LIVE VIDEO STREAMING OVER PACKET NETWORKS AND WIRELESS CHANNELS

Vladimir Stanković, Raouf Hamzaoui

University of Konstanz
Department of Computer and Information Science
stankovi,hamzaoui@inf.uni-konstanz.de

Zixiang Xiong

Texas A &M University
College Station, TX 77843
zx@lena.tamu.edu

ABSTRACT

The transmission of live video over noisy channels requires very low end-to-end delay. Although automatic repeat request ensures lossless transmission, its usefulness to live video streaming is restricted to short connections because of the unbounded retransmission latency. An alternative is to use forward error correction (FEC). Since finding an optimal error protection strategy can be time expensive, FEC systems are commonly designed for the worst case condition of the channel, which limits the end-to-end performance. We study the suitability of two scalable FEC-based systems to the transmission of live video over packet networks. The first one uses Reed-Solomon codes and is appropriate for the Internet. The second one uses a product channel code and is appropriate for wireless channels. We show how fast and robust transmission can be achieved by exploiting a parametric model for the distortion-rate curve of the source coder and by using fast joint source-channel allocation algorithms. Experimental results for the 3D set partitioning in hierarchical tree video coder show that the systems have good reconstruction quality even in severe channel conditions. Finally, we compare the performance of the systems to the state-of-the-art for video transmission over the Internet.

1. INTRODUCTION

In video streaming, the encoded bitstream is not fully downloaded, but played out while parts of it are still being received. In some streaming applications, for example, video on demand, the video content is encoded and stored offline on a media server. Other applications, including broadcasting, traffic control, and video surveillance require online encoding of a live sequence. The delay tolerance of video on demand is on the order of 10 s. Thus, many packets lost or corrupted during transmission can be retransmitted, making automatic repeat request (ARQ) well-suited for error control. In contrast, the delay tolerance of live video streaming is much lower and ARQ may be inappropriate. For example, Wah and Su [1] report that the average round trip delay for the delivery of 64 TCP packets over the Internet from a site in Urbana to two different remote servers, one in Berkeley (low-loss connection) and one in China (high-loss connection) was 0.8 s and 149.37 s, respectively. Due to this limitation, current real-time interactive communication systems commonly use error-resilience and concealment techniques [2, 3]. However, such methods are vulnerable in highly noisy channels.

An alternative to ARQ and error-resilience techniques is forward error correction (FEC). For UDP transmission of the same number of packets over the same two connections as above, the average delay was 0.17 s and 0.81 s, respectively. Thus, UDP transfer, which may include FEC, was 10 to 100 times faster than the ARQ-based TCP transmission.

One problem with FEC is that for optimal rate-distortion results the error protection strategy has to be adapted to both the operational distortion-rate function of the source coder and to the channel condition, which are both changing in time. The problem of varying channel statistics is usually solved by using a protection that is optimal for the worst possible case. However, this often leads to overprotection and significantly reduces the performance when the channel noise is low.

This paper studies the feasibility of two systems based on scalable video coding and forward error correction [4, 5] for live video streaming over the Internet and third generation (3G) mobile networks. We answer the question of how to quickly compute an efficient error protection solution and show how to adapt the solution to varying source and channel characteristics.

The rest of the paper is organized as follows. In Section 2, we describe the proposed online streaming systems. In particular, we provide a fast algorithm that selects the length of the channel packet together with the corresponding unequal error protection. In Section 3, we present experimental results for packet erasure and Rayleigh fading channels and compare the performance of the systems to that of the state-of-the-art systems of [3] and [6].

2. LIVE VIDEO STREAMING

The used video systems consist of a scalable source coder, an error protection coder, and a logic for rate allocation with control devices. A brief description of each part follows.

2.1. Source coding

For source coding, we use a scalable source coder, for example, the 3D set partitioning in hierarchical trees (SPIHT) compression algorithm [7], MPEG-4 with the fine granularity scalability (FGS) mode [8], or H.26L with the progressive FGS mode [9]. In this paper, we focus on 3D

SPIHT, which is a fast and efficient 3D wavelet-based video coder that generates an embedded bitstream. It outperforms the MPEG-2 coder and provides performance comparable to that of H.263 when operating at bit rates between 30 to 60 kilo bits per second (kbps) [7]. In addition to being rate scalable, 3D SPIHT offers multiresolutional scalability in encoding and decoding in both time and space, precise rate constant bit rate traffic, and low complexity for possible software video applications. To keep the complexity of the algorithm low, we do not use the motion compensation mode of 3D SPIHT.

