VARIABLE TIME–SCALE STREAMING FOR MULTIMEDIA TRANSMISSION OVER IP NETWORKS

Enrico Masala, Davide Quaglia, Juan Carlos De Martin[*]

Dipartimento di Automatica e Informatica / *IEIIT–CNR — Politecnico di Torino
c.so Duca degli Abruzzi, 24 — 10129 Torino, Italy
phone: (+39) 011 5647036, fax: (+39) 011 5647099
email: [masala|davide.quaglia|demartin]@polito.it, web: http://media.polito.it

ABSTRACT

This paper presents an analysis of a rate adaptive multimedia streaming technique according to which rate changes are obtained by varying the inter-packet transmission interval, rather than altering the source coding rate. Instead of constraining the transmitter to operate in real–time, the time scale of the proposed 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. Simulation results comparing a TCP–friendly test implementation of the variable time-scale streaming (VTSS) approach with an *ideal* source rate–adaptive technique — whose performance represents the upper bound of any transmission system based on source rate adaptation— show that the VTSS approach delivers higher perceptual quality (up to 1.2 dB PSNR in the considered scenario) and reduced quality fluctuations, for a wide range of standard video sequences. The gains are even more pronounced when the proposed technique is compared to constant bit–rate transmission.

1. INTRODUCTION

Multimedia applications over IP networks are currently at the center of an extraordinary deal of attention. However, for this appealing class of applications to succeed, several major problems need to be solved. Among them, perhaps the most challenging one is how to best deal with the strongly time–varying nature of IP networks, both wired and—even more so—wireless.

Many proposals have been made to address the problem of the time–varying bandwidth, loss rates and delays. They range from physical–layer solutions (e.g., constant–BER modulation schemes) to application–layer (e.g., joint source–channel coding) approaches [1]. Several studies also suggested that an end-to-end flow control should be adopted by multimedia flows to prevent congestions and unfair use of network resources [2]. Thus, a large number of rate–adaptive approaches have been proposed, in which the transmission rate is adapted to the estimated value of the instantaneous channel capacity.

Traditional rate–adaptive techniques are typically based on source rate variations, the assumption being that the distortion introduced by lowering the source coding rate is smaller than the expectedly larger distortion due to packet losses [3]. This work analyzes a new approach to adapt the rate of pre–compressed multimedia content to the instantaneous channel capacity, based on the decoupling of the transmission rate from the playback rate. Streaming systems usually transmit video or audio frames in real–time, that is with the same rate at which they will be decoded and presented to the user, while according to the approach analyzed in this paper, hereafter referred to as variable time–scale streaming (VTSS), the packet scheduler is free to change its instantaneous transmission rate from zero (i.e., the transmission pauses) when the channel capacity is low, to as faster than real–time as the channel bandwidth allows. The original playback rate, however, is not changed, and the source coding rate remains constant (although it is certainly conceivable to combine source coding rate adaptation and variable time scale transmission).

Several works (see [4] 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 on a constant–bit–rate channel. Other proposals vary the inter–packet gap in order to adjust transmission speed, such as in the Rate Adaptation Protocol (RAP) [5]. That work, however, mainly focuses on the end–to–end congestion control rather then on analyzing the effect of the technique on multimedia quality. In [6], a variable transmission rate technique is studied; the work focuses on optimizing bandwidth usage by a streaming server, while controlling the buffer fullness of the clients; yet the case in which the transmission rate is zero is not considered, nor 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 to quickly fill the playout buffer at the beginning of the transmission; a rigorous analysis of their behavior is, however, impossible due to the confidentiality surrounding the algorithms internally used by such applications.

In this paper we analyze the variable time–scale streaming approach, firstly formulating the problem analytically, then providing results about the perceived quality that can be obtained by the VTSS and other reference techniques. More specifically, a comparison has been made with an ideal implementation of the rate-adaptive approach, whose performance represents the upper bound of any transmission system based on source rate adaptation, e.g. Fine Grain Scalability (FGS) and similar techniques. 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.

The paper is organized as follows. Section 2 introduces the video streaming scenario, then Section 3 presents the VTSS approach. Simulation setup is described in Section 4, followed by results comparing the VTSS approach with other reference techniques (Section 5). Finally, conclusions are drawn in Section 6.

