# Multiple Description Coding versus Transport Layer FEC for Resilient Video Transmission


C. Soldani, G. Leduc
*University of Liège*
*Cyril.Soldani@ulg.ac.be*

F. Verdicchio, A. Munteanu
*Vrije Universiteit Brussel*
*fverdicc@etro.vub.ac.be*



## Abstract

*Video content delivery is a challenging task due to its large bandwidth requirements and its sensitivity to transmission errors. In this context, simultaneously providing scalability and resilience against transmission errors is of paramount importance. Layered video coding coupled with multicast video transmission employing no feedback provides the required scalability features and reduces the network burden. However, it is not easy to provide reliability in such environments. This is crucial to mobile users as error rates on wireless links can be high. To compensate for transmission errors, redundancy must be added to the video content. This can be added at the application layer by the employed video coding system, or can be added transparently by the transport layer.*

*In this paper, we compare the performance obtained with a scalable wavelet-based video codec that adds redundancy using multiple description coding with the equivalent system adding redundancy at the transport level, using forward error correction.*


## 1. Introduction

Delivery of video content over the Internet presents many challenges. Video traffic requires considerable bandwidth and, in real-time or streaming applications, it must also respect severe delay constraints. Furthermore, compressed videos are extremely sensitive to data losses. Providing reliability and scalability at the same time is thus an open challenge.

These problems are even more emphasised by a large heterogeneity amongst the receivers who may use different devices (computers, cell phones, PDAs, ...) with different displays and computing power. In this context, scalable or layered video coding is a desired feature in order to meet each user's requirement. Moreover, from the network perspective, channel capacities may also vary widely from lossy low-bandwidth wireless connections to high bandwidth wired connections.

Multicast alleviates the bandwidth requirement as it allows servers to stream videos to more than a few users at a time, without exhausting their network capacity. In multicast transmissions, the same stream sent by the server can be used by many clients. Each packet passes only once on the (overlay) links and many users share the same bandwidth. Packets are duplicated only as needed (when the paths to different receivers split) and resources usage is minimized.

The heterogeneity of receivers may be tackled by decomposing the video stream into several layers. A single layer provides a basic quality suitable for low capacity receivers. More powerful receivers can use additional layers to progressively refine the quality of the received video. Using multicast, each user can subscribe to a number of layers adapted to its capacity and network state.

However, to obtain a fully scalable transfer scheme, we must avoid any feedback from receivers to the source. This feedback traffic would else grow with the number of receivers and could exacerbate the network condition. Some schemes exist to bound this feedback (such as the TRee based ACKnowledgment, TRACK [1]), but they are complex and not widely deployed (multicast itself is far from being fully deployed in today's Internet). Moreover, some receivers may have an expensive return path or no return path at all (*e.g.* on satellite broadcasting links). Finally, the delay constraint posed by video streaming may be shorter than the round-trip time, thus rendering retransmission useless.

As transmission errors cannot be corrected by on demand retransmissions, the server should provide the users with a compressed stream which is resilient to packet losses. This becomes especially important today with the increasing use of mobility, because wireless links can have much higher error rates than their wired counterparts [2].

One possible solution is to add redundancy at the video coding stage. This is achieved by using multiple description coding (MDC) [3], which generates multiple complementary versions (descriptions) of the input video. Each description provides a representation of the source at an acceptable quality. Combining all the received descriptions, a more accurate reconstruction is obtained. It is also possible to smoothly increase the quality provided by each coded description, as more data is available from the stream. This approach is referred to as scalable multiple description video coding. As the video encoder knows which information contributes the most to the perceived quality, it can efficiently add redundancy by protecting critical information more strongly [4].

Another approach that can be used to add redundancy is the use of forward error correction (FEC). The compressed video stream is fragmented into blocks. Recovery blocks are computed upon these video blocks using general error correcting channel codes. The channel-coded blocks are then transmitted. Even though some blocks may be lost, the original video data can be completely recovered, provided that enough blocks have been received.

The scope of this paper is to perform a comparative analysis of the two basic approaches of adding redundancy to the video content (MDC at the application layer versus FEC at the transport layer) and to shed some light on their advantages and disadvantages. The remainder of this article is structured as follows. A brief overview of scalable multiple description video coding is given in section 2. The basic principles of forward error correction are briefly presented in section 3 while our our experimental methodology is presented in section 4. Comparison of the results of the different methods is discussed in section 5. Finally, section 6 draws the conclusion of our work.