3D SPIHT partitions the original video sequence into groups of frames (GOF). Each GOF contains a fixed number of frames and is coded independently of the other GOFs. In all experiments, we used 16 frames per GOF to obtain an acceptable tradeoff between the encoding efficiency and the time spent for error protection.

2.2. Channel models

We are mainly interested in transmission over the Internet and 3G mobile networks. During data transmission over the Internet, packets are lost because they arrive at the destination with an unacceptable delay or due to buffer overflowing in routers. Since packet erasures usually appear in bursts, we use a two-state Markov process to simulate this behavior. The channel is fully described by the mean packet loss rate and the packet loss burst length [10]. However, over relatively short periods of time, packet loss probabilities are approximately independent [11]. Therefore, we also use a random packet erasure channel and model the packet loss probability function as an exponentially decreasing function. For wireless transmission, we use a Rayleigh flat-fading channel model.

2.3. Channel coding

For the packet erasure channel, we use systematic Reed-Solomon (RS) codes of maximal distance [12]. However, the RS codes can be replaced by any other systematic erasure correction codes, for example those of [13]. An (N, k) RS code can recover all information data if the number of lost symbols is less than or equal to $N - k$ and the positions of the lost symbols are known. To send N packets of L symbols each, we build L segments $S_1, \ldots, S_L$ consisting of $m_i \in \{1, \ldots, N\}$ source symbols (e.g., bytes) and protect each segment S_i with an (N, m_i) systematic RS code. Then, if n packets of N are lost, the RS ensures that all segments that contain at most $N - n$ source symbols can be recovered. By adding the constraint $m_1 \leq m_2 \leq \cdots \leq m_L$, if at most $N - m_i$ packets are lost, then the receiver can recover at least the first i segments. The system provides excellent protection against packet erasures and offers the desirable property that the reconstruction quality gracefully degrades as the packet loss rate increases [4]. The system can be extended to fading channels by additionally protecting each row with a concatenation of a cyclic redundancy check (CRC) code and a rate-compatible punctured convolutional (RCPC) code [5]. The source symbols are first encoded with RS codes column by column. Then, CRC detection bits are added to each row. Finally, each row is protected with the same RCPC code rate.

1	2	3	6	9	13	17
x	x	4	7	10	14	18
x	x	5	8	11	15	19
x	x	x	x	12	16	20
x	x	x	x	x	x	21

Table 1. BOP for an RS-based system. There are $N = 5$ packets of $L = 7$ symbols each. x denotes an RS redundant symbol, and cells designated by numbers are information symbols, where 1 is the most important and 21 is the least important symbol.

1	2	3	6	9	13	17	+	+	o	o	o
x	x	4	7	10	14	18	+	+	o	o	o
x	x	5	8	11	15	19	+	+	o	o	o
x	x	x	x	12	16	20	+	+	o	o	o
x	x	x	x	x	x	21	+	+	o	o	o

Table 2. BOP for a product code protection. There are $N = 5$ packets of $L = 12$ symbols each. x denotes an RS redundant symbol, + a CRC redundant symbol, o an RCPC redundant symbol. Cells designated by numbers are information symbols, where 1 is the most important and 21 is the least important symbol. RCPC need not be systematic.

Both systems generate a group of N packets of L symbols each called *block of packets* (BOP). Tables 1 and 2 show a BOP for a pure RS protection and for a product code protection, respectively.

2.4. Unequal error protection

An optimal joint source-channel rate allocation must exploit both the channel statistics and the operational distortion-rate curve of the source coder. Since the channel conditions are changing with time, optimization must be done online. For the Internet, we use passive network monitoring where the decoder computes the channel packet loss rate according to the number of received packets and sends this information back to the transmitter. Using this information, the transmitter adjusts the protection. The same approach can be used for a wireless network.

2.4.1. Estimation of the operational distortion-rate curve

Because of the online constraint, the operational distortion-rate points of the source coder must be quickly computed. One solution is to estimate the distortion at all rates during the encoding process without actual decoding (see [14] for 2D-SPIHT). This method is very fast, but since it can be realized only at the encoder side, the protection solution must be strongly protected and sent to the decoder. An alternative is to use a model of the distortion-rate function. Several models were proposed for the distortion-rate function of wavelet-based coders. The most appropriate one for

real-time applications is the four-parameter Weibull model of [15]. To determine the four parameters of the model, we need four distortion-rate points. We calculate these points with one encoding at the highest rate and four decodings at lower rates.