2. VIDEO STREAMING SCENARIO

Several steps need to be performed to transmit multimedia data over packet networks. The first is source encoding, that reduces the redundancy of the source. In case of video coding, the compression operations are performed on segments of input data. The compressed data can be divided into *presentation units* (PU), with the property that all the data belonging to a single PU are played back at the same time during the decoding process. Packet networks enforce a maximum size constraint on each *transmission unit* (TU), namely a packet. Therefore, each PU may be encapsulated in one or more TU's, that share the presentation time. Each TU is then transmitted and buffered at the receiver, waiting to be reassembled into PU's, decoded and presented to the user. All the TU's that con-

This work was supported in part by MIUR, Project FIRB PRIMO, http://primo.ismb.it

stitute a PU must be received before the PU's presentation time in order to decode the data without errors.

Let $t_{i,j}$ be the time instant the j-th TU of the i-th PU is sent, and let τ_i be the presentation time of the PU with which the TU is associated. Moreover, each TU is subject to a transmission delay $\delta_{i,j}$. To allow error-free decoding at the receiver, the following condition must hold for each TU:

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

Let $s_{i,j}$ be the size in bytes of the j-th TU in the i-th PU. Assuming to know the instantaneous available channel capacity $C(t)$ along the path between the source and the destination, each TU can be transmitted only if

$$s_{i,j} \leq \int_{t_{i,j}}^{t_{i,j+1}} C(t')\mathrm{d}t'. \tag{2}$$

A scheduling strategy is needed to determine the $t_{i,j}$ values. One of the simplest approaches is to equispace the TU's inside each PU. A different approach consists in sending all the TU's of a certain PU consecutively at the beginning of the PU time. In the first case, the rate offered to the network tends to be smooth, while in the latter case the probability to receive the TU's in time is greater because TU's can tolerate a higher delay than in the previous case.

The encoding and packetization strategy determines the amount and the size $s_{i,j}$ of the TU's. In case of on demand streaming, the video is compressed for later transmission. Hence the $s_{i,j}$ values are fixed in advance. But the channel capacity is usually not known, therefore this approach is highly suboptimal. A number of techniques have been proposed to allow dynamic adjustment of the rate of precompressed video at transmission time, such as scalable coding techniques. Since these techniques suffer of an intrinsic performance reduction compared with non-scalable techniques, rate adaptation methods for single-layer encoded video have also been proposed. An example is to compress or transcode the video data shortly before transmitting it, maximizing the quality of the source coding for the current channel conditions. Another technique consists in maintaining on the server multiple copies that are pre-encoded at different bitrates, and then switching between them at predefined points according to the decisions of a source rate control algorithm.

Summarizing, the *ideal* encoding technique should be able to obtain the maximum coding efficiency adjusting the source coding rate to be as close as possible to the effectively available channel bandwidth.

3. VARIABLE TIME SCALE STREAMING

Regardless of the encoding strategy, PU's are generally transmitted on the network with the same rate at which they will be decoded and presented to the user. We refer to this approach as *real–time transmission*. Assuming that the first TU is transmitted at the beginning of each PU, the following condition holds:

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

A Variable Time–Scale Streaming (VTSS) approach, instead, is based on the concept of varying the transmission rate of the TU's according to certain criteria, for instance, the instantaneous network conditions, while the original playback rate remains unchanged. 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 equality holds for Equation (2).

Consequently, PU's may be locally more closely set in time than in the real–time case or more apart from each other. Fig. 1 shows examples of a TU transmission schedule for both regular and VTSS streaming. Fig. 1(a) represents TU transmission times in the case of traditional real–time streaming. TU's can be spaced in any way in their PU time, while PU's time is constant. Fig. 1(b) and (c) refer to a VTSS transmission, for which the PU's time is not constant. Fig. 1(c) describes a VTSS transmission that ends earlier

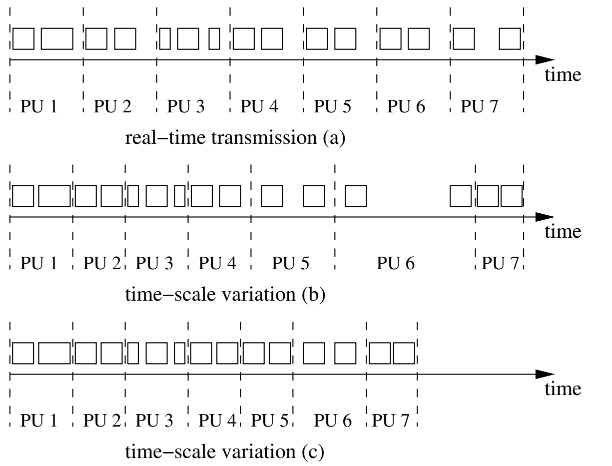

Figure 1: Examples of TU's transmission schedule for (a) standard real-time streaming, (b) VTSS with global real–time behavior and (c) VTSS with faster than real–time transmission.

