In the Proceedings of the IEEE Packet Video Workshop, March 2000.
FORMATTED FOR PRINTING ON A4 PAPER.

High-Bandwidth Internet Video Telephony

Sergio D. Servetto Martin Vetterli*

Laboratoire de Communications Audiovisuelles
Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.
http://lcavwww.epfl.ch/~{servetto,vetterli}/

Abstract

We consider the problem of full-duplex video transmission over the Internet, under real-time delay constraints. In this work we present the design of a complete communications system, whose salient features are: (a) a soft real-time transport protocol, whose generated traffic characteristics are identical to those of bulk transfers using TCP/IP, thus ensuring complete fairness to other flows in the network; (b) a video coder resilient to moderate amounts of loss of data that are induced by the soft real-time constraint, and also by a continuous probing of the available channel capacity; (c) a unified control module, responsible for both congestion avoidance as well as coder configuration tasks; and (d) very low computational complexity. A key component of our system design is a predictive multiple description video coder working under feedback control. The whole system is implemented on a Linux PC, only in software, without making use of priviliged system calls (e.g., to raise the priority of a process). Using our prototype system, over certain high-speed segments of the public network we were able to deliver YUV signals of size 352 × 240 pixels, at about 15-25 frames/sec, and at average bit rates in the range of 0.5-1.5 Mbits/sec.

1 Introduction

1.1 Video Transmission under Real-Time Delay Constraints

In this paper we study the problem of transmission of good quality video signals over the public Internet under real-time delay constraints. Our interest is in the development of techniques suitable for high-bandwidth telephony systems, in which encoding and decoding can be done only in software, on relatively inexpensive computing platforms (e.g., a fast Linux PC with a frame grabber and a sound card).

Compared to other problems in real-time visual communications (e.g., streaming for Video On Demand applications [14], or Video Multicast [8, 18]), the telephony problem is a particularly difficult one. In this application the signal to transmit is not available a priori, cannot be buffered before transmission, and needs to be reconstructed at the receiving end within a short period of time after its capture at the transmitting end.[1] This poses severe limitations on (a) the complexity of the encoding/decoding techniques and (b) the nature of the transmission protocols that can be effectively used. However, by tightly coupling the design and the control operations of the video coder with the

*M. Vetterli is also with the Dept. of EECS, UC Berkeley.

[1] Typically within 0.1-0.2 sec (encoding, queueing, propagation, and decoding times included).

underlying network infrastructure, a significant amount of channel state information can be made available to the encoder to improve the overall system performance.

This last observation is yet another indicator of something we have argued about extensively in previous work: the fact that the problem of providing real-time services over packet networks naturally calls for joint source/channel coding methods [3, 8, 14]. Indeed, even disregarding practical issues related to the complexity of encoders and decoders (as argued above), real-time communication over packet networks is a prime example of a situation in which the assumptions of the Source/Channel Separation Theorem [1, Ch. 8] are not satisfied: that theorem does not hold in the presence of delay constraints, in the presence of channel uncertainty, and for multiple access channels. Our main goal in this paper then is to explore, in a practical and implementable setting, the performance of video telephony systems designed around joint source/channel coding principles, capable of delivering signals of quality substantially higher than what is currently possible using low bit rate systems. These concepts are illustrated in Fig. 1.

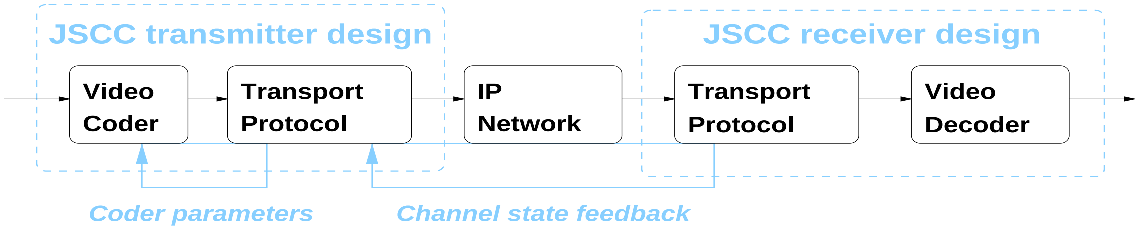

Figure 1: Design of communications systems for real-time video transmission based on joint source/channel coding principles. The goal here is to design a video coding algorithm that explicitly uses channel state information to boost its performance.