2.4.2. Local search

To protect the source code, we use the algorithms of [16] and [17] for the RS-based system and the product code-based system, respectively. The algorithms are based on the observation that a solution that maximizes the expected number of received source bits (rate-optimal solution) uses fewer protection bits than a solution that minimizes the expected distortion (distortion-optimal solution). In contrast to a distortion-optimal solution, a rate-optimal one is straightforward to find, and it is independent of the source. Our algorithms start from a rate-optimal solution and iteratively improve this solution by increasing the number of protection symbols. Experimental results for SPIHT, JPEG2000 [18], and 3D SPIHT show that our algorithms converge very fast and provide close to optimal average peak signal-to-noise ratio (PSNR) in both packet erasure and Rayleigh flat-fading channels.

2.4.3. Packet length selection

The performance of the source-channel bit allocation algorithms [16, 17, 19, 20, 21] depends on the BOP packet length L and the number of packets N. Increasing N increases the correction capabilities of the RS code, but also increases the redundant header information. For the system based on the product code, increasing L increases the number of bit errors in a packet and, consequently, reduces the RCPC protection capability. Since today's networks can handle different channel packet lengths, we allow the packet length in different BOPs to be different. However, all packets inside one BOP have the same length. We propose the following algorithm to determine the BOP packet length and the source-channel bit allocation.

Algorithm 1 *Let R_T be the target number of sent symbols. Let $\mathcal{L} = \{(L_1, N_1), \ldots, (L_k, N_k)\}$ be such that for all i, $1 \leq i \leq k$, L_i and N_i are positive integers with $L_i N_i \leq t$, where t is the largest non-prime number smaller than or equal to R_T.*
1. Compute a rate-optimal solution for each pair $(L_i, N_i) \in \mathcal{L}$ and sort the L_is in set $\mathcal{L}$ such that $E_{L_i}[r] \geq E_{L_j}[r]$ for $i < j$, where $E_{L_i}[r]$ is the expected number of correctly decoded source symbols of a rate-optimal solution when L_i ($1 \leq i \leq k$) is used. Set $j = 1$.
2. Find a protection using the local search of Section 2.4.2 with L_j as a packet length and N_j as the number of packets. Let $E_{L_j}[d]$ be the expected distortion of such a solution.
3. If $j > 1$ and $E_{L_j}[d] > E_{L_{j-1}}[d]$, then use the solution obtained with L_{j-1} and stop.
4. If $j = k$, then use the solution obtained with L_k and stop.
5. Set $j = j + 1$ and go to Step 2.

The variables N_i and L_i are not arbitrary. When the RS symbols are bytes, N_i should be smaller than or equal to 255. Also to limit the running time of the local search algorithm, the BOP packet length is also bounded.

In the worst case, Algorithm 1 uses the local search algorithm k times. This may be time prohibitive for some applications. Therefore, we propose to stop the algorithm after Step 2. In Section 3, we show that this strategy is reasonable.

2.5. Systems overview and complexity

Using the channel statistics, the systems first compute a rate-optimal protection. The video coder encodes the first 16 frames (one GOF) of the input video sequence at the source rate given by a rate-optimal solution (this is the highest achievable source rate). Four decodings are done at four lower rates, and the Weibull model parameters are calculated. Using the distortion-rate model, the rate-optimal solution, and the channel statistics, the method explained in Section 2.4.3 is used to find the number of packets and the packet length in a BOP, together with a new source-channel allocation. Then the channel codewords for the BOP are computed. Channel encoding is followed by the packetization step, in which channel packets are formed by adding headers to the BOP packets. The header includes one byte that indicates the number of the packet. With each BOP, we also send a header packet that contains the number of the BOP and the model parameters (four floating point numbers).

Note that the system can change the transmission rate for each BOP according to the network bandwidth fluctuation.

At the receiver, a buffer that stores the packets of a BOP is needed. The size of the buffer must be at least $N_{max} - 1$ packets, where N_{max} is the greatest possible number of packets in a BOP. Using the model parameters, the receiver can determine the same unequal error protection solution as the transmitter.

We now look at the time complexity of the systems. For each GOF, the systems need one source encoding, five source decodings (four by the sender and one by the receiver), one Weibull modeling, two runs of the optimization algorithm (one at the encoder and one at the decoder side), one channel encoding and one channel decoding. Therefore, the total end-to-end delay is

$$\begin{aligned} T_{total} &= T_{cap} + T_{enc} + 5T_{dec} + T_{fenc} + T_{fdec} + \\ &\quad T_{model} + 2T_{ls} + T_{net} + T_{buf}, \end{aligned}$$

where T_{cap} is the time needed to capture a GOF, T_{enc} and T_{dec} are the time of the source encoding and decoding, respectively, T_{fenc} and T_{fdec} are the time needed for FEC encoding and decoding, respectively, T_{model} is the time used to compute the parameters of the Weibull model, T_{ls} is the running time of the optimization algorithm, T_{net} is the network propagation delay, and T_{buf} is the delay caused by the receiver buffer.

