead, varies greatly, depending on the network conditions which dictate the source coding rate. In other words, the VTSS video quality is both higher and more stable, i.e., more pleasant [64], than the one provided by the ideal real-time source rate-adaptive technique. The result is even more interesting considering that the reference rate-adaptive technique, not only enjoys, as mentioned before, a slightly higher encoding rate, but, being ideal, experiences no packet losses whereas the VTSS implementation, being realistic, does. The VTSS technique, in fact, experiences some packet losses, due to the imperfections of the TCP rate control mechanisms. The CBR transmission case, instead, is completely different, as expected. The quality of the CBR transmission technique, in fact, varies strongly due to the very high packet losses taking place when bandwidth is insufficient, as shown by the right-most column of Table II.

The next set of results illustrate four important aspects of the performance of the VTSS transmission technique, namely, throughput, PSNR, packet loss rate (PLR) and playout buffer fullness, as a function of time. Figs. 5–8 refer to simulation runs

TABLE II
PERFORMANCE OF THE VTSS IMPLEMENTATION WITH RESPECT TO THE IDEAL RATE-ADAPTIVE AND CBR TECHNIQUES (SCENARIO 1)

Sequence	Transmission scheme	Avg. PSNR (dB)	PSNR std. dev. (dB)	PVQM (DMOS)	Packet loss rate (%)
tempete	VTSS	**34.79**	**0.75**	**5.85**	0.02
	Ideal rate-adaptive	33.57	5.25	12.16	0.00
	CBR	28.57	6.59	34.58	13.43
mobile	VTSS	**33.45**	**1.79**	**15.27**	0.13
	Ideal rate-adaptive	32.47	5.62	15.35	0.00
	CBR	25.61	8.54	45.56	15.22
foreman	VTSS	**36.97**	**1.47**	**4.21**	0.10
	Ideal rate-adaptive	35.80	4.41	9.19	0.00
	CBR	25.76	11.55	44.01	11.61
silent	VTSS	**36.21**	**1.42**	**6.10**	0.06
	Ideal rate-adaptive	35.25	4.67	10.77	0.00
	CBR	32.47	4.74	27.48	15.11
mother & daughter	VTSS	**37.98**	**2.60**	**6.62**	0.77
	Ideal rate-adaptive	37.08	4.37	7.12	0.00
	CBR	28.92	9.80	38.51	14.00
average values	VTSS	**35.88**	**1.61**	**7.61**	0.22
	Ideal rate-adaptive	34.83	4.86	10.92	0.00
	CBR	28.27	8.23	38.03	13.87

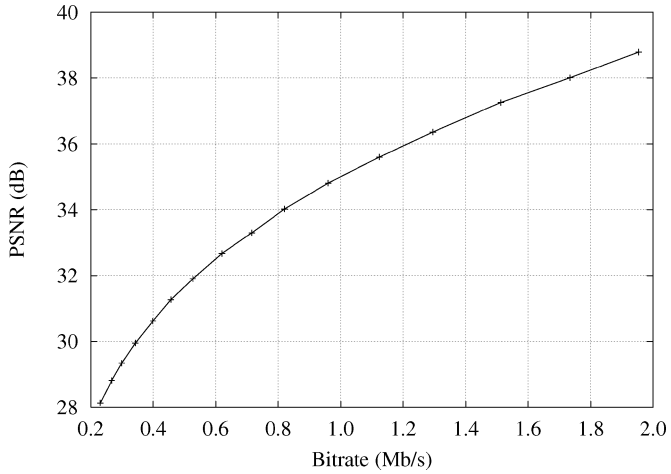

Fig. 4. Rate distortion function of the *tempete* sequence. H.264 encoding at CIF resolution, 25 fps. Each point is obtained with a fixed quantization stepsize.

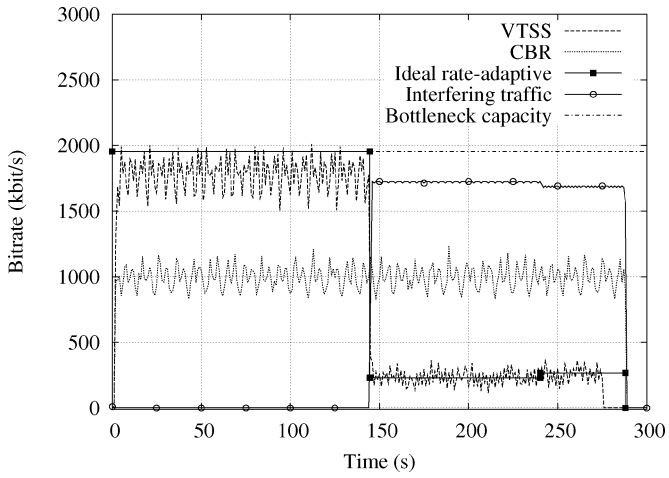

Fig. 5. Throughput of the various techniques as a function of time. The bottleneck capacity is fixed, represented by the dash and dot line. The interfering traffic is turned on at 144.54 s, approximately taking 85% of the bottleneck bandwidth. *Tempete* sequence, Scenario 1.

using the *tempete* sequence, whose rate distortion function is shown in Fig. 4.

