Perceptually–Evaluated Loss–Delay Controlled Adaptive Transmission of MPEG Video over IP

G. Davini*, D. Quaglia*, J.C. De Martin† and C. Casetti‡
*Dipartimento di Automatica e Informatica
†Istituto di Ricerca sull'Ingegneria delle Telecomunicazioni e dell'Informazione—CNR
‡Dipartimento di Elettronica
Politecnico di Torino, Italy
E-mail: [davini, davide.quaglia, demartin, casetti]@polito.it

***Abstract*— This paper presents the Adaptive Video over IP (AViP) approach to transmit video sequences. A rate-selection algorithm based on both delay and loss indications is presented and its performance measured using actual MPEG-2 video sequences, networks simulations and objective measures of perceptual quality. The results show that the AViP approach leads to efficient use of available network resources, reactiveness to congestions and TCP-friendliness. Moreover, AViP delivers significantly higher perceptual levels of quality of service than traditional constant-bit-rate systems operating at the same average rate.**

I. Introduction

Video traffic is a rapidly increasing portion of the overall traffic being transmitted over IP networks. Spurred by the success of multimedia applications like video streaming and video conferencing, video is set to replicate the achievements of other real-time multimedia traffic over IP, e.g., speech (Voice over IP) and audio (MPEG-1 Layer III, i.e., MP3).

Video traffic, however, faces the same quality of service (QoS) issues of all real-time multimedia traffic over IP networks, in particular, no guarantees on delay, jitter and packet losses [1]. Moreover, the increasing weight of UDP-based multimedia traffic has raised concerns for its effects on concurrent TCP connections, concerns that are particularly acute for the case of bandwidth–hungry video traffic.

Several solutions have so far been proposed to address these challenging issues. At the network architecture level, new models introducing QoS into IP networks have been discussed, most notably the Differentiated Services Model [2].

At the application level, too, several techniques to enhance QoS have been proposed, including Forward Error Correction, data partitioning, error concealment, selective packet retransmission, and various kinds of adaptive encoding (see, e.g., [3], [4], [5]).

Among the adaptive solutions, we have recently proposed [6], together with other authors, a *rate–adaptive* approach, where a variable-bit-rate source encoder transmits at the instantaneous bit rate deemed most suitable to the current network conditions. The specific rate-selection algorithm proposed for Voice over IP applications worked on delay and loss indications sent by the receiver. The main objectives of this approach are: 1) efficient usage of network resources, 2) prompt reaction to network congestion, and 3) friendliness to concurrent TCP streams. Results also showed how the adaptive approach delivers higher perceptual quality than constant bit-rate transmission operating at the same average rate.

In this paper, an Adaptive Video over IP (AViP) approach is presented. A rate-selection algorithm specifically designed for video transmission is described in detail. Its performance is then tested using actual MPEG-2 video sequences, a network simulator and objective measures of perceptual quality. Results show how the AViP approach not only reaches the above described objectives of efficiency, reactivity and fairness, but also consistently outperforms the traditional constant-bit-rate approach. The major contributions of this paper are thus the following:

- the proposed scheme —unlike many rate-adaptation algorithms for video traffic, relying only on loss detection—introduces the delivery *delay into the control loop*;
- *perceptual quality*, both average and instantaneous, is objectively measured using actual video material.

This paper is organized as follows: in Section II, the Adaptive Video over IP approach is presented. Section III describes flow and congestion control algorithms for video streaming in the Internet. In Section IV, the proposed adaptive rate–selection algorithm is described. The simulation setup is detailed in Section V, while results are reported and discussed in Section VI. Finally, conclusions are drawn in Section VII.

II. Adaptive MPEG video over IP

The block diagram of the proposed adaptive video coding and transmission system is shown in Fig. 1. The video source feeds a variable bit-rate video encoder. The coding rate of the video encoder is determined by the adaptive algorithm depending on instantaneous network conditions, i.e., delay and packet loss rate.