## 2. Scalable Multiple Description Video Coding

The employed scalable video coding system [4] is composed of three parts. First, the input video stream is temporally and spatially decomposed via wavelet based motion compensated temporal filtering (MCTF). Then, MDC is applied on the resulting coefficients using embedded multiple description scalar quantizers (EMDSQ) [5, 6], introducing a fixed amount of redundancy. Finally a post-compression rate-distortion optimization is employed to adapt the amount of redundancy in the coded stream to the channel conditions.

### 2.1. Motion Compensated Temporal Filtering

Motion compensated temporal filtering removes the temporal redundancies in the video sequence by performing a wavelet transform along the motion trajectory.

In a first stage, the input video frames are filtered, resulting into $L^1$ frames (low frequency or average) and $H^1$ frames (high frequency or difference). This filtering is

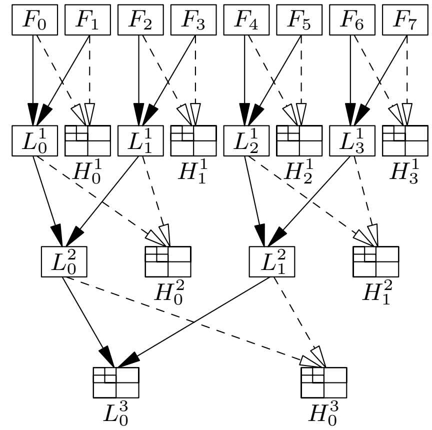


**Figure 1. MCTF pyramid. The top row represents successive input frames. Solid lines represent low-pass filtering and dashed ones represents high-pass filtering. Sent frames are encoded using a wavelet transform. Motion vectors are omitted for clarity.**

performed in the direction of motion. Assuming HAAR wavelets, MCTF can be written using the lifting formulation [7] as follows:

$$H_t^1(x, y) = \frac{1}{\sqrt{2}}(F_{2t+1}(x, y) - F_{2t}(x + \nu_x, y + \nu_y)),$$

$$L_t^1(x, y) = \sqrt{2}F_{2t}(x, y) + H_t^1(x - \nu_x, y - \nu_y)$$

where $(\nu_x, \nu_y)$ is the motion vector representing the motion of the input frame $F_{2t}$ with respect to the frame $F_{2t+1}$.

At a second filtering stage, $L^1$ frames are further decomposed into $L^2$ frames (average of average) and $H^2$ frames (difference of average). Iterating this process $T$ times produces a pyramidal decomposition as illustrated in figure 1 for $T = 3$.

The above temporal filtering yields the first form of scalability. In fact, the decoder readily obtains an estimation of the input sequence at (dyadically) reduced frame rate by halting the temporal synthesis without processing one or more lower temporal levels.

The second coding stage consists of applying a spatial wavelet transform to the lowest $L$ frames and $H$ frames to remove spacial redundancies. This spatial transformation brings a second level of scalability because the decoded frames may be reconstructed at (dyadically) reduced spatial resolutions by excluding one or more levels from the spacial synthesis.

The final coding stage consists in embedded quantization of the produced spatio-temporal sub-bands (and relative motion vectors) followed by embedded entropy coding.

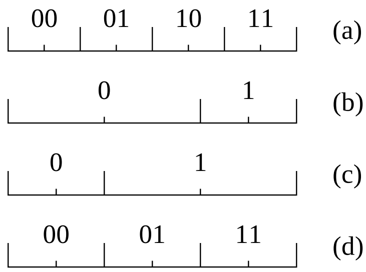


**Figure 2. Quantization example. (a) 2-bit quantization. (b) First 1-bit description. (c) Second 1-bit description. (d) Reconstruction from the two 1-bit descriptions.**

Therefore the accuracy of the reconstructed sub-bands and motion vectors is progressively refined as more data is received. This latter form of scalability is referred to as quality scalability

### 2.2. Multiple Description Coding with EMDSQ

A large number of MDC techniques exist in the literature. For an excellent survey, the reader is referred to [3]. This section focuses on the MDC scheme employed in this paper, which uses embedded multiple description scalar quantization (EMDSQ) to build quality-scalable multiple descriptions [5, 6].