Fig. 5 shows a comparison of the throughput of the three transmission schemes under analysis. The horizontal dash-and-dot line represents the bottleneck bandwidth, whereas the solid line represents the interfering traffic, which is absent in the first half of the simulation (up to time 144.54 s), while it occupies a large share of the available bandwidth during the second half. The VTSS throughput at the sender closely follows the available channel bandwidth as a regular TCP flow would do. The ideal real-time source rate-adaptive technique, being optimal, does even better and uses the available bandwidth perfectly. The CBR transmission technique outputs data at the same bitrate regardless of the available channel bandwidth.

Fig. 6 shows the packet loss rate for the VTSS transmission technique for the *tempete* sequence. A small amount of packets is lost in presence of interfering traffic (around $\mathrm{t} = 200$ s and 240 s), with a imperceptible impact on the overall video quality, i.e., 0.02 dB, as it can be surmised by comparing the first row of Table II to the first row of Table I.

The packet loss effect on VTSS quality is visible in Fig. 7, near to the end of the simulation, as a small and limited in time PSNR decrease. However, apart from occasional losses, the VTSS technique is able to keep the video quality equal to the encoding quality over the whole duration of the simulation. The ideal real-time source rate-adaptive technique, instead, presents a completely different PSNR behavior. Although no losses are possible since the transmission technique is ideal and the minimum channel capacity, due to design choices, is always higher than the minimum rate of the source encoder, video quality varies greatly, as shown by the PSNR value, depending on the interfering traffic bandwidth, which dictates the source coding rate. The performance shown in Fig. 7 represents the upper bound that can be achieved by any source rate-adaptive technique. The situation would be even worse if a non-ideal technique, which experiences packet losses and imperfect as well as delayed knowledge of channel conditions, was used as reference, or if the channel capacity dropped below the minimum rate of the source encoder, thus causing heavy packet losses.

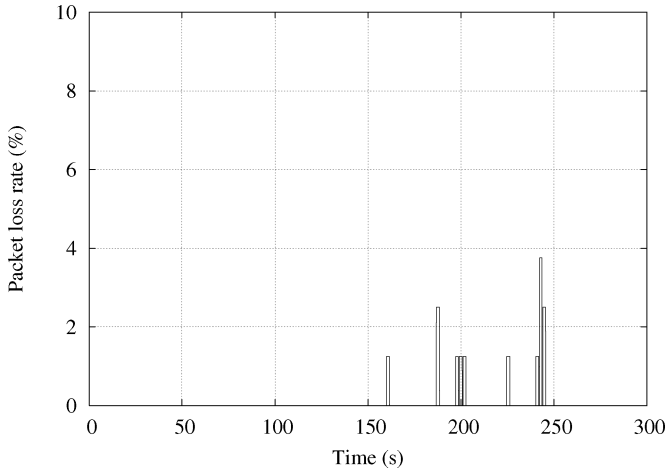

Fig. 6. Packet loss rate (moving average) as a function of time for the VTSS technique, for sequence *tempete*, Scenario 1.

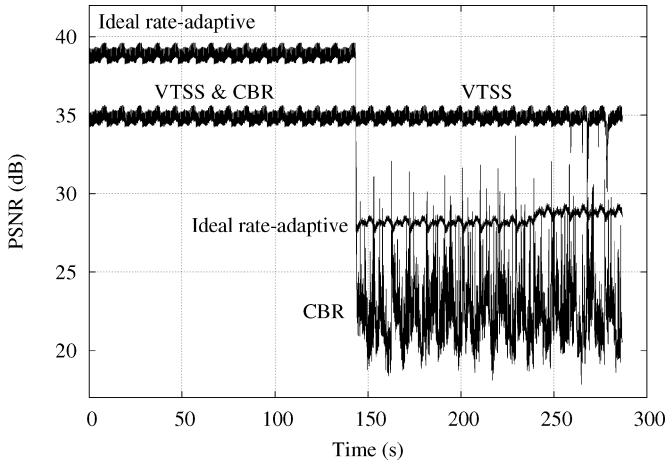

Fig. 7. PSNR as a function of time for the VTSS technique, compared with the ideal rate-adaptive and the CBR techniques. *Tempete* sequence, Scenario 1.