We focused on the specific case of MPEG-2 video encoding [7]. Network–driven variable-bit-rate operation can be achieved in various fashions, e.g., adjusting the quantization factor, Q, or changing the kind of pictures forming the Group of Pictures, GOP, from the case of I-pictures only to more complex combinations of I-, P- and B-pictures.

In the MPEG-2 standard, an arbitrary number of consecutive macroblocks belonging to the same row is coded into a "slice"; the slice is the smallest unit which can be decoded

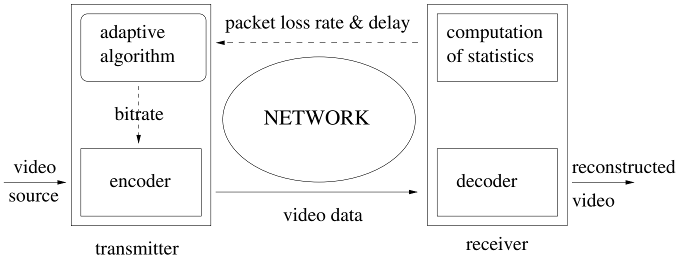


Fig. 1. Adaptive Video over IP model.

independently. Packets are created by grouping together the slices belonging to the same picture until a maximum transfer unit (MTU) is reached. In doing so, the overhead due to IP/UDP/RTP headers is minimized. According to RFC 2250 [8], a packet should contain an integer number of slices. If a slice segment is lost, in fact, the remaining part cannot be decoded because of the differential encoding of some texture info and motion vectors and the use of variable length codes.

III. Congestion control algorithms in video streaming over IP

Flow and congestion control for streaming multimedia traffic in the Internet is an issue addressed by a wide number of researchers in the past. The "standard" transport-layer congestion control protocol of the Internet, i.e., the TCP protocol, results in a stop-and-go source behavior in case of congestion and is ill-suited to handle packets with tightly-timed content. Besides, even though a media-friendly transport-layer protocol were devised, it would require the modification of scores of installed TCP/IP stacks, limiting the practical viability of such a solution.

For these reasons, having application layers handle congestion control in case of multimedia traffic is an appealing alternative, as is shown in [9], [10], as well as by many commercial products. Application-layer congestion control algorithms attempt to reduce the load on the network when congestion (i.e., queue build-up at intermediate nodes) occurs. If a large part of network traffic is composed of variable-bit-rate multimedia traffic, and the algorithm regulating each source rate is devised so as to react to specific "warnings" such as increasing delay or sudden packet losses, then a control strategy may successfully reduce the occurrence of congestion. Application-level congestion avoidance must be accomplished by looking at end-to-end metrics, such as packet loss, latency, and jitter. Packet losses are an indication of a severe network congestion: any action taken following this notification may be belated, thus leading to a prolonged congestion. However, preemptive measures may also be undertaken by acting upon an increasing-delay notification. Since the delay experienced by packets traveling from a source to a destination is a function of the number of hops, and of the queue length at intermediate nodes, a sudden increase in the delay observed by the receiver (and relayed to the sender, through periodic receiver's reports) is often followed by packet loss within the next few round-trip times. It should be pointed out that a high degree of statistical multiplexing (i.e., large number of concurring flows, large capacity) results in a more effective delay-based congestion notification.

IV. The Adaptive Algorithm

As previously noted, we focus on an application-layer implementation of congestion control algorithms. Since the IP and UDP service model do not provide for congestion or flow control, we need a reliable, end-to-end way of estimating the state of the network: namely, the available end-to-end bandwidth, largely determined by the bottlenecks along the connection, and the onset of temporary congestions. Based on such state, the rate control algorithm will select bit rates compatible with the estimated bandwidth and react to the throttling of one or more links.