The principle of multiple description scalar quantization is explained by means of the following example. Imagine you want to transmit, with finite accuracy, a value between 0 and 1. If you encode this value with two bits, you can partition the $[0, 1]$ domain into $2^2 = 4$ intervals as illustrated in figure 2 (a). The encoding then consists of mapping a value to the interval where it lies. At the decoder, the value is taken as being at the centroid of the corresponding uncertainty interval. Generally, some accuracy is lost in the process, due to the quantization error. In our example, the accuracy (maximum reconstruction error) is 1/8.

If you want to transmit the value on two descriptions which can mutually refine the accuracy on the decoded value, you can partition the quantization interval differently on each description. An example using two 1-bit descriptions is given in figure 2 (b) and (c). If only one description is received, the decoding process is the same as previously, the value is taken as the centroid of the indicated interval. If more descriptions are received, the interval where the value lies is the intersection of the individual intervals from each description. The accuracy improves as we receive more descriptions because the uncertainty interval where the original value lies becomes smaller.

In our example, if you receive 0 for the first description, then you know that the value lies between 0 and 2/3 and it is decoded as 1/3 with a maximum error of 1/3. Similarly, if you receive 1 for the second description only, you know that the value is between 1/3 and 1 and you decode it as 2/3 with a maximum error of 1/3. If you have received both bits, then you know that the value lies between 1/3 and 2/3. It is decoded as 1/2 with an accuracy of 1/6.

The advantage of using such quantizers for the two descriptions rather than sending high and low order bits directly is that the two generated descriptions are balanced, *i.e.* equally important, since they yield on average the same accuracy.

Clearly, the redundancy between the descriptions has a cost. We obtain an accuracy of 1/6 with our two 1-bit descriptions when the 2-bit representation had an accuracy of 1/8. Information received from the two descriptions is only $\log_2 3 = 1.58$ bits instead of two so we have lost more than 20% of the information in the process.

This toy example is meant to provide a general idea of the principles behind multiple description quantization. Without entering the details, the employed EMDSQ can be considered as the extension to embedded quantization of the above example. For more detailed information, the reader is referred to [5, 6].

Coupling the MCTF decomposition with EMDSQ, we obtain a fully scalable MDC video codec [4]. Multiple descriptions are created during the quantization stage, assuming a worst-case scenario and introducing a maximum amount of redundancy. Then, an optimized rate-redundancy allocation can be performed in order to reduce the amount of introduced redundancy whenever the channel is more reliable than the worst case. We mention here that such adaptation can be performed on the fly, without additional coding steps. For further details the reader is referred to [4].

## 3. Forward Error Correction

Forward Error Correction consists of adding redundant information to the transmitted data, which allows the reconstruction of some amount of missing data at the receiver.

The more frequently used FEC codes employed at the transport level are the so-called *erasure codes*. An $(n, k)$ erasure code encodes $k$ source packets into $n > k$ packets, which are then transmitted via an erasure channel. At the receiver, the original $k$ source packets can be recovered provided that a sufficient number of transmitted packets are received. Such a sufficient number of packets may be exactly $k$ (we then say it is an *optimal* code) but may be a little bit more than $k$ as in the case of Low Density Parity Check (LDPC) codes [8].

When the FEC code is *linear*, the encoding and decoding processes may be represented as matrix operations. Let $\mathbf{x} =$

$x_0 \ldots x_{k-1}$ be the source data and $G$ a $n \times k$ matrix. A $(n, k)$ linear erasure code may be represented by

$$\mathbf{y} = G\mathbf{x}$$

where $G$ is such that any sub-matrix made of $k$ rows from $G$ is invertible. $\mathbf{y}$ gives the data to transmit. If $\mathbf{y}$ contains a verbatim copy of all elements of $\mathbf{x}$, the code is said to be *systematic*. The advantage of a systematic code is that decoding is not needed if no source packet was lost. A simple way to achieve a systematic code in linear erasure codes is to include the identity matrix into $G$.

Assuming that $\mathbf{y}'$ made of any $k$ elements of $\mathbf{y}$ is available at the receiver, the original source data is recovered by solving the system of $k$ equations with $k$ unknowns given by

$$\mathbf{x} = G'^{-1}\mathbf{y}'$$

where $G'$ is the $k \times k$ matrix representing these equations ($G'$ is the sub-matrix of $G$ which has the same $k$ rows as $\mathbf{y}'$).