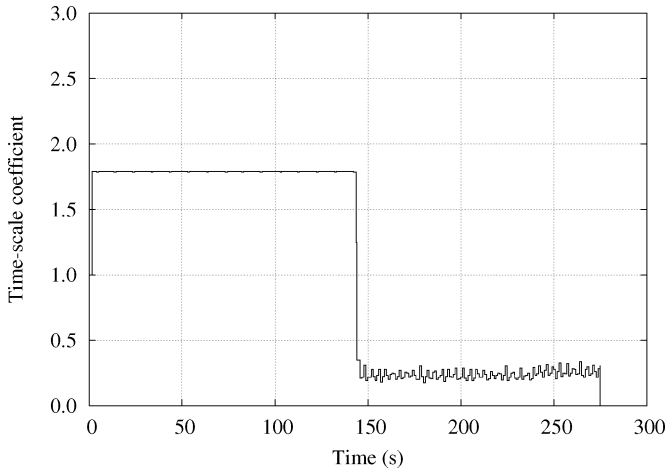

Fig. 8. Time-scale coefficient as a function of time. *Tempete* sequence, Scenario 1.

Fig. 8 shows the time-scale coefficient [see (4)] of the VTSS technique as a function of time. When the available channel bandwidth is greater than the bandwidth required by the encoded video, the time-scale coefficient is greater than one, and the video flows at almost twice the real-time; in the second half of the transmission, instead, video flows at approximately 1/4 real-time. The time-scale coefficient continuously adapts, as expected, to the available bandwidth. A slight increase of its mean is, for instance, visible when the amount of concurrent traffic decreases (after 240 s).

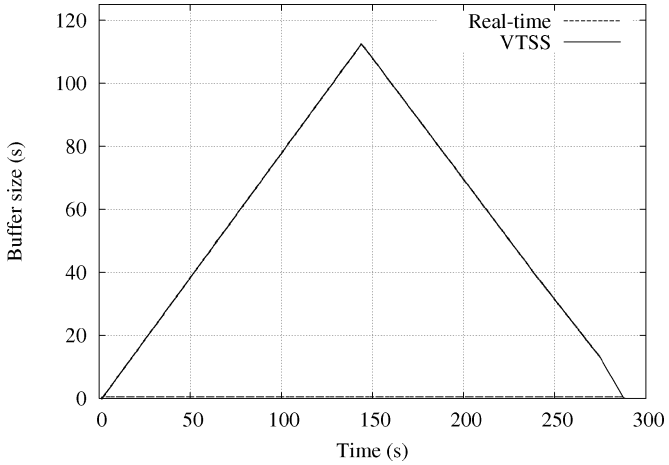

Fig. 9. Buffer size at the receiver as a function of time. *Tempete* sequence, Scenario 1. For real-time techniques, the playout buffer size has been set to 480 ms, which corresponds to the interval between two I-type frames.

Finally, receiver buffer fullness, measured in seconds, is shown in Fig. 9. Buffer fullness is a direct consequence of the behavior of the time-scale coefficient: when the coefficient is greater than one, the buffer size increases, because the amount of data that are being transmitted is greater than the amount consumed by the player. The opposite consideration holds when the coefficient is less than one. The slope of the function plotted in Fig. 9 is proportional to the value of the time-scale coefficient as well as to the data consumption rate of the player. The buffer decrease due to playback is visible after 275 s, when the VTSS transmission ends and the buffer fullness is only influenced by the playback rate. Buffer fullness after the initial buffer filling time is constant in case of real-time transmission, i.e., time-scale coefficient set to one.

B. Scenario 2

This scenario is similar to Scenario 1, but in this case an actual network trace has been used to simulate real interfering traffic. We use a publicly available trace from [69], between time instants 2900 and 3187 s. The bottleneck capacity has been limited so that it never exceeds the maximum bandwidth of the ideal real-time source rate adaptive technique, therefore it does not waste bandwidth. Comparisons have been performed with a VTSS transmission whose average bandwidth is the same as the average available channel bandwidth. Simulation results include the *tempete* and *mobile* sequences only, because simulating the other sequences with the same interfering traffic trace would cause long pauses of the ideal real-time source rate adaptive transmission in the second part of the simulation since the interfering traffic would occupy all the available channel bandwidth. In this scenario we assume that if the available channel

TABLE III
PERFORMANCE OF THE VTSS IMPLEMENTATION WITH RESPECT TO THE IDEAL RATE-ADAPTIVE AND CBR TECHNIQUES WHEN AN ACTUAL NETWORK TRACE IS USED AS INTERFERING TRAFFIC (SCENARIO 2)

Sequence	Transmission scheme	Avg. PSNR (dB)	PSNR std. dev. (dB)	PVQM (DMOS)	Packet loss rate (%)	Freeze events	Avg. freeze length (s)
tempete	VTSS	35.00	2.28	11.41	0.57	0	0
	Ideal rate-adaptive	34.28	6.13	27.68	0.00	24	1.05
mobile	VTSS	34.90	2.84	13.72	2.65	0	0
	Ideal rate-adaptive	34.84	4.32	16.63	0.00	9	0.84

bandwidth is enough to transmit at least the lowest quality version of the video sequence, the ideal real-time source rate-adaptive transmission does not experience losses. For some time intervals, however, it may happen that the interfering traffic does not leave the minimum necessary bandwidth to carry out the transmission. In such a case, we assume that the ideal real-time source rate-adaptive transmission is subject to a freeze. Transmission immediately resumes when at least the minimum bandwidth is available again.