Adaptive algorithms can act upon several aspects of a multimedia streaming connection, i.e. the coding algorithm, the packet generation rate, the packet length, or the playout delay at the receiver. The first three parameters can be chosen by the transmitter, while the latter can be adjusted by the receiver. Specifically, in the case of MPEG-coded video streaming, the coding algorithm imposes some constraints upon the packet generation rate, forcing average rate changes to occur only on GOP boundaries. As a consequence, the time it takes for the algorithm to react to congestion indications can be as long as the duration of a GOP. Beside the source transmission rate, the coding algorithm also affects other aspects of multimedia communication, while the packet length trades off transmission overhead and packetization time (and thus, delay). The playout delay can be controlled at the receiver by proper regulation of the playout buffer length. The size of the playout buffer can be fixed, determined at call start-up and then kept constant, or allowed to change in the course of the communication to adapt to variations of the behavior of the network. The main trade-off is between degradation due to delay versus degradation due to packet losses: on the one hand, longer buffers decrease the number of packets discarded because of late arrival (packets are discarded at the receiver if they are "too" late), at the expense of increased delay; on the other hand, shorter buffers can curb the playout delay, but at the expense of increased packet loss rates.

The rate control algorithm described in this paper, starting from delay and loss measurements relayed by RTCP reports, tries to regulate the output rate of several video sources. It uses the information that can be carried by periodic RTCP receiver reports to let the source know the state of the on-going connection. The control algorithm runs at the source itself. The estimation of delay to adapt the rate of video sources was already used in [10], although it was not included in the feedback loop, as in our case, but it affected the dynamical determination of the additive increase rate.

The adaptive algorithm follows the "additive increase, multiplicative decrease" paradigm that is successfully employed by many congestion control algorithms, such as TCP or ABR. The algorithm principles, as outlined in Fig. 2, provide for

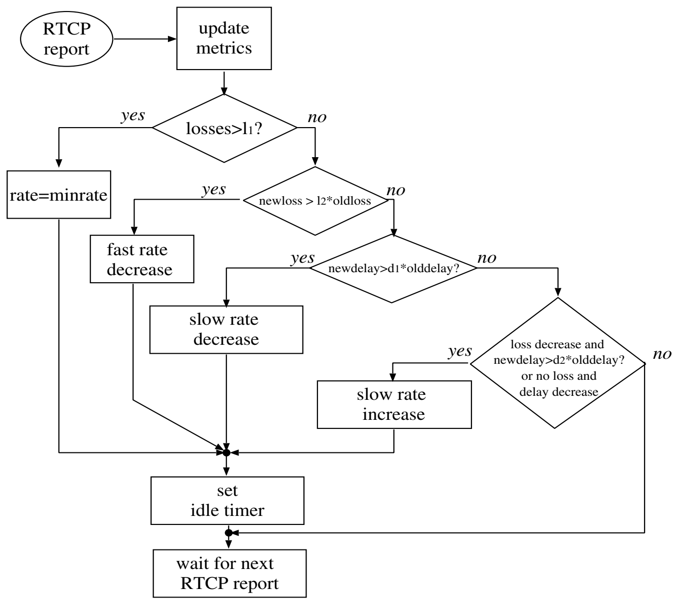


Fig. 2. Flow-chart of the adaptive Video-over-IP rate–selection algorithm.

the source coder to increase or decrease its rate depending on delay and loss estimates inferred from RTCP data. Specifically:

- the coder should drastically *reduce* its rate to the lowest possible rate if losses have been reported to be higher than a 'high-mark threshold' l_1 by the latest RTCP message;
- a strong rate *reduction* (unless the rate already amounts to the lowest possible value) is suggested if losses have been estimated to have increased by a factor l_2 with respect to the latest RTCP message;
- the coder should mildly *reduce* its rate when packet delays have been observed to have considerably increased, at least by a factor d_1 compared to the latest RTCP message;
- the coder should operate a mild rate *increase* if the following conditions are verified: either losses are decreasing (by an unspecified amount) and the delay has decreased by at least a factor d_2 compared to the latest RTCP message, or no losses are reported and the experienced delay is smaller than the one previously reported.

To allow some *idle* time so that rate changes are effective, the source coder waits for a time t_d before performing a rate decrease following any previous rate change; a time t_i elapses between any rate change and a rate increase.