To further decrease the time complexity, we propose to use the same source-channel bit allocation for several successive BOPs. Thus, modeling the distortion-rate curve and computing the error protection solution will be repeated

Trans. rate (kbps)	True function	Weibull model
56	172.65	172.85
100	112.75	113.54
128	91.85	91.76
200	60.77	60.67

Table 3. Expected Y-MSE at various transmission rates for the first 16 frames of the QCIF Foreman sequence. Packet loss protection in a packet erasure channel with a 10% mean loss rate is optimized with the true distortion-rate function and with the Weibull model.

only after a given number of BOPs. This reduces the total delay for one GOF to

$$T_{total} = T_{cap} + T_{enc} + T_{dec} + T_{fenc} + T_{fdec} + T_{net} + T_{buf}.$$

3. RESULTS

We present results for two standard 30 frames per second (fps) YUV 4:2:0 QCIF (176×144) video sequences: Foreman and Carphone. Unless otherwise stated, the frame rate was 10 fps, obtained by coding every third frame. The original sequences were divided into six GOFs of 16 frames each. The size of the header of the channel packet was 41 bytes (20 bytes for IP, 8 bytes for UDP, 12 bytes for RTP, 1 byte for the packet number) for the RS-based system and one byte (packet number) for the product code system. The maximum size of a channel packet was 1000 bytes. We consider transmission rates up to 200 kbps.

We first study the time complexity of the systems. The CPU time was measured on a PC having an AMD Athlon XP 1600 1400 MHz processor and a main memory size of 1 Gbyte. The time needed to capture the 16 frames of a GOF is $T_{cap} = 1.59$ s, T_{enc} was always less than 0.3 s. T_{dec} was less than 0.08 s, T_{fenc} and T_{fdec} were always less than 0.05 s, T_{model} was less than 0.01 s, T_{ls} was always less than 0.11 s. Finally, we assume that T_{net} is less than 0.2 s and that T_{buf} is less than 0.05 s. Thus, the overall end-to-end delay was less 2.87 s for the first GOF and less than 2.32 s for each successive frame. This is too much for video conferencing and telephony, but acceptable for many other applications such as live broadcasting and surveillance.

Table 3 shows the performance of the RS-based system when the joint source-channel allocation is based on the Weibull model instead of the true distortion-rate function. The table gives the average expected mean square error of the luminance component (Y-MSE) for the first GOF of the Foreman sequence. The expected Y-MSE is computed for the true distortion-rate points. The table shows that the Weibull model provides similar performance to the true distortion-rate function.

Figure 1 illustrates the importance of a good joint source-channel allocation for the RS-based system. It shows the average of the expected Y-MSE as a function of the channel packet length for the first GOF of the Foreman sequence. The packets were transmitted over a packet erasure channel with a mean packet loss rate of 10%. The total transmission

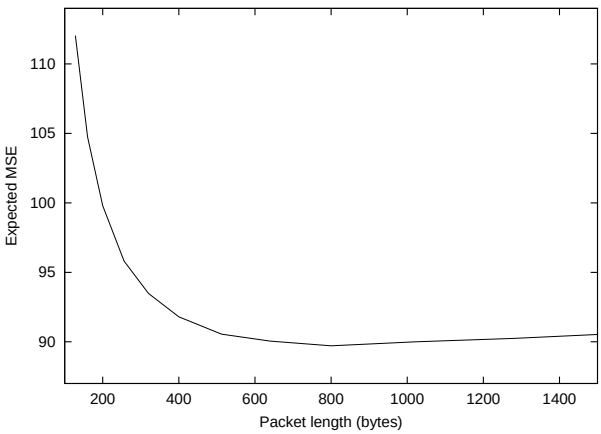

Fig. 1. Average of the expected Y-MSE as a function of the channel packet length for the first GOF of the QCIF Foreman sequence. Packets were transmitted over a packet erasure channel with mean packet loss rate 10%. The transmission rate is 128 kbps.

Channel packet length (bytes)	MSE
80	171.18
100	168.32
160	168.79
320	176.57
640	182.34

Table 4. Average of the expected Y-MSE as a function of the channel packet size for the first GOF of the QCIF Foreman sequence. The channel is a Rayleigh fading channel with average signal-to-noise ratio 10 dB and normalized Doppler spread $f_D = 10^{-5}$. The transmission rate is 64 kbps.

rate was 128 kbps. The best result was obtained for a channel packet length of 800 bytes. The experiment shows that the choice of the packet length has a significant influence on the reconstruction quality.