than real–time, while in Fig. 1(b) it is globally in real-time. In order to discriminate between the various cases, we introduce the *time–scale coefficient* ρ_i, defined as follows:

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

When the condition $t_{i+1,1} - t_{i,1} = \tau_{i+1} - \tau_i$ holds, then the transmission is carried out in the usual *real–time* fashion, and $\rho_i = 1$. If $\rho_i > 1$, the transmitter is sending packets at a rate higher than the receiver's consumption rate. The opposite consideration holds for $\rho_i < 1$.

Fig. 2 describes the block diagram of a video transmission system using the VTSS approach. Note that in the remainder of this paper the terms TU and packet will be used interchangeably, as well as PU and frame. With the VTSS approach, a pre–compressed bitstream is transmitted as a sequence of packets on the network; the pacing of the packets is determined by the scheduler in order to achieve the time–scale coefficient ρ_i imposed by the time–scale selection algorithm. When data packets enter the receiver they are retained into the playout buffer until they are decoded and presented to the user. The channel state monitor determines the instantaneous channel state, then the status information is sent back to the time–scale selection algorithm which modifies the time–scale coefficient according to the estimate of the instantaneous channel capacity.

The playout buffer should handle the potentially large variations of the rate that characterize the technique. When the time–scale coefficient ρ_i is greater than one, the playout buffer grows and the accumulated bits are used to assure continuous playback. The time–scale coefficient is lower than one, for instance, when the channel capacity is less than the source encoding rate. Therefore, the maximum rate variation is limited by the size of the playout buffer.

Another important aspect of the VTSS design is to take into account, when choosing the current time–scale coefficient, the instantaneous conditions of the receiver–side playout buffer. If the time–scale coefficient has been less than one for a long time, in fact, the playout buffer may empty, potentially disrupting playback. In this case, the transmission strategy could be modified reducing the source bit rate as for the classic rate–adaptive approach, returning to the VTSS when the bandwidth is large enough.

Finally, note that the VTSS approach can be applied to any multimedia coding standard and may be used in conjunction with any kind of source coder (i.e., constant–bit–rate, variable–bit–rate, scalable, etc.). Source coding can be performed freely, e.g., at constant playback quality. On the contrary, in many traditional adaptive approaches source coding rate and source quality strictly depend on

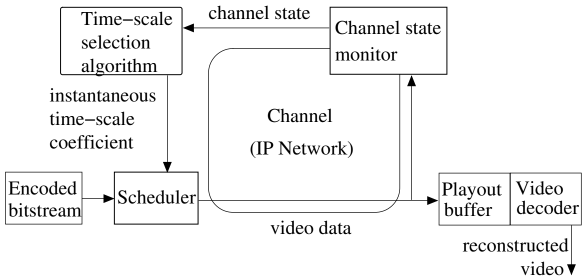

Figure 2: Block diagram of a VTSS–based video transmission system.

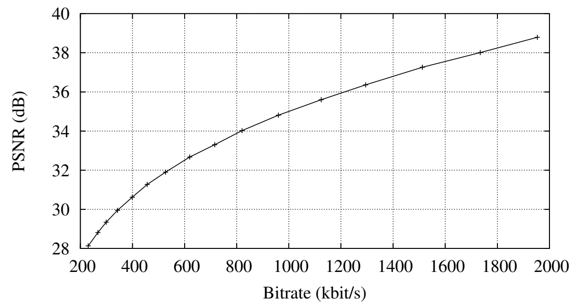

Figure 3: Rate distortion function for the *tempete* sequence.

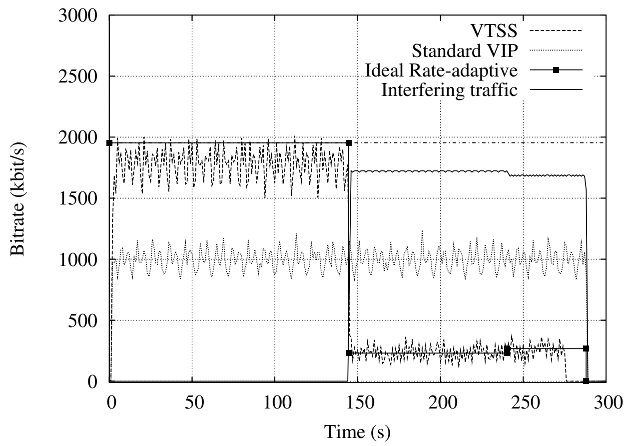

Figure 4: Throughput of the various techniques as a function of time.

the channel state. 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.