1.2 High-Bandwidth Systems without Network QoS

Two important assumptions made throughout this work are (a) the availability of a reasonably large amount of bandwidth (say, above 500 Kbits/sec), and (b) that the network does not provide any form of Quality of Service (QoS) guarantees. These assumptions can be justified based on a number of factors:

- A significant amount of work has been done already in the low bit rate coding area. Under the assumption of a clean channel, it seems unlikely at this point that performance drastically superior than that attained, e.g., by the H.263 standard, can be attained by some other carefully designed coder. However, interesting research questions come up when considering higher bit rates and dirty channels. For example, it would be unacceptable to temporarily reduce the frame rate of a full-motion, CD-ROM like quality video sequence from 20-30 frames/sec to 10-15 frames/sec because a channel error occured: but this is exactly what one of the error-resilience modes for the H.263 standard does, certainly with good results in its target range of bit rates. Also, in networks with a large bandwidth/delay product and congestion control based only on end-to-end packet acknowledgements, during a roundtrip time the state of the channel can change substantially, and it may not be possible to infer what these changes are just from looking at the acks received: as a result, the dynamics of the packet loss process differ significantly from those in the low bandwidth regime.

- Certain segments of the current public Internet provide very high transmission speeds.[2] Until the massive deployment of routers capable of providing Integrated and Differentiated Services takes off [19], it seems reasonable to assume that the Internet will continue to evolve as it has done over the last years, by simply adding bandwidth to the current infrastructure, but without any form of network QoS support. And even if the QoS-enabled Internet becomes a reality, pricing considerations and/or multiple users falling into a common service class still make the best-effort only case a most important one to consider.
- We feel that a major factor preventing the massive deployment of video telephones is precisely the low quality of the signals delivered by current low bit rate systems.

We see therefore that the design of systems capable of providing real-time services over high speed –but otherwise completely uncooperative– networks is a most important practical problem. Furthermore, even disregarding practicality considerations, this problem is also a good source for challenging research questions to work on.

1.3 Main Contributions and Organization of the Paper

The main contribution presented in this paper is the design, implementation and testing of a complete video telephony system. Our proposed system works over moderately high-bandwidth Internet connections, can be implemented only in software on a fast linux PC, and is capable of delivering video signals of the quality currently attained by applications involving playback of CD-ROM stored video.

The rest of this paper is organized as follows. In Section 2 we present a block diagram outlining the main components of our system, and discuss their functionality. In Section 3 we present our modifications to the standard Internet transport protocol, and in Section 4 we present the design of a video coder to be used in this context. In Section 5 we present experimental results obtained in actual network transmissions. Concluding remarks are presented in Section 6.

2 System Architecture

In the context of high bandwidth telephony, we implement the JSCC transmitter/receiver designs of Fig. 1 using a predictive multiple description video coder, whose operating parameters are determined under feedback control. The resulting system architecture is shown in Fig. 2.

The most important components in this system are the error-resilient video coder, and a single module that simultaneously performs both congestion avoidance and coder control decisions. Details on these two are presented in the next two sections.

[2]For example, the campus networks of almost every major university in the US are connected via vBNS (`http://www.vbns.net/`), an experimental backbone with *gigabit* capacity. In Switzerland, a handful of universities are interconnected via SWITCH (`http://www.switch.ch/`), the ATM backbone of the Swiss National Science Foundation, at capacities ranging between 34-155 Mbits/sec.