Table 4 shows a similar experiment for the product code system.

The following experiment shows that stopping Algorithm 1 after Step 2 causes only a negligible loss in quality. Table 5 compares the expected Y-MSE for a brute force solution, the solution computed by Algorithm 1, and the solution found after Step 2 of the algorithm. Packets were sent over a packet erasure channel with a mean packet loss rate of 10%. We obtained similar results for other channel conditions and other video sequences.

Figure 2 and Table 6 show that the protection is not sensitive to a change in the contents of the sequence when 3D-SPIHT is used a source coder. Figure 2 gives for each test sequence the PSNR of the expected Y-MSE when the protection was determined for the first GOF and used for all six GOFs. Table 6 shows that only a small drop in quality is observed when the protection computed for the first GOF of Foreman is applied to the first GOF of Carphone. Similar results were obtained for other channel conditions and

Trans. rate (kbps)	Step 2	Algorithm 1	Brute force
56	164.20	164.20	164.18
100	108.18	108.17	108.17
128	89.71	89.71	89.71
200	59.37	59.37	59.37

Table 5. Average of the expected Y-MSE at five different transmission rates for the first GOF of Foreman. The channel is a packet erasure channel with mean loss rate 10%. Step 2 is the solution found after Step 2 of Algorithm 1, and Brute force is a solution found by applying the local search algorithm to all pairs in $\mathcal{L}$.

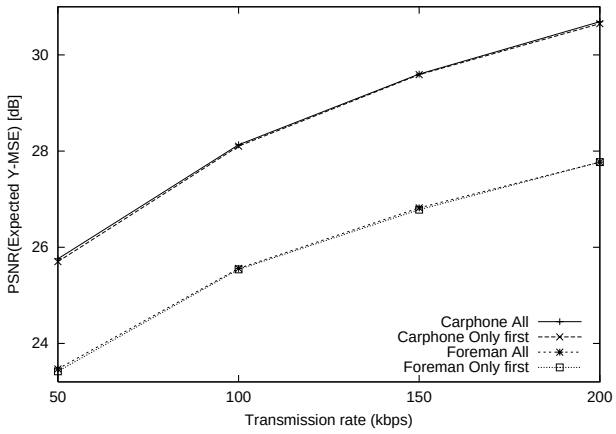

Fig. 2. PSNR of the average expected Y-MSE of the GOFs as a function of the transmission rate for Foreman and Carphone. The packet loss rate of the channel is 20%. Method *All* computes the protection for each GOF, whereas *Only first* computes the protection for the first GOF and applies the result to all GOFs.

video sequences.

Figure 3 shows the performance of the RS-based system at different packet loss rates when the loss protection strategy is optimized for the worst considered packet loss rate (20%) and when it is optimized for the actual packet loss rate. The experiment shows that optimizing for the worst case causes a significant drop in quality when the packet loss rate is very low.

The Internet can be well modeled as a two-state Markov channel [10]. Whereas it is easy to estimate the mean packet loss rate, estimation of the packet loss burst length is much harder. In the following experiment, we optimize the RS-based system for a random packet erasure channel and test it over a two-state Markov channel with the same mean packet loss rate. Results for target transmission rate 200 kbps are shown in Figure 4. The results are reported as PSNR of Y-MSE obtained by averaging over all frames and over 100 independent simulations. The algorithm selected channel packet length 1000 bytes. The figure shows graceful quality degradation with the increase of the mean packet loss rate. Figure 5 shows the average Y-PSNR value for each frame of the sequences.

To also cover streaming video for 3G mobile systems,

Trans. rate (kbps)	Car/Car	For/Car
56	73.67	76.06
100	42.66	43.17
128	32.26	32.76
200	19.01	19.38

Table 6. Average expected Y-MSE of the first GOF of the Carphone sequence. In the second column, the protection is determined for this GOF. In the third column, the protection is computed for the first GOF of Foreman. The channel packet loss rate is 0.1.