4. SIMULATION SETUP

To test the VTSS approach we designed a specific implementation in which the rate of the VTSS source has been forced to be less than or equal to the one of a generic TCP transmission in presence of the same concurrent interfering traffic. Thus our specific VTSS implementation is absolutely TCP-friendly.

This VTSS implementation was tested with the *ns* network simulator [7] and actual video sequences. We used 239 frames of several CIF (352×288) standard video sequences. Simulations were performed on the sequences concatenated with themselves 30 times, for a total of 7170 frames, to achieve statistical significance. At 25 frame per second the length of the resulting sequence is 286.8 s when transmitted in real–time. The sequence was encoded using the H.264 [8] reference encoder version JM 8.0. Each GOP consists of 12 pictures with a B-picture between the I- or P-pictures. When the video is transmitted in real–time, the GOP length is 480 ms. We instructed the Network Abstraction Layer of H.264 to put two rows of macroblocks in each packet, with a maximum transmission unit of 1500 byte. The network topology features a simple bottleneck topology, with a 5-ms source-to-destination propagation delay. The other links are oversized in bandwidth not to impact on the results.

The performance of the proposed VTSS sample implementation was compared with an ideal rate–adaptive algorithm and with the traditional non–adaptive approach which consists of sending the video sequence encoded at constant–bit–rate (VIP). In each simulation, one video source transmits packets to its destination. Network conditions change during the simulation because a concurrent on/off UDP source is activated. This source generates a constant bit-rate traffic from time 114.54 s to 287.94 s with a bandwidth variation at time 240.14 s, as shown in Fig. 4.

The VTSS transmission technique changes its rate as a TCP source in the same channel conditions. The ideal rate-adapted source, instead, varies the encoding rate to match the available capacity of the bottleneck. To maximize the performance we assume that the transmitter exactly knows this value and changes the source rate instantaneously. Finally, the non–adaptive approach keeps its transmission rate constant, regardless of network conditions.

5. RESULTS

Table 1 shows the performance of the VTSS implementation with respect to the rate-adaptive transmission and VIP in the simulation conditions previously described. The VTSS technique shows consistently higher PSNR values than the ideal rate-adaptive transmission, in the order of 1 dB, for all the tested video sequences, even if the rate–adaptive technique is ideal, thus not feasible in practice. The gain is definitely more pronounced with respect to the standard constant–bit–rate transmission technique.

Moreover, the PSNR standard deviation is considerably lower for the VTSS source. This is an intrinsic advantage of the VTSS technique. The PSNR variation in the VTSS case is introduced by the encoding process and minimally by the limited packet losses. The ideal rate-adaptive technique, instead, presents strong PSNR variations by definition. Being adaptive to the available channel rate, in fact, it naturally changes the average PSNR, thus increasing the variance.

Therefore, the higher quality stability of VTSS is due not only to the lower byte loss rate but also to the fact that VTSS approach does not change the source coding rate, as other traditional rate–adaptive techniques need to do. The quality variation of the standard CBR transmission technique is high due to the very high loss rate in case of insufficient available bandwidth, as the last column of Table 1 shows.

The next results illustrate the behavior of the VTSS transmission technique as a function of time in terms of a number of parameters, i.e. bandwidth, PSNR and packet loss rate. All the graphs refer to the simulation based on the *tempete* sequence. The rate distortion function for the *tempete* sequence is shown in Figure 3. That graph can be used to accurately determine the PSNR values corresponding to the rate values assumed by the ideal rate–adaptive flow.

Figure 4 shows a comparison of the throughput of the various transmission schemes. The horizontal dash-and-dot line at about 1950 kbit/s represents the bottleneck bandwidth. The solid line represents the interfering traffic, that is absent in the first half of the simulation, 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, with some oscillations. The ideal rate–adaptive technique, being optimal, uses the available bandwidth completely. The standard video transmission technique outputs data at the same bitrate regardless of the available channel bandwidth, therefore a high packet loss rate is experienced

Table 1: Performance of the VTSS implementation with respect to the ideal rate-adaptive and standard VIP techniques.