V. Simulation Setup

Simulations were run on the *ns* network simulator [11] using sources based on actual MPEG-2 video sequences.

A. MPEG Video Sources

We tested the performance of the proposed adaptive strategy with a video sequence produced by a digital videocamera and encoded in MPEG-2 format. To simulate the use of a variable bit-rate video encoder, the same video material was encoded at nine different bitrates, from 3.2 Mb/s to 5.6 Mb/s, with a 0.3 Mb/s step-size. We used a modified version of the ISO MPEG-2 Test Model 5 [12] with an improved rate-control algorithm; coding was performed using the Main Profile, Main Level. Every GOP is independent of the others (closed GOP) and consists of 13 pictures; the GOP pattern (in display order) is $IBBPBBPBBPBBP$.

Since the frame rate is 25 frame/s, the video sources produce a frame every 40 ms. The corresponding packets generated by the video encoder are transmitted equally spaced in time. The adaptive algorithm can change the bitrate only at GOP boundaries and, therefore, its maximum latency is 520 ms. The packetization process assumes that the maximum transmission unit is 1500 bytes. In our tests, we obtained an average packet size of 1358 byte with low standard deviation. The original length of the video sequence is 300 s. The video sequence was repeated ten times to reach an overall simulation time of 3,000 s. Sources start-up times are staggered between 0 s and 300 s. The initial rate set by the adaptive algorithm is 4.4 Mb/s.

B. Network topology

The network topology features a simple two-node bottleneck with a 5-ms source-to-destination propagation delay. The first bottleneck node has a capacity of C Mb/s and a buffer size of B bytes. N video sources are connected to the first bottleneck node by a 100-Mb/s link. Results are reported for two different choices of the above parameters: Scenario 1 includes $N = 15$ sources, a bottleneck capacity of $C = 82.5$ Mb/s and a bottleneck buffer size of $B = 384$ Kbytes; in Scenario 2 we chose $N = 10$, $C = 55$ Mb/s and $B = 256$ Kbytes. The values of C were chosen so that sources sending at their maximum rate of 5.6 Mb/s would nominally saturate the bottleneck link.

C. Algorithm parameters

The loss threshold l_1 was set to 7%; this choice was based on preliminary perceptual quality measurements. The other parameter values are the result of a heuristic optimization. They are: loss factor l_2 = 0.2, delay factor d_1 = 0.1, delay factor d_2 = 0.1.

VI. Simulation Results

Our simulations aimed at showing the following four properties of the adaptive algorithm:

1) *Greedy bandwidth utilization*: when the network is not congested, the algorithm will try and increase the source rate;
2) *Reactivity to changing conditions*: the algorithm will quickly react to network congestions by reducing the coding rate;
3) *Fairness and TCP-friendliness*: the algorithm leads to fair share of network resources among sources using the same adaptive strategy; also, TCP connections will not be severely slowed down, let alone shut down, by the adaptive-rate video sources;
4) *Gain over constant-bit-rate sources*: when compared with a constant-bit-rate video source that sends at approximately the same average rate of adaptive sources.

0-7803-7802-4/03/$17.00 (C) 2003 IEEE

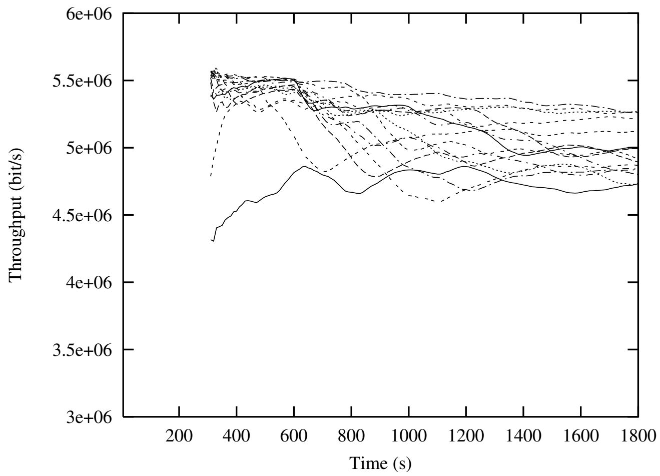


Fig. 3. Throughput of 15 AViP sources as a function of time; concurrent traffic: 3 continuous FTP sources; network scenario #1.