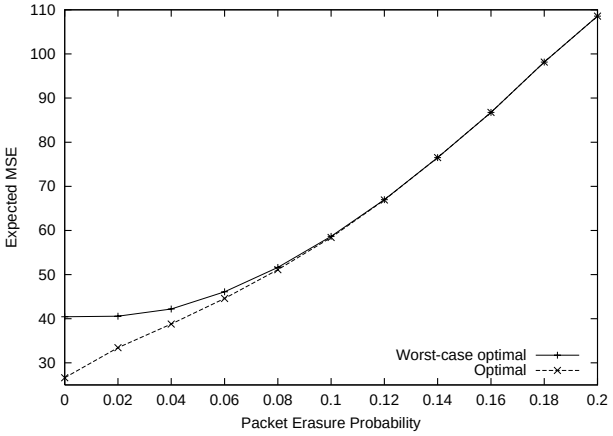

Fig. 3. Average expected Y-MSE at 200 kbps for Foreman.

we considered a Rayleigh flat-fading channel where the average signal-to-noise ratio (SNR) was 10 dB, and the normalized Doppler spread f_D was 10^{-5}. The channel was simulated with Jakes' [22] method. We used the communication system based on the product code. The row coder was a concatenation of a 16-bit CRC code with generator polynomial 0x15935 and an RCPC coder with generator polynomials (0117, 0127, 0155, 0171), mother code rate 1/4, and puncturing rate eight [23]. Thus, the set of RCPC rates was $\{8/32, \ldots, 8/10, 8/9\}$. The decoding was done with a list Viterbi algorithm where the maximum number of candidate paths was 100 [24]. Figure 6 gives the average Y-PSNR values for each encoded frame at transmission rate of 64 kbps. The algorithm selected channel packet length 160 bytes. The simulations show that the system provided efficient protection even in severe channel conditions.

We now compare the performance of the RS-based system to that of the state-of-the-art for a wireline best-effort IP environment. We compare the results of our system to those of [3] and [6]. Wenger [3] uses H.26L as a source coder. He tests various error resilient and concealment techniques to protect the information. In the experiments, packet losses were simulated using the same erasure pattern files obtained by actual simulations on Internet backbones [25]. Four error patterns with average packet loss probabilities of 3, 5, 10, and 20 $\%$ were used. The encoding frame rate was 7.5 fps. Instead of minimizing the expected MSE, we maximized the expected PSNR. Figure 7 compares our results to those of [3]. It shows the average Y-PSNR as a function

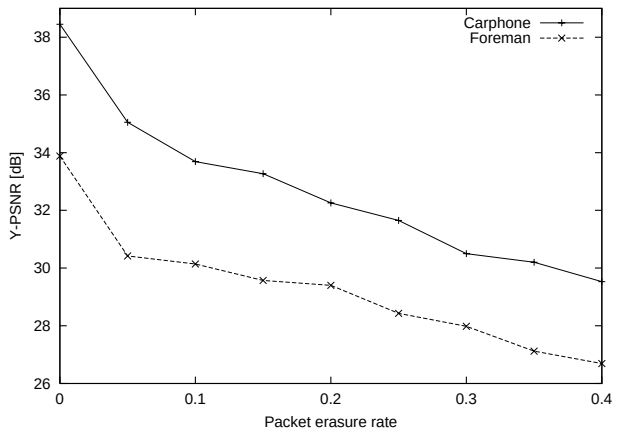

Fig. 4. PSNR of the average Y-MSE as a function of the mean packet loss rate. Results are given for a two-state Markov process with packet loss burst length five. The transmission rate is 200 kbps.

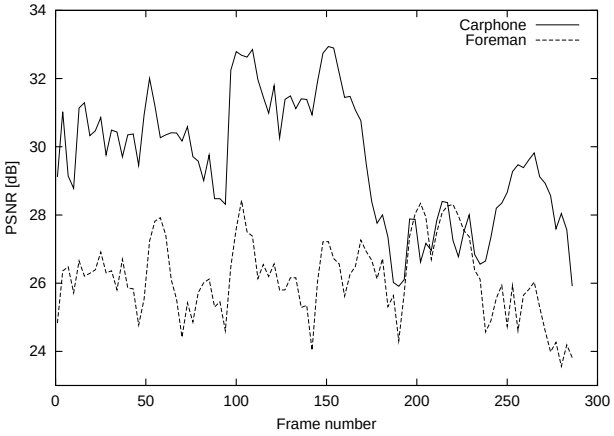

Fig. 6. Average Y-PSNR in dB for each frame of the two video sequences over a Rayleigh flat-fading channel with average SNR $= 10$ dB and $f_D = 10^{-5}$. The transmission rate is 64 kbps.