Sequence	Transmission scheme	Avg. PSNR (dB)	PSNR std. dev. (dB)	Packet loss rate (%)
tempete	VTSS	34.79	0.75	0.02
	Ideal Rate-adaptive	33.57	5.25	0.00
	Standard Video over IP	28.57	6.59	13.43
mobile	VTSS	33.45	1.79	0.13
	Ideal Rate-adaptive	32.47	5.62	0.00
	Standard Video over IP	25.61	8.54	15.22
foreman	VTSS	36.97	1.47	0.10
	Ideal Rate-adaptive	35.80	4.41	0.00
	Standard Video over IP	25.76	11.55	11.61
mother & daughter	VTSS	37.98	2.60	0.77
	Ideal Rate-adaptive	37.08	4.37	0.00
	Standard Video over IP	28.92	9.80	14.00

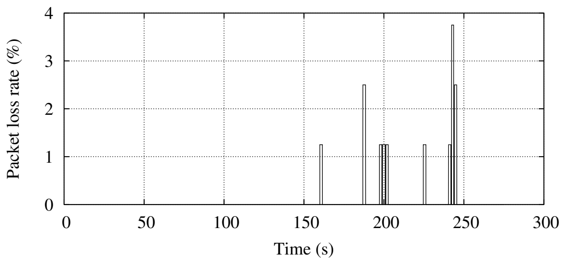

Figure 5: Packet loss rate as a function of time for the VTSS technique.

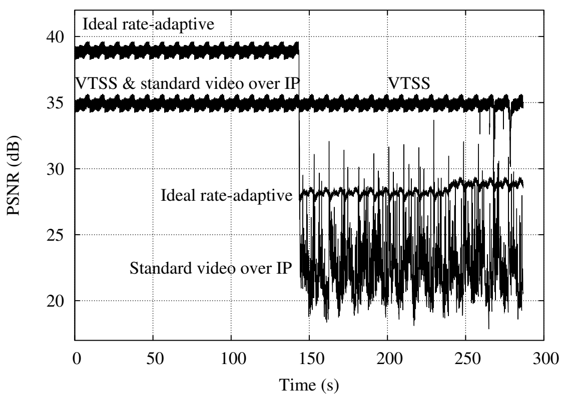

Figure 6: PSNR as a function of time for the VTSS technique, compared with the ideal rate-adaptive and the standard techniques.

by the video flow in presence of interfering traffic.

Figure 5 shows the packet loss rate for the VTSS transmission technique. While no losses are reported when the bottleneck bandwidth is enough to transmit the whole video flow, a very limited amount of packets is lost in presence of interfering traffic. The impact on the video quality, however, is very limited, as shown by Figure 6. Quantitatively, the impact is about 0.02 dB. The artifacts cause a limited PSNR decrease, visible in Figure 6 near the end of the simulation time. The ideal rate–adaptive technique presents a completely different PSNR behavior. Although no losses are possible with this ideal transmission technique, the PSNR greatly varies depending on the interfering traffic bandwidth. This PSNR variability is an intrinsic characteristic of all the rate–adaptive transmission techniques, even if ideal. A measure of the PSNR variability is given by the standard deviation values of Table 1. The PSNR standard deviation of the VTSS technique is consistently lower with respect to the other techniques, including the ideal rate–adaptive technique.

6. CONCLUSIONS

A variable time-scale approach for multimedia streaming over IP networks has been analyzed. According to this approach, 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. Network simulation results for a TCP–friendly test implementation of the Variable Time-Scale Streaming (VTSS) approach have been presented using H.264 test video sequences and objective measures of video quality. Results show that, compared to an ideal rate–adaptive transmission, VTSS delivers consistently higher and more constant quality.

7. ACKNOWLEDGMENT

The authors would like to thank Marco Boeris Frusca for its valuable contribution in the implementation of a first version of the VTSS algorithm in NS, and for running several early simulations.

REFERENCES

[1] Y. Wang and Q. Zhu, “Error control and concealment for video communication: a review,” *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[2] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, August 1999.

[3] H. Kanakia, P. Mishra, and A. Reibman, “An adaptive congestion control scheme for real time packet video transport,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 6, pp. 671–682, December 1995.

[4] W. Feng and J. Rexford, “Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video,” *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 302–312, Sept. 1999.

[5] 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 *Proceedings of INFOCOM*, March 1999.

[6] A. Allen, “Optimal delivery of multi-media content over networks,” in *Proc. of the ninth ACM Int. Conf. on Multimedia*, Ottawa, Canada, 2001, pp. 79–88.

[7] UCB/LBNL/VINT, “Network Simulator – ns – version 2,” *URL: http://www.isi.edu/nsnam/ns*, 1997.

[8] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, “Advanced video coding for generic audiovisual services,” *ITU-T*, May 2003.