Table III shows the performance results, in terms of PSNR and PVQM, for the VTSS implementation and the ideal real-time source rate adaptive technique. The VTSS technique, even in this scenario, shows consistently higher PSNR values, up to 0.7 dB, than the ideal real-time source rate-adaptive transmission, as well as considerably lower PSNR fluctuations. On average, the PSNR standard deviation is 2.56 dB instead of 5.22 dB. The perceptual quality measure — up to 16.2 DMOS difference — confirms the results.

Moreover, to better understand the video quality impairment experienced by the users, Table III also shows the number of freeze events and total freeze duration for the ideal real-time source rate-adaptive transmission. For the *tempete* sequence, for instance, 24 freeze events are experienced during the transmission, and their average duration exceeds one second. Fig. 10 shows the throughput of the two transmission schemes. Such freeze events are experienced when the channel bandwidth does not allow to transmit even the lowest quality version of the video, due to the presence of a large amount of interfering traffic. In the same conditions, the VTSS technique suspends the transmission, as shown by the zero value of the instantaneous time-scale coefficient in Fig. 11, but playback is not disrupted because it is fed by the receiver buffer. The VTSS technique resumes transmission as soon as channel bandwidth rises again.

Table IV shows the jitter experienced by packets with the VTSS scheme. The average value is small, about 3–4 ms. Occasionally, in presence of burst of interfering traffic, jitter can rise up to about 400 ms. However, assuming that the same happens for the ideal real-time source rate-adaptive technique, a half-second playout buffer would be sufficient to compensate such delay variations, whereas in the case of the VTSS technique the receiver buffer already compensates such jitter values.

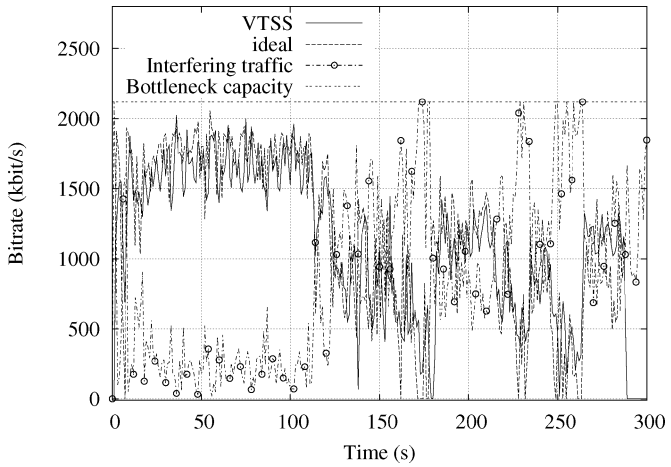

Fig. 10. Throughput of the various techniques as a function of time when the actual network trace is used as interfering traffic. The bottleneck capacity is fixed, represented by the horizontal line. *Tempete* sequence, Scenario 2.

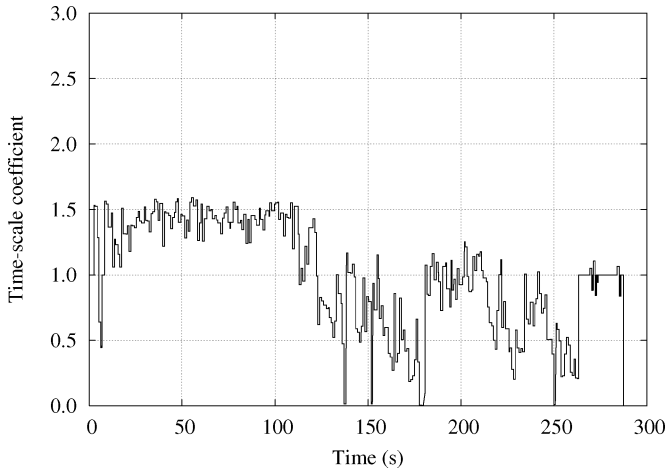

Fig. 11. Time-scale coefficient as a function of time, when the actual network trace is used as interfering traffic. *Tempete* sequence, Scenario 2.