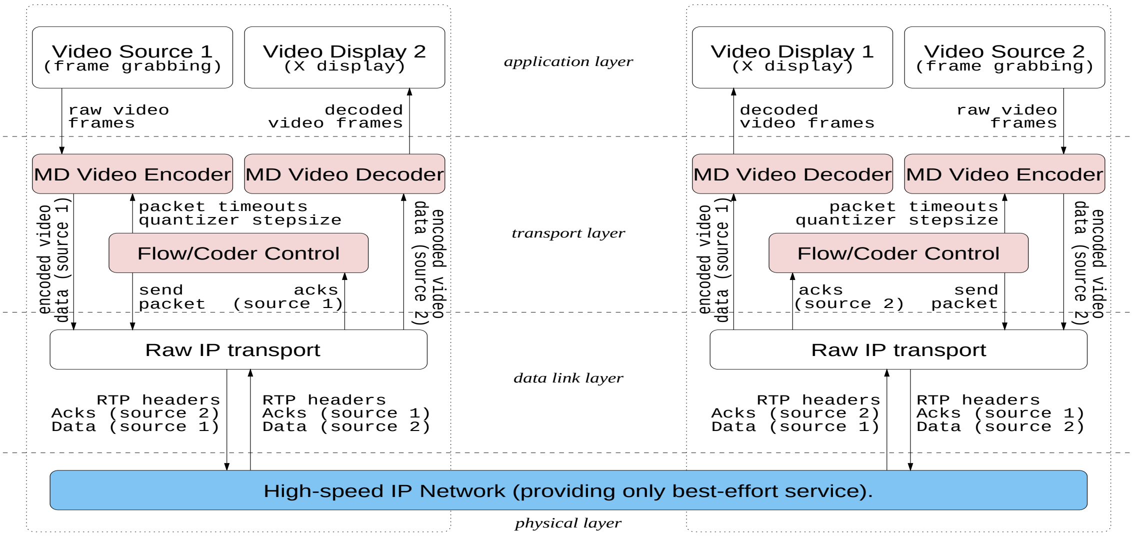

Figure 2: System architecture. Since the coding and flow control tasks are highly interdependent, we regard these as different sub-modules of the transport layer.

3 Congestion Avoidance and Coder Control

3.1 TCP Friendliness

The current Internet works under the assumption that all end-systems react to congestion by adjusting accordingly their packet injection rates. Doing so has a number of benefits, such as keeping network utilization high, or preventing congestion collapse. Another benefit is that of sharing in a fair manner the instantaneous available bandwidth among multiple competing flows, implemented using possibly different protocols. But since most current Internet traffic is based on the standard Transmission Control Protocol (TCP), it is crucial that any new protocol remains fair to TCP. Congestion avoidance and control is an old networking topic, and a wealth of knowledge is already available on this topic. However, the literature on soft real-time, "TCP-friendly" protocols is not as rich.

Perhaps the simplest way of ensuring TCP friendliness for the transport of delay-constrained data would be to just eliminate retransmissions from TCP, but otherwise use it as is. Until recently however, this approach had not led to feasible system designs, since without retransmissions –or else some other mechanism that can ensure error-free transmission– classical *layered* video coders take a severe performance hit in the presence of even a small amount of randomly placed packet losses. A system for streaming video based on a TCP-friendly protocol and a layered coder is presented in [11, 12]. In [14, 15, 16], we proposed the first system based on a video coder robust to randomly placed packet losses. A system based on similar principles was then presented in [7], which combines the error-resilience method of [9] with the congestion control method of [23].

Being able to use TCP's flow control as is (may be with non-essential changes) has a definite advantage: as far as the network is concerned, the traffic generated by such a source is no different from the traffic generated by bulk TCP transfers. We are motivated to take this "evolutionary" approach to the transport of delay-constrained data (as opposed to "revolutionary" approaches, in which entirely new protocols are built from scratch), by the simple practical observation that TCP's flow control has delivered outstanding network performance over many years, not just in simulations but in actual transmissions performed by a very large number of Internet users. Because of this, we feel it is a good idea to keep flow control as it is now and build applications on top, instead of replacing it with entirely new controllers [14, 15].

3.2 R-TCP: A Modified TCP for Delay-Constrained Data

The transport and control methods used in this system –dubbed R-TCP, for Real-time TCP, following [14]– essentially consist of:

- IP packets carrying RTP headers [13].[3]
- Elimination of retransmissions: upon timeout of a packet a fresh new one is sent, instead of retransmitting the lost one.
- Redefinition of the meaning of ack messages, to accomodate the fact that some packets may be missing.
- A controller with four state variables:
  - `cwnd`, size of the congestion window.
  - `ssthresh`, slow-start threshold.
  - `qstep`, quantizer stepsize.
  - `lastTimeout`, identifier of the last unacknowledged packet.

In the controller, `cwnd` and `ssthresh` are exactly as defined in TCP's flow control algorithm [5], and follow exactly the same dynamics: this is how a complete emulation of bulk TCP traffic is achieved. In `lastTimeout` the id of the last unacknowledged packet is kept, information that will be used by the robust coder. And `qstep` is updated after completing the transmission of each encoded video frame: it is increased by a fixed amount (to reduce the encoding rate) if the transmission time for that frame exceeded 1/<frame_rate> seconds, decreased by a fixed amount otherwise.