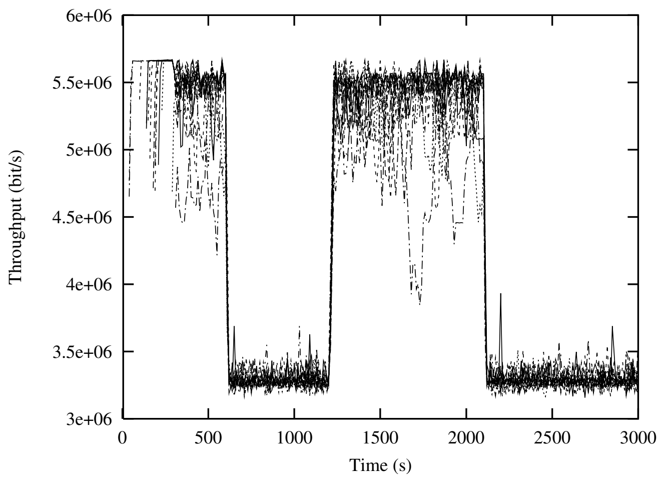
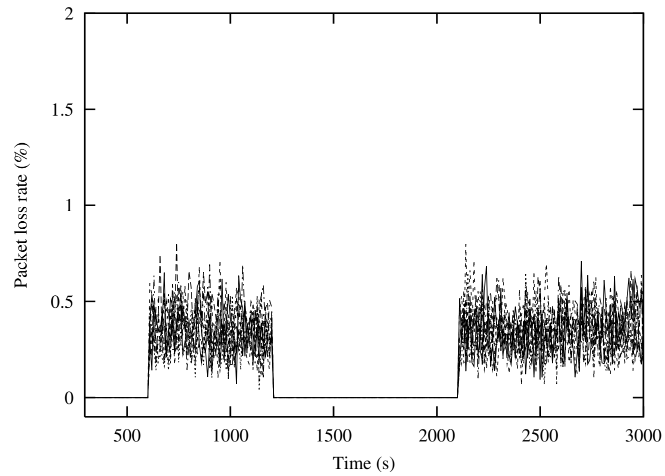


Fig. 4. Throughput of 15 AViP sources as a function of time; concurrent traffic: 5 greedy on/off (600s–1200s–2100s) FTP sources; network scenario #1.

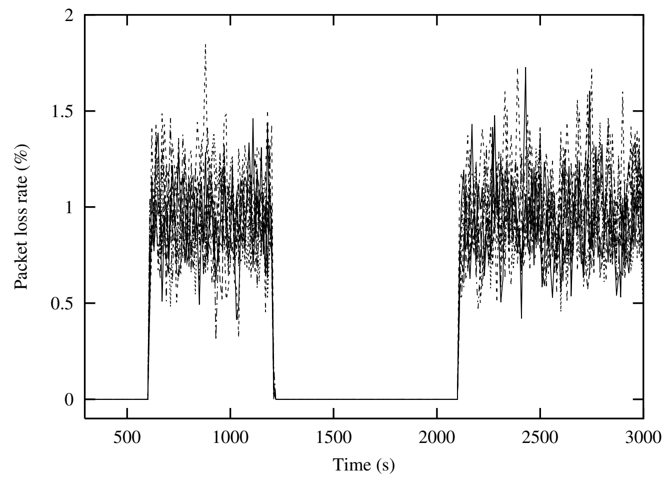
Due to space limitations, we will only present a limited set of results that show the above four properties of the algorithm; however, several other experiments confirm the findings.