TABLE IV
JITTER EXPERIENCED BY THE VTSS TRANSMISSION (SCENARIO 2)

Sequence	Average (ms)	Std. dev. (ms)	Maximum (ms)
tempete	3.4	5.7	442.2
mobile	3.1	4.6	370.7

C. Receiver Buffer Size

Video streaming clients are increasingly used in set-top-boxes and mobile devices, not to mention personal computers. While personal computers are typically characterized by large pre-installed memory (one or more GB of RAM) and storage capacity (32 GB or more), set-top-boxes and mobile devices usually come with less pre-installed memory, currently up to two hundred megabytes for set-top-boxes [70], [71] and tens of megabytes for low-end devices such as mobile phones with streaming capabilities [72]. However, such devices are highly heterogeneous, as some of them may also have additional storage capacity in the form of an internal hard disk or they can

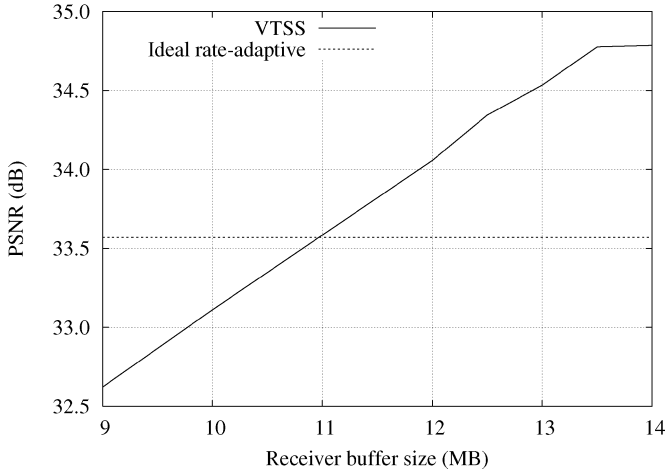

Fig. 12. PSNR as a function of receiver buffer size for the VTSS technique, compared with the ideal rate-adaptive technique. *Tempete* sequence, Scenario 1.

host large flash memories. Therefore, assuming large receiver buffers does not seem unrealistic for a large number of devices. With the VTSS technique, such devices can take full advantage of the transmission intervals during which the channel bandwidth is much higher than the average video bandwidth, thus accumulating data as fast as possible into the receiver buffer.

However, in some cases a large receiver buffer is not available, therefore we also study the performance of the VTSS technique as a function of the receiver buffer size. As pointed out while commenting on (24), the sender can estimate the amount of data in the receiver buffer (the actual value might be lower due to packet losses). To manage a limited client buffer size scenario, we modified the VTSS transmission algorithm so that it estimates, at each time instant, the receiver buffer level by means of (24) and when the level is close to the maximum the algorithm sets the instantaneous time-scale coefficient equal to one (i.e., the transmission is carried out in real-time), thus keeping the buffer level approximately constant.

Fig. 12 shows the PSNR performance, obtained in Scenario 1, as a function of various receiver buffer sizes. Clearly, the VTSS performance does not increase if the buffer is larger than a certain threshold which represents the maximum requirement for a given video sequence and network scenario. Fig. 12 also shows that, as expected, the ideal real-time source rate-adaptive technique becomes increasingly more attractive as the buffer size decreases. In fact, most of the advantages of the VTSS technique are due to the possibility to store content at the receiver as much as possible when channel is good, to ensure smooth and good-quality reproduction at the receiver when the channel bandwidth drops.

To roughly quantify the buffer requirements in a real case, assume that the channel bandwidth is constant, equal to C, and that the video sequence bitrate is $B_S < C$ and its duration is L. The VTSS will transmit using an instantaneous time scale coefficient $\rho = C/B_S$. From (24), the buffer size level can be approximately computed as $B_R = Ct - B_S t$. In this simplified constant channel bandwidth scenario, ρ is constant, therefore $B_R = B_S\rho t - B_S t = B_S t(\rho - 1)$. The maximum receiver buffer size level is experienced at the end of the faster-than-realtime transmission, at time $t^* = LB_S/C$. By substituting t^* in the previous expression, $B_{R,\max} = B_S(LB_S/C)(\rho - 1) = B_S L(1 - 1/\rho)$. Therefore, in this simple scenario, the required buffer size at the receiver can be easily computed by multiplying the whole video sequence size in bytes by factor $f_B = (1 - 1/\rho)$. Such factor varies from zero to one depending on the ratio between available channel bandwidth and video sequence bitrate. In a realistic case, e.g., standard-resolution TV over DSL, $B_S = 4\ \mathrm{Mb/s}$, $C = 8\ \mathrm{Mb/s}$, $L = 1\ \mathrm{hour}$, f_B is equal to 0.5 and therefore the receiver buffer size should be able to accommodate half of the video sequence, i.e., 858 MB, which is not an issue for any currently available personal computer. The same amount of data can be easily accommodated in a 1 GB cheap flash memory unit, as found in an increasing number of mobile devices, or in the internal hard-disk of a set-top-box.

However, if the buffer size is smaller, for instance, 192 MB as in a currently available disk-less set-top-box [70], with reference to the previous example the VTSS algorithm will behave as follows. Let B_M be the buffer size. At the beginning of the transmission the buffer will increase its level, up to time $t_F = B_M/(C - B_S)$, i.e., 384 s. Then, the transmission will proceed in real-time not to overflow the buffer, although the channel bandwidth would allow to transmit faster. However, even with such a reduced buffer size compared to $B_{R,\max}$, a smooth playback is guaranteed for at least $T = B_M/B_S$, i.e., more than 6 min in the example, even if the channel bandwidth drops to zero. Mobile devices may have even less memory. However, in this case the video sequence bit rate is also lower when compared to the example, therefore the receiver buffer requirements are correspondingly reduced.