Note that, ideally, one would *search* for the value of `qstep` leading to the best reconstruction quality subject to a constraint on the number of packets that the network can deliver in time. However, such optimizations are too complex to be performed in real-time, and so what we do instead is to adjust `qstep` incrementally.

4 Error-Resilient Video Coding

4.1 Robust Video Coding based on Multiple Descriptions

Due to the elimination of retransmissions, the proposed transport protocol cannot ensure the delivery of every transmitted packet, can only ensure the delivery of *most* transmitted packets.[4] As a result, the compressed bit stream needs to be robust to the presence of errors of this nature. And for this purpose we design a robust video coder based on

[3]Whereas there is no compelling a-priori reason to use the format specified by RTP in our system, we have chosen to go with it because this is the standard format used by Internet Telephony applications [20].

[4]Besides, a moderate amount of lost packets is desirable anyway, since without network feedback a continuous probing of the available bandwidth is the only way to make efficient use of network resources.

Multiple Descriptions (MDs) [2, 21]. The basic idea behind MDs is to design coders for systems that employ diversity to overcome channel impairments. A source signal is encoded into multiple bit streams (descriptions), one to be sent over each channel. These descriptions have a property known as *mutual refinability*: the signal decoded out of individual descriptions is of good quality, and different descriptions can be combined to decode a signal of higher quality, thus ensuring graceful degradation in the presence of channel failures.

In the context of Internet transmission, the use of a model based on multiple channels requires some justification: sources have little influence on the route followed by a packet (most packets will follow the same route over a period of time), and therefore there is no "physical" interpretation for this model. The justification however comes from the mechanisms for fragmentation and reassembly of packets by the IP protocol. A large packet is fragmented into a number of smaller packets, and these small packets are transmitted independently; then, at the receiving end, if any one of the small packets is lost the entire large packet is discarded, and if all arrive then the large packet is reassembled and delivered to the user. Since large packets are lost whenever any of the small fragmented packets are lost, we see that the probability of losing a large packet increases with its size. Therefore, the multiple channel model in this case corresponds to the transmission over the same physical channel of a single message broken into a number of smaller packets. Creating multiple descriptions of a video signal fitting into these small packets having the mutual refinability property is ideally matched to transmission over a network without any form of prioritization among packets.

MD coding of still images has been considered recently in [4, 6, 17, 24]. Extensions to video coding have been considered in [14, 15] (in the context of subband coding), and in [10, 22] (in the context of predictive coding).

4.2 Predictive Multiple Description Coding with Feedback

A nontrivial problem in the design of predictive MD video coders is that of encoding prediction errors. The difficulty stems from the fact that a number of different signals may or may not be available to the decoder to form predictions, and which ones are available is not known to the encoder. As a result, the encoder may encode a prediction error for which the decoder does not have the basis on which to predict, resulting in loss of synchronization and in a (potentially large) increase in distortion.

Since the problem of loss of synchronization is caused by some information not being available at the decoder, to prevent it from happening there is a cost to be paid in terms of increased bit rate relative to that of a pure MD system. The approach proposed in [22] increases bit rate by designing predictors for the coarse-quantized single channel data as opposed to the fine-quantized two channel data. The approach proposed in [10] increases bit rate by encoding various mismatch signals that result in all different cases in which the encoder predicts based on something different from the decoder.

In the context of packet video, there is often an extra source of information: a feedback channel, carrying packet acknowledgements. Acks, used by the flow control algorithm for an entirely different purpose (and therefore available for free to the encoder), explicitly reveal which exactly is the signal available at the decoder to compute predictions. This observation motivates us to consider an alternative structure for predictive MD coding. In our system, predictions are computed always based on the most recent output of the two-channel decoder, except when a packet timeout occurs, in which case the prediction is formed based on the most recent output of the two-channel decoder *acknowledged prior to this timeout.* A block diagram of our encoder is shown in Fig. 3.