Fig. 3 was obtained simulating Scenario 1 and adding 3 FTP sources that become active after 600 s. The FTP sources were using a window-limited TCP connection, that did not allow their throughput to exceed 9 Mb/s. Fig. 3 shows the average throughput experienced by each of the 15 video sources and underscores both Properties 1 and 3: not only does the adaptive algorithm keep the sending rate around the upper end of the rate range, providing a 97% link utilization, but it also ensures a fair network sharing: the difference between source rates never exceeds 10% of the link rate. As for the background FTP connections, their throughput reduction is 9.13% with respect to the case in which there are no concurrent AViP sources.

Fig. 4, again referring to Scenario 1, shows the adapting behavior of AViP sources in presence of 5 greedy FTP connections starting at time 600 s, then going off at time 1200 s, and finally starting again at time 2100 s. Property 2 is clearly shown in this Figure, which also reaffirms the algorithm's ability at providing a fair treatment to all AViP sources: only one source during the second inactivity interval appears to fluctuate while climbing back to full rate.



(a)



(b)

Fig. 5. Packet loss rate as a function of time for (a) AViP and (b) ViP video sources; concurrent traffic: on/off FTP sources; network scenario #2.

The last Property, i.e., the gain over the constant-bit-rate approach, was explored by looking at packet losses and perceptual quality of the received video streams. Fig. 5 and Fig. 6 compare the instantaneous loss rates of concurrent video and On/Off FTP sources in Scenario 2. It can be seen that, on average, the loss rate is two to three times larger for non-adaptive than for adaptive sources. The FTP flows, also, are clearly more affected by the presence of non-adaptive video sources.

Table I reports perceptual quality measures of an adaptive video source compared with a non-adaptive one for the same case of concurrent On/Off FTP sources. Average PSNR with respect to the original uncompressed frames is reported for

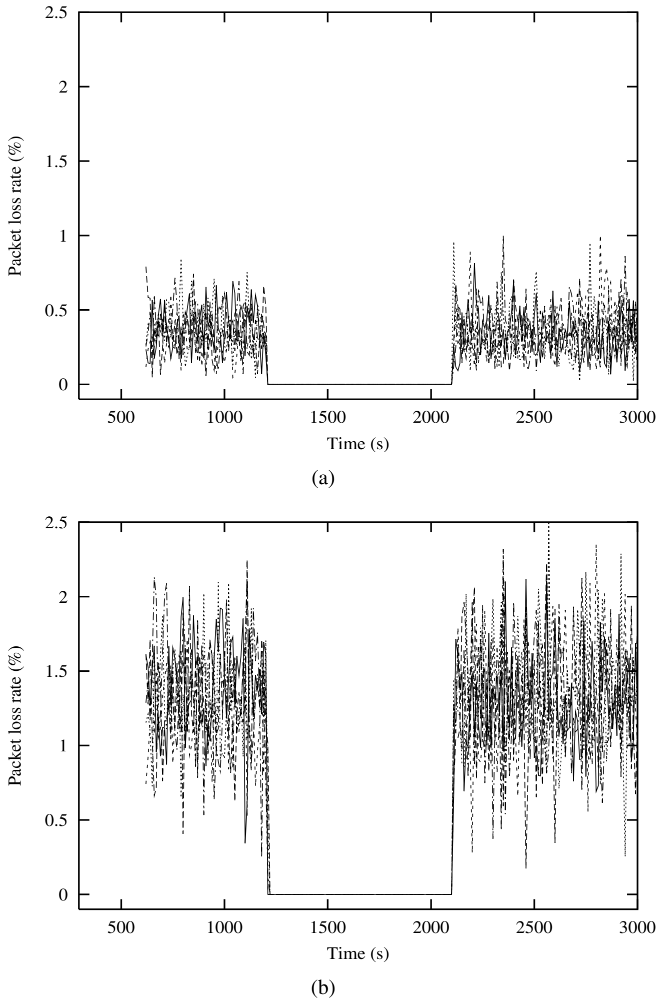


Fig. 6. Packet loss rate as a function of time for on/off FTP sources; concurrent traffic: (a) AViP and (b) ViP video sources; network scenario #2.