VII. Conclusion

A VTSS technique for multimedia streaming over IP networks has been studied, via both a theoretical analysis and network simulations. According to the VTSS technique, rather than constraining the transmitter to operate in real-time, the time scale of the packet scheduler can change from zero when the network is congested to as faster than real-time as the channel bandwidth allows when the network is lightly loaded. Under the hypothesis of no losses, we analytically showed that the minimum total distortion achieved by the real-time source rate adaptive strategy operating in ideal conditions is lowered or, at worst, left unchanged, by using the VTSS transmission strategy. Assuming typical video rate-distortion curves, the distortion can always be lowered. To validate the theoretical analysis and to test the VTSS approach in more practical terms, network simulation results of a TCP-friendly test implementation of the VTSS approach were presented using H.264 test video sequences, realistic network conditions and objective measures of video quality. Results show that the test VTSS implementation delivers consistently higher as well as more constant quality when compared to an ideal real-time source rate-adaptive transmission.

Acknowledgment

The authors would like to thank M. B. Frusca for its valuable contribution in the implementation of a first version of the

VTSS algorithm in *ns*, and for running several early simulations. Moreover, the authors are grateful to the anonymous reviewers for their insightful comments.

REFERENCES

[1] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 289–298, Oct. 1993.

[2] K. K. Vadde and V. R. Syrotiuk, "Factor interaction on service delivery in mobile ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 7, pp. 1335–1346, Sep. 2004.

[3] *802.16 IEEE Standard for Local and Metropolitan area networks — Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Std. 802.16, Oct. 2004.

[4] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Process. Mag.*, vol. 17, no. 4, pp. 61–82, Jul. 2000.

[5] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[6] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[7] H. Kanakia, P. P. Mishra, and A. R. Reibman, "An adaptive congestion control scheme for real time packet video transport," *IEEE/ACM Trans. Netw.*, vol. 3, no. 6, pp. 671–682, Dec. 1995.

[8] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video," *IEEE Trans. Multimedia*, vol. 1, no. 3, pp. 302–312, Sep. 1999.

[9] J. M. McManus and K. W. Ross, "Video-on-demand over ATM: Constant-rate transmission and transport," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 1087–1098, Aug. 1996.

[10] Z.-L. Zhang, J. F. Kurose, D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 1148–1166, Aug. 1997.

[11] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 397–410, Aug. 1998.

[12] G. Cao, W. Feng, and W. Singhal, "Online variable-bit-rate video traffic smoothing," *Comput. Commun.*, vol. 26, no. 7, pp. 639–651, May 2003.

[13] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate based congestion control mechanism for real-time streams in the Internet," in *Proc. IEEE INFOCOM*, Mar. 1999.

[14] A. Allen, "Optimal delivery of multi-media content over networks," in *Proc. ACM Multimedia*, 2001, pp. 79–88.

[15] A. Odlyzko, Telecom Dogmas and Spectrum Allocations. Wireless Unleashed 2004 [Online]. Available: http://www.wirelessunleashed.com

[16] E. Masala, D. Quaglia, and J. C. De Martin, "Variable time-scale streaming for multimedia transmission over IP networks," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Antalya, Turkey, Sep. 2005.

[17] P. Pancha and M. El Zarki, "Bandwidth requirements of variable bit rate MPEG sources in ATM networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1993, pp. 902–909.

[18] J. Lauderdale and D. H. K. Tsang, "Bandwidth scheduling of prerecorded VBR video sources for ATM networks," in *Proc. IEEE ATM Workshop*, Washington, DC, Oct. 1995.

[19] B. J. Vickers, M. Lee, and T. Suda, "Feedback control mechanism for real-time multipoint video services," *IEEE J. Select. Areas Commun.*, vol. 15, no. 3, pp. 512–530, Apr. 1997.

[20] V. Jacobson, "Congestion avoidance and control," in *Proc. Symp. Communications Architectures and Protocols*, Aug. 1988, pp. 314–329.

[21] X. Wang and H. Schulzrinne, "Comparison of adaptive Internet multimedia applications," *IEICE Trans. Commun.*, vol. E82-B, no. 6, pp. 806–818, Jun. 1999.

[22] G. De Marco, M. Longo, and F. Postiglione, "Run-time adjusted congestion control for multimedia: Experimental results," in *Proc. IEEE Int. Conf. Advanced Information Networking and Applications*, Mar. 2004, vol. 1, pp. 531–536.

[23] S. Jin, L. Guo, I. Matta, and A. Bestavros, "A spectrum of TCP-friendly window-based congestion control algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 341–355, Jun. 2003.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its emperical validation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 303–314, Oct. 1998.