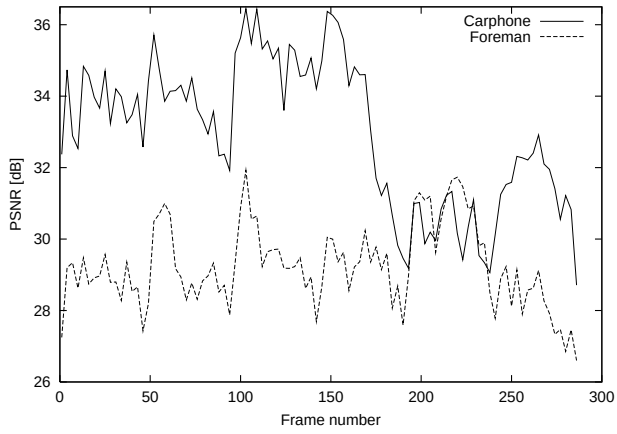

Fig. 5. Average Y-PSNR in dB for each frame of the two video sequences over a two-state Markov channel with mean packet loss 0.2 and packet loss burst length five. The transmission rate is 200 kbps.

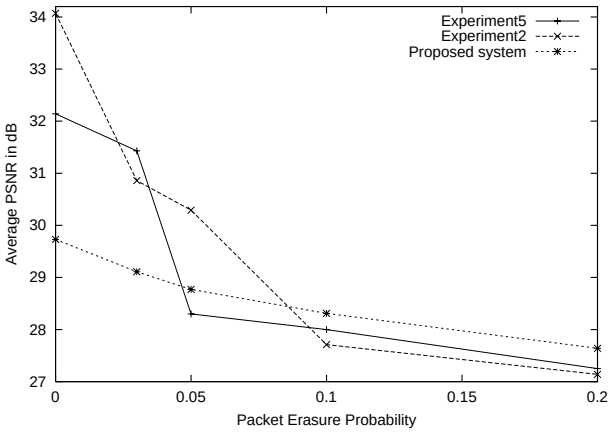

Fig. 7. Average Y-PSNR as a function of packet loss probability. Experiment 2 and Experiment 5 are the two best solutions of [3]. The video sequence is Foreman coded at 7.5 fps, and the transmission rate is 64 kbps.

of the mean packet loss probability at target transmission rate 64 kbps. We compare our results to the two experiments of [3] that provided the best performance: *Experiment 2* packetized each picture in one packet and used loss-aware rate-distortion optimization, whereas *Experiment 5* did not use rate-optimization, but exploited packet duplication. The system of [3] had a better performance than our system when the erasure rate was low. This is due to the fact that H.26L is superior to 3D SPIHT. However, due to the powerful FEC, the situation changed at high erasure rates. Note also that while we compute the PSNR for all encoded frames (every fourth frame of the original 30 fps sequence), Wenger [3] measures the PSNR only for the frames that were reconstructed (in whole or in part using error concealment). Therefore, the lost frames were not taken into account. It is observed in [3] that including them into the measure would cause significant quality loss for Experiment 2 and a very small loss for Experiment 5.

Figure 8 compares our results to those of [6] for an encoding frame rate of 7.5 fps. Stockhammer and Buchner [6] use the Progressive Texture Video Codec (PTVC) for source coding, which is a progressive video coder based on the H.26L test model. For channel coding, they use the RS codes of Section 2. They compute optimal unequal loss protection with dynamic programming, assuming that the distortion-rate curve is available. They fix the BOP packet size to 493 bytes. For both systems, the PSNR was computed for all frames of the original 30 fps sequence. This was done by comparing each reconstructed frame with four successive frames of the original sequence. The difference in reconstruction quality is mainly due to the superiority of PTVC over 3D-SPIHT. But since PTVC uses motion estimation with 1/4 pixel accuracy, it is not clear if the encoding can be done online.

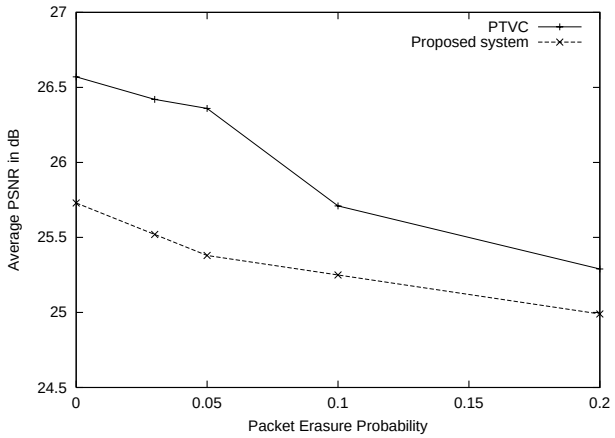

Fig. 8. Average Y-PSNR as a function of the packet loss probability for the Foreman sequence at 64 kbps.

4. CONCLUSION

We studied two video transmission systems that use a scalable source coder and forward error correction. We proposed a real-time algorithm for selecting the channel packet length together with an appropriate source-channel rate allocation. This solution was robust against changes in the sequence content. The systems have good end-to-end performance in packet erasure and wireless channels. Their low complexity makes them suitable to live video streaming applications. Finally, the scalabilty of the source code simplifies the problem of rate adaptation.