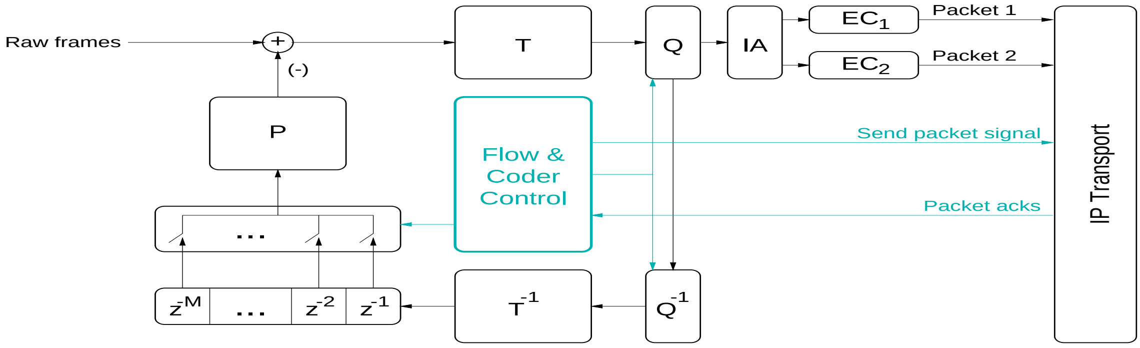

Figure 3: Block diagram for the encoder. A prediction of the current signal is formed based on a previous frame: the most recent one if no packet timeouts occured, else the most recent acknowledged one prior to the last timeout. Then the prediction error is transformed with a linear transform, and the resulting coefficients are quantized with a scalar quantizer. Each of these coefficients is split into two descriptions using the staggered index assignment of [21] (the one which is most robust to errors), entropy coded, and sent to the channel. The decoder is a straightforward MD decoder.

Note that in this structure, the timeliness of feedback may be crucial. The longer feedback is delayed, the farther apart the ack'd frame based on which a prediction is computed to resynchronize and the current one are: as a result, coding efficiency takes a hit, and a temporary loss of synchronization occurs. Although theoretically possible, experiments in Section 5 will show these issues to have a negligible practical impact.

4.3 Implementation Issues

A number of choices for the components of the block diagram from Fig. 3 were dictated by the requirement that the end system runs in real time. Note that this is a fairly tight constraint: on a single machine we need to run (a) a frame grabbing application, (b) a video encoder, (c) two video decoders (one for the outgoing signal, one for the incoming signal), and (d) a video display application. Here we discuss a number of issues related to our implementation:

Cache-Efficient Haar Trees and Quantization using Integer Operations Only. The transform chosen is a 2D Haar subband decomposition: the 2D array $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ is transformed into $\frac{1}{2}\left[\begin{smallmatrix} a+b+c+d & a+c-b-d \\ a+b-c-d & a+d-b-c \end{smallmatrix}\right]$, and the division is implemented as a shift to the right by 1. The processing is done on blocks of size $2^k \times 2^k$ (k is the depth of the trees) at a time, instead of on each frequency subband at a time, to keep these small blocks in cache memory and minimize the number of references to main memory. Quantization adds one integer division to this process.

Index Assignments using Integer Operations Only. Whereas in their full generality the index assignments designed in [21] need to be implemented as a search by primary key in a pre-stored table, by restricting our attention to the staggered configuration a much simpler (and faster) algorithm is possible, whose operation is explained in Fig. 4. Given an integer bin n, the two descriptions are computed as

-4	-3	-3	-2	-2	-1	-1	0	0	1	1	2	2	3	3	4	4	→ to Packet 1
-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	
-3	-3	-2	-2	-1	-1	0	0	1	1	2	2	3	3	4	4	5	→ to Packet 2

Figure 4: Staggered index assignment for MD coding. Two quantizers of stepsize 2Δ can be aligned such that from the bins of these two a new quantizer of stepsize Δ can be created. In our case, the fine integer grid corresponds to the output of the quantizer denoted Q in Fig. 3.

$n_1 = \frac{n}{2} - (\text{odd}(n) \wedge n < 0)$ and $n_2 = \frac{n}{2} + (\text{odd}(n) \wedge n > 0)$. From these two descriptions, we see that $n = n_1 + n_2$, and that the midpoint of the interval containing n_1 is $2n_1 + \frac{1}{2}$, and for n_2 it is $2n_2 - \frac{1}{2}$. All these operations are trivially implemented as integers adds, shifts and comparisons.