[25] H. Schulzrinne, "A comprehensive multimedia control architecture for the Internet," in *Proc. IEEE NOSSDAV*, St. Louis, MO, May 1997, pp. 65–76.

[26] Y. Liu, K. N. Srijith, L. Jacob, and A. L. Ananda, "TCP-CM: A transport protocol for TCP-friendly transmission of continuous media," in *Proc. IEEE Int. Conf. Performance, Computing, and Communications*, Apr. 2002, pp. 83–91.

[27] S. Jacobs and A. Eleftheriadis, "Real-time dynamic rate shaping and control for internet video applications," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Jun. 1997, pp. 558–563.

[28] B. Mukherjee and T. Brecht, "Timelined TCP for the TCP-friendly delivery of streaming media over the Internet," in *Proc. ICNP 2000*, Nov. 2000.

[29] J. C. Bolot and T. Turletti, "A rate-control mechanism for packet video in the Internet," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, Jun. 1994, vol. 3, pp. 1216–1223.

[30] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," *Comput. Commun.*, vol. 19, no. 1, pp. 49–58, Jan. 1996.

[31] Y. R. Yang and S. S. Lam, "General AIMD congestion control," in *Proc. IEEE Int. Conf. Network Protocols*, Nov. 2000, pp. 187–198.

[32] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 2, pp. 631–640.

[33] A. Barberis, C. Casetti, J. C. De Martin, and M. Meo, "A simulation study of adaptive voice communications on IP networks," *Comput. Commun.*, vol. 24, no. 9, pp. 757–767, May 2001.

[34] G. Davini, D. Quaglia, J. C. De Martin, and C. Casetti, "Perceptually-evaluated loss-delay controlled adaptive transmission of MPEG video over IP," in *Proc. IEEE Int. Conf. Communications*, Anchorage, AK, May 2003, vol. 1, pp. 577–581.

[35] J. Mahdavi and S. Floyd, TCP-Friendly Unicast Rate Based Flow Control. Note Sent to the End2end-Interest Mailing List 1997 [Online]. Available: http://www.psc.edu/networking/papers/tcp_friendly.html

[36] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in *Proc. IEEE NOSSDAV*, Jun. 1999.

[37] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 43–56, Oct. 2000.

[38] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 339–355, Sep. 2001.

[39] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," in *RFC3448*, Jan. 2003.

[40] D. Sisalem and H. Schulzrinne, "The loss-delay adjustment algorithm: A TCP-friendly adaptation scheme," in *Proc. IEEE NOSSDAV*, Cambridge, U.K., Jul. 1998.

[41] E. Kohler, M. Handley, and S. Floyd, "Datagram congestion control protocol (DCCP)," in *RFC4340*, Mar. 2006.

[42] N. Yin and M. G. Hluchyj, "A dynamic rate control mechanism for source coded traffic in a fast packet network," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1003–1012, Sep. 1991.

[43] U. Kazunori, O. Hiroyuki, S. Shinji, and M. Hideo, "Design and implementation of real-time digital video streaming system over IPv6 network using feedback control," in *Proc. IEEE Symp. Applications and the Internet*, 2003.

[44] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, no. 5, pp. 756–773, May 1999.

[45] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[46] A. Balk, M. Gerla, D. Maggiorini, and M. Sanadidi, "Adaptive video streaming: Pre-encoded MPEG-4 with bandwidth scaling," *Comput. Netw.*, vol. 44, no. 4, pp. 415–439, Mar. 2004.

[47] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637–644, Jul. 2003.

[48] N. Yeadon, F. Garcia, D. Hutchison, and D. Shepherd, "Filters: QoS support mechanisms for multipeer communications," *IEEE J. Select. Areas Commun.*, vol. 14, no. 7, pp. 1245–1262, Sep. 1996.

[49] *Coding of Audio-Visual Objects – Part 2: Visual*, ISO/IEC Std. 14496-2 (MPEG-4), 2001.