5. ACKNOWLEDGMENT

We thank Thomas Stockhammer for helpful comments.

6. REFERENCES

[1] B.W. Wah and X. Su, "Loss concealments of subband coded images for real-time transmission in the Internet," *Proc. ICME-2002 IEEE International Conference on Multimedia and Expo*, pp. 449–452, Lausanne, Aug. 2002.

[2] T. Stockhammer, T. Oelbaum, and T. Wiegand, "JVT/H.26L video transmission in 3G wireless environments," *International Conference on Third Generation (3G) Wireless & Beyond*, San Francisco, CA, May 2002.

[3] S. Wenger, "H.26L over IP: The IP-Network adaptation layer," *Proc. PV 2002* 12^{th} *International Packet Video Workshop,* Pittsburgh, PA, April 2002.

[4] A.E. Mohr, E.A. Riskin, and R.E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Comm.*, vol. 18, no. 7, pp. 819–828, Dec. 2000.

[5] D.G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated code," *Proc. SPIE Image and Video Communications and Processing*, vol. 3974, pp. 300–311, San Jose, CA, Jan. 2000.

[6] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," *Proc. PV 2001* 11^{th} *International Packet Video Workshop,* Kyongju, May 2001.

[7] B. Kim, Z. Xiong, and W.A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 692–695, Dec. 2000.

[8] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 11, pp. 301–317, March 2001.

[9] Y. He, F. Wu, S. Li, Y. Zhong, and S. Yang, "H.26L-based fine granularity scalable video coding," *Proc. ISCAS-2002 IEEE International Symposium on Circuits and Systems*, Scottsdale, AZ, May 2002.

[10] U. Horn, K. Stuhlemüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable Coding and unequal error protection," *Signal. Proc.: Image Communication*, vol. 15, pp. 77–94, Sept. 1999.

[11] P.A. Chou, A. Mohr, A. Wang, and S. Mehrota, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. on Multimedia*, vol. 3, pp. 108–122, Mar. 2001.

[12] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, April 1997.

[13] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, and V. Stemann, "Practical loss-resilient codes," *Proc.* 29^{th} *Annual ACM Symposium on the Theory of Computing*, El Paso, TX, May 1997.

[14] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[15] Y. Charfi, R. Hamzaoui, and D. Saupe, "Model-based real-time progressive transmission of images over noisy channels," *Proc. WCNC-2003 IEEE Wireless Communications and Networking Conference,* New Orleans, LA, March 2003.

[16] V. Stanković, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," *Proc. ICIP-2002 IEEE International Conference on Image Processing*, vol. 2, pp. 165–168, Rochester, NY, Sept. 2002.

[17] V. Stanković, R. Hamzaoui, and Z. Xiong, "Joint product code optimization for scalable multimedia transmission over wireless channels," *Proc. ICME-2002 IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 865–868, Lausanne, Aug. 2002.

[18] D.S. Taubman and M. Marcellin, "Jpeg2000: Image Compression Fundamentals, Standard, and Practice," Kluwer, 2001.

[19] R. Puri and K. Ramchandran, "Multiple description coding using forward error correction codes," *Proc. 33rd Asilomar Conference on Signal, Systems, and Computers*, vol. 1, pp. 342–346, Pacific Grove, CA, Oct. 1999.

[20] A.E. Mohr, R.E. Ladner, and E.A. Riskin, "Approximately optimal assignment for unequal loss protection," *Proc. IEEE ICIP-2000 International Conference on Image Processing*, vol. 1, pp. 367–370, Vancouver, Sept. 2000.

[21] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protection packetization of scalable code streams," *Proc. DCC'02 Data Compression Conference*, J. A. Storer and M. Cohn (Eds.), IEEE Computer Society Press, pp. 73-82, Snowbird, Utah, April 2002.

[22] W.C. Jakes, "Microwave Mobile Communications," *Wiley*, 1974.

[23] P. Frenger, P. Orten, T. Ottosson, and A. Svensson, "Multi-rate Convolutional Codes," Technical report R021/1998, Göteborg, 1998.

[24] M. Röder, R. Hamzaoui, "Fast list Viterbi decoding and application for source-channel coding of images," *ICME-2002 IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 801–804, Lausanne, Aug. 2002.

[25] S. Wenger, "Internet Error Patterns," VCEG-N16r1.zip, available from `ftp://standard.pictel.com/video-site/9910_Red/q15i16r1.zip`, Oct. 1999.