Selective Entropy Coding. Entropy coding is performed by putting through an arithmetic coder a pre-specified subset of the Haar trees. For the sequences of size 352×240 considered in this work, the maximum tree depth on the Y component is 4, on the U and V components is 3. In a given frame, Y blocks of size 16x16 are split into 4 subblocks of size 8x8, the upper left, upper right and lower left blocks are each coded relative to their own probability model, and the bottom right block is discarded. The U and V components are split into 4x4 blocks, but in this case only the upper left blocks are coded, all others are discarded.

Zero-Order-Hold Predictor. Motion estimation is still too complex to perform under the low-complexity constraints of this problem, and therefore the previous decoded image is taken to be the prediction for the current image. Coding raw frame differences may produce not very good results on high-motion-type sequences, but on telephony-type sequences this is not a problem.

5 Experimental Results

Results are presented here on the performance of the proposed video coder under packet losses. We compare the average PSNR of the reconstructed signal when there is no packet loss against the case when a loss occurs at the third frame (out of 120), without ever resynchronizing. This is done to measure how the signal quality degrades, should a packet acknowledgement get delayed. PSNR and bit rate plots are shown in Fig. 5, and sample reconstructions are shown in Fig. 6.

Observe how, even after a few seconds, the PSNR drop due to loss of synchronization appears to be stable around 0.5 dB for the MD coder, whereas it is much worse for the non-robust system (about 3 dB). This quality drop can be recognized visually too, in the form of a ghost image for the non-robust system in Fig. 6. Since even between sites in the US and Switzerland roundtrip times are around 0.15 seconds (and well below 0.1 seconds among vBNS hosts), this shows that in most practical cases, the PSNR performance of our coder is (a) essentially that attained under no packet losses, and (b) unlike the non-robust case, after a packet is lost and until encoder and decoder resynchronize no annoying visual artifacts occur, even if acks get delayed.

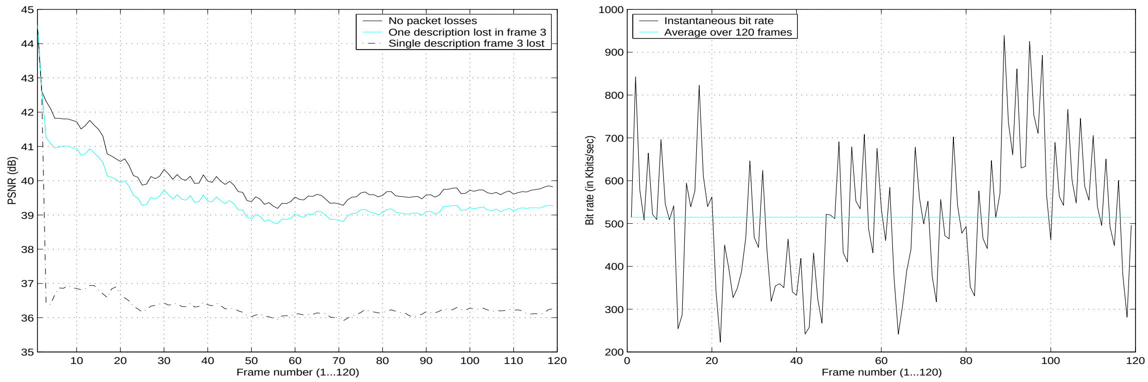

Figure 5: Robustness to errors. Left: PSNR plot for three cases = (a) no lost packets, (b) one packet loss for the robust coder, (c) one packet loss for an equivalent non-robust coder that uses a regular scalar quantizer instead of a MD quantizer; Right: bit rate for a fixed `qstep` = 10.

Figure 6: Reconstruction of frame 90 for the encoding of Fig. 5. Left: MD coder; right: single description equivalent coder (obtained by removing the index assignment from the coder in Fig. 3). Notice the ghost around the face in the picture to the right, not there in the other.

6 Conclusions

In this paper we presented an overview of our work on the design of communications systems for video telephony, in the context of high-bandwidth Internet transmission. We explained why this is a most important practical problem to consider, presented the architecture of our system, and showed preliminary experimental results which reveal the great potential of our proposed techniques.

An important conclusion that we come to is related to the design of MD video coders. Recent work on predictive MD coding techniques was motivated by the need to encode motion compensation residuals in a MD context [10, 22]. In this work we present an alternative solution to the same problem, much simpler, which is applicable whenever a feedback channel is available (as in the very important case of unicast Internet video).