[50] H. M. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, Mar. 2001.
[51] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.
[52] M. van der Schaar and H. Radha, "A hybrid temporal-SNR fine-granular scalability for internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 318–331, Mar. 2001.
[53] W.-T. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. Multimedia*, vol. 1, no. 2, pp. 172–186, Jun. 1999.
[54] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.
[55] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.
[56] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for internet video streaming and multicast," *Signal Process. Image Commun.*, vol. 16, no. 8, pp. 745–762, May 2001.
[57] C.-M. Chen, C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Error-resilient video streaming over wireless networks using combined scalable coding and multiple-description coding," *Signal Process. Image Commun.*, vol. 22, no. 4, pp. 403–420, Apr. 2007.
[58] D. Kozen, Y. Minsky, and B. Smith, "Efficient algorithms for optimal video transmission," in *Proc. Data Compression Conf.*, Mar./Apr. 1998, pp. 229–238.
[59] C. W. Fung and S. C. Liew, "End-to-end frame-rate adaptive streaming of video data," in *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, Jun. 1999, vol. 2, pp. 67–71.
[60] M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, "MPEG system streams in best-effort networks," in *Proc. Packet Video Workshop*, May 1999.
[61] Z.-L. Zhang, S. Nelakuditi, R. Aggarwal, and R. P. Tsang, "Efficient selective frame discard algorithms for stored video delivery across resource constrained networks," in *Proc. IEEE INFOCOM*, Mar. 1999, vol. 2, pp. 472–479.
[62] UCB/LBNL/VINT Network Simulator—ns (version 2) 1997 [Online]. Available: http://www.isi.edu/nsnam/ns
[63] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T & ISO/IEC Std. H.264 & 14496-10, May 2003.
[64] X. Lu, R. O. Morando, and M. El Zarki, "Understanding video quality and its use in feedback control," in *Proc. Packet Video Workshop*, Pittsburgh, PA, Apr. 2002.
[65] A. P. Hekstra *et al.*, "PVQM—A perceptual video quality measure," *Signal Process. Image Commun.*, vol. 17, no. 10, pp. 781–798, Nov. 2002.
[66] *Methodology for the Subjective Assesment of the Quality of Television Pictures*, ITU-R Std. BT.500-11, Jun. 2002.
[67] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. Chichester, U.K.: Wiley, 2005, ch. 3, pp. 51–54.
[68] R. Aldridge, J. Davidoff, M. Ghanbari, D. Hands, and D. Pearson, "Recency effect in the subjective assessment of digitally-coded television pictures," in *Proc. Int. Conf. on Image Processing and its Applications*, Edinburgh, U.K., Jul. 1995, pp. 336–339.
[69] Comput. Dept, Univ. Naples, Naples, Italy, "Network Tools and Traffic Traces" 2008, 6th trace [Online]. Available: http://www.grid.unina.it/Traffic/Traces/ttraces.php
[70] ADB-3800W Data Sheet: Advanced, High-Definition, IPTV Set-Top Box with Home-Networking Capabilities ADB, 2008 [Online]. Available: http://www.adb-global.com/files/ADB_datasheet_3800W_HD_USqtr.pdf
[71] GCT K35 Data Sheet GCT, 2008 [Online]. Available: http://gctglobal.com/Products/Set_Top_Box/K35/k35.html
[72] Nokia 6120 Classic Technical Specifications Nokia, 2008 [Online]. Available: http://shop.nokia.co.uk/nokia-uk/product.aspx?sku=3762320&section_id=530&culture=en-GB

Enrico Masala (S'01–M'04) received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 2004.

In 2003, he was a Visiting Researcher at the Signal Compression Laboratory, University of California, Santa Barbara, where he worked on joint source channel coding algorithms for video transmission. He is currently a Postdoctorate Researcher at the Politecnico di Torino. His main research interests include multimedia processing (especially video), coding, and transmission over wireline and wireless packet networks.

Dr. Masala is a member of the Multimedia Communications Technical Committee within the IEEE Communications Society.

Davide Quaglia (S'00–M'04) received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 2003.

Since January 2005, he has been an Assistant Professor at the Computer Science Department, University of Verona, Verona, Italy, where he teaches computer networks and multimedia architectures. His current research interests include networked embedded systems, wireless sensor networks, multimedia communications, robust delivery of multimedia signals over packet networks, video coding, and applications for impaired-people.

Dr. Quaglia was the Chairman of the IEEE Student Branch of Politecnico di Torino and is currently a member of the IEEE Communications Society.

Juan Carlos De Martin (M'92) received the Ph.D. degree in computer engineering, from the Politecnico di Torino, Turin, Italy, in 1996.

He spent two years (1993–1995) as Visiting Scholar at the Signal Compression Laboratory, University of California Santa Barbara, and two years (1996–1998) as Texas Instruments, Dallas, TX, as a Member of Technical Staff and as an Adjunct Professor at the University of Texas (1999). Between 1998 and 2005, he was a Principal Researcher at the National Research Council (CNR) of Italy, Torino, where he led the Multimedia Communications Research Group. He is currently an Associate Professor at the Information Engineering School, Politecnico di Torino, where he coordinates the Internet Media Research Group. His research activities are focused on multimedia processing and transmission, with a special emphasis on high-performance streaming techniques. He is also active in exploring the interaction between digital technologies and society. In this regard, in November 2006 he founded and currently directs the NEXA Center for Internet and Society of the Politecnico di Torino; he is also the Coordinator of COMMUNIA, the European thematic network on the digital public domain funded by the European Commission (2007–2010). He is the author or co-author of over 80 international scientific publications and of several patents; he is also an expert Evaluator of research programs for the Italian Ministry of University and Research, for the Ministry of Industrial Activities, and for the Swiss Science Foundation.

Dr. De Martin serves as a member of the IEEE Multimedia Communications and of the IEEE Signal Processing Education Technical Committees.