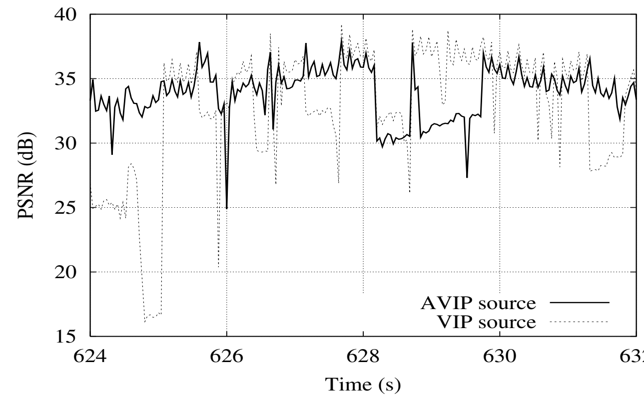


Fig. 7. Perceptual performance as a function of time, AViP vs. ViP (both at 4 Mb/s), during a network congestion.

TABLE I

PERCEPTUAL PERFORMANCE AVIP VS. VIP, OVERALL (0–3,000 S) AND DURING A NETWORK CONGESTION (624–632 S); SCENARIO #2

	Time Interval (s)	PSNR (dB) Avg	PSNR (dB) St. Dev
Adaptive ViP	0–3,000	43.2	7.1
Constant ViP	0–3,000	42.8	7.6
Adaptive ViP	624–632	33.9	2.0
Constant ViP	624–632	32.7	5.1

both the whole simulation (0–3,000 s) and the interval after the FTP sources start to transmit (624–632 s). Instantaneous PSNR values for both video sources during this shorter interval are shown in Fig. 7. The adaptive strategy leads to a higher perceptual quality with respect to the non-adaptive approach and it is particularly effective, as desired, when network conditions change; in this case the gain is greater than 1 dB. Moreover, the PSNR standard deviation is clearly lower for the adaptive source, showing that in case of network congestions the adaptive algorithm delivers more stable perceptual quality than the non-adaptive source.

VII. CONCLUSIONS

We presented a new algorithm to dynamically select the best transmission rate of a variable-bit-rate video encoder when network conditions vary. The proposed technique was tested under various network scenarios using actual MPEG-2 video sequences, a network simulator and objective quality measures of the received video data. The proposed Adaptive Video over IP (AViP) technique consistently delivered better perceptual quality than the traditional constant-bit-rate approach, while exhibiting efficient use of network resources, fairness to concurrent flows and prompt reactivity to network congestions.

REFERENCES

[1] D. Wu, Y. Hou, and Y. Zhang, "Transporting Real-Time Video over the Internet: Challenges and Approaches," in *Proceedings of the IEEE*, vol. 88, December 2002, pp. 1855–1875.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, December 1998.

[3] N. Yin and M. Hluchyj, "A dynamic rate control mechanism for source coded traffic in a fast packet network," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1003–1012, September 1991.

[4] J. Bolot and T. Turletti, "A rate control mechanism for packet video in the internet," in *Proceedings of IEEE INFOCOM*, June 1994.

[5] D. Kozen, Y. Minsky, and B. Smith, "Efficient algorithms for optimal video transmission," in *Proceedings of Data Compression Conference*, March–April 1998, pp. 229–238.

[6] A. Barberis, C. Casetti, J. D. Martin, and M. Meo, "A simulation study of adaptive voice communications for ip networks," *Computer Communications Journal, Elsevier*, vol. 24, no. 9, pp. 757–767, May 2001.

[7] ISO/IEC, "MPEG-2 generic coding of moving pictures and associated audio information," *ISO/IEC 13818*, 1996.

[8] R. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," *RFC 2250*, January 1998.

[9] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," in *Proceedings of 1st Int. Workshop on High Speed Networks and Open Distributed Platforms*, June 1995.

[10] D. Sisalem and H. Schulzrinne, "The loss-delay adjustment algorithm: A TCP-Friendly adaptation scheme," in *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video*, July 1998.

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

[12] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, vol. 2419, pp. 100–118, Apr. 1995.