Our current work focuses essentially on further optimizing for speed the performance of our prototype. For example, in detailed studies (not reported here) of the distribution of CPU cycles among the different modules in our current implementation of this system, we have found that more than 60% of the time is spent inside the arithmetic coder. On the other hand, the combined forward and inverse transforms and quantizers consume about 30% of the time only, thus suggesting the effectiveness of the optimizations discussed above in this regard. Hence, we see that may be at the cost of a slight degradation in compression efficiency, fast, table-based entropy codes may potentially result in large savings in CPU time. We are currently exploring such issues, as our end goal with this project is to eventually convert this prototype into a stable, real, usable system.

References

[1] T. M. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., 1991.

[2] A. A. El Gamal and T. Cover. Achievable Rates for Multiple Descriptions. *IEEE Trans. Inform. Theory*, IT-28(6):851–857, 1982.

[3] M. W. Garrett and M. Vetterli. Joint Source/Channel Coding of Statistically Multiplexed Real-Time Services over Packet Networks. *IEEE/ACM Trans. Networking*, 1(1):71–80, 1993.

[4] V. K. Goyal, J. Kovačević, R. Arean, and M. Vetterli. Multiple Description Transform Coding of Images. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Chicago, IL, 1998.

[5] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, 1988.

[6] W. Jiang and A. Ortega. Multiple Description Coding via Polyphase Transform and Selective Quantization. In *Proc. SPIE Visual Commun. Image Proc. Conf. (VCIP)*, San Jose, CA, 1999.

[7] K.-W. Lee, R. Puri, T.-E. Kim, K. Ramchandran, and V. Bharghavan. An Integrated Source Coding and Congestion Control Framework for Video Streaming in the Internet. In *Proc. IEEE INFOCOM*, 2000.

[8] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video.* PhD thesis, University of California, Berkeley, 1996.

[9] A. E. Mohr, E. A. Riskin, and R. E. Ladner. Graceful Degradation over Packet Erasure Channels through Forward Error Correction. In *Proc. IEEE Data Compression Conf. (DCC)*, Snowbird, UT, 1999.

[10] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri. Multiple Description Coding for Video Using Motion Compensated Prediction. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Kobe, Japan, 1999.

[11] R. Rejaie, M. Handley, and D. Estrin. Quality Adaptation for Congestion Controlled Video Playback over the Internet. In *Proc. ACM SIGCOMM*, 1999.

[12] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proc. IEEE INFOCOM*, 1999.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications.* RFC 1889, available from `http://www.ietf.org/`.

[14] S. D. Servetto. *Compression and Reliable Transmission of Digital Image and Video Signals.* PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[15] S. D. Servetto and K. Nahrstedt. Broadcast Quality Video over IP. Submitted for publication. Available from `http://lcavwww.epfl.ch/~servetto/`.

[16] S. D. Servetto and K. Nahrstedt. Video Streaming over the Public Internet: Multiple Description Codes and Adaptive Transport Protocols. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Kobe, Japan, 1999.

[17] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple Description Wavelet Based Image Coding. *IEEE Trans. Image Proc.*, May 2000. To appear. Available from `http://lcavwww.epfl.ch/~servetto/`.

[18] S. D. Servetto and M. Vetterli. Video Multicast over Networks with a Stateless Core. Submitted for publication. Abstract available from `http://lcavwww.epfl.ch/~servetto/`.

[19] Special issue on Integrated and Differentiated Services for the Internet. IEEE Network, 13(5):8–71, 1999.

[20] Special issue on Internet Telephony. IEEE Network, 13(3):6–64, 1999.

[21] V. A. Vaishampayan. Design of Multiple Description Scalar Quantizers. *IEEE Trans. Inform. Theory*, 39(3):821–834, 1993.

[22] V. A. Vaishampayan and S. John. Balanced Interframe Multiple Description Video Compression. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Kobe, Japan, 1999.

[23] N. Venkitaraman, T.-E. Kim, K.-W. Lee, S. Lu, and V. Bharghavan. Design and Evaluation of a Suite of Congestion Control Algorithms for the Future Internet. In *Proc. ACM SIGMETRICS*, 1999.

[24] Y. Wang, M. T. Orchard, and A. Reibman. Multiple Description Image Coding for Noisy Channels by Pairing Transform Coefficients. In *Proc. 1st Workshop IEEE Signal Proc. Soc. on Multimedia Signal Proc.*, Princeton, NJ, 1997.