In our experiments, we have used the REED - SOLOMON code based on the implementation by [9]. This code, based on Galois fields (or finite fields), is linear and systematic. It can be implemented efficiently in software provided than $n$ is not too high (say lower than 256). So it is well adapted to protect multimedia content in interactive applications (where the blocks must be small due to delay constraints).

## 4. Experimental Methodology

The experiments reported in this paper have been conducted on the two classical YUV color sequences *bus* and *football*, featuring CIF resolution and 30 frames per second. The sequences are encoded using two instantiation of the MCTF-based wavelet codec. The first instantiation applies EMDSQ to produce two descriptions. The second codec is based on single description (SD) instantiation and introduces redundancy within the transmitted stream by means of FEC.

For MDC, different streams targeting the desired bit-rate are produced for various expected probability of loss. That is, for each case, the MDC encoder varies the amount of redundancy within the stream, matching the expected channel conditions.

For the FEC experiments, on top of the SD stream generated by the MCTF encoder, we varied the redundancy level by adding FEC packets to the video data. The SD stream bit-rate was decreased accordingly to compensate for the addition of FEC packets. So, in all experiments, the rate and amount of transmitted data were kept constant.

In both MDC and FEC-based systems, the redundancy is added considering groups of pictures (GoP) of 16 frames at a time. The efficiency of error correction is higher on larger blocks of data, as this allows for fighting the effect of burst of losses. However, if we use larger blocks, we must delay their transmission which is incompatible with real-time applications.

Encoded video streams were transmitted between two machines in our network testbed composed of a few computers and routers. We used network traffic generators to load the links and nodes on the path of the video streams in order to produce losses. These simulations were repeated several thousands times with the two error-protection methods and with different redundancy levels between 0 (no protection) and 30% of redundancy or expected loss rate.

Motion vectors have not been considered in our experiments. They were not transmitted and were made available at the receiver reliably. We point out that such assumption is reasonable since the motion vector rate constitutes a negligible fraction of the high bit-rate chosen for the simulations. As such, we can safely assume that a sufficient number of replicas are transmitted to ensure correct reception.

At the receiver, the streams were decoded and compared to the original sequences to compute the average PSNR of all the frames and the average PSNR of the worst three consecutive frames for the luminance (Y) component.

In the experimental section we focus on the transmission of the football sequence at 2048 Kbits per second (Kbps). Given the video content (football contains a lot of motion), we selected a bit-rate providing good visual quality and allowing us to safely neglect the bit-rate needed to transmit the motion vectors. Similar results were obtained for the bus sequence.

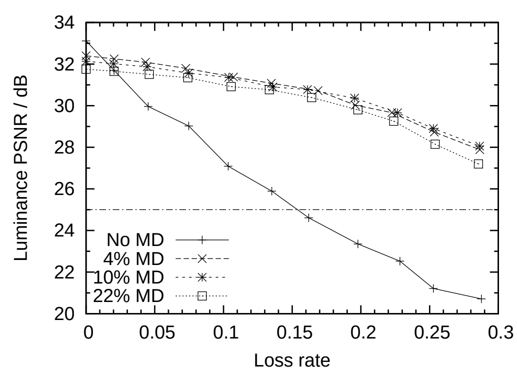


**Figure 3. Average luminance PSNR for MDC. We see MDC improves the quality in presence of losses with a smooth quality degradation.**

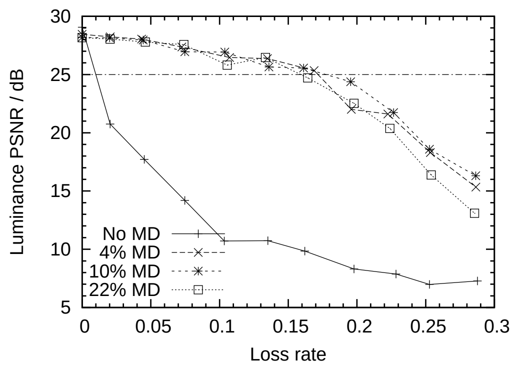


**Figure 4. Luminance PSNR of the worst three consecutive frames for MDC. We see a drastic improvement with MDC which allows a far higher loss rate than the SD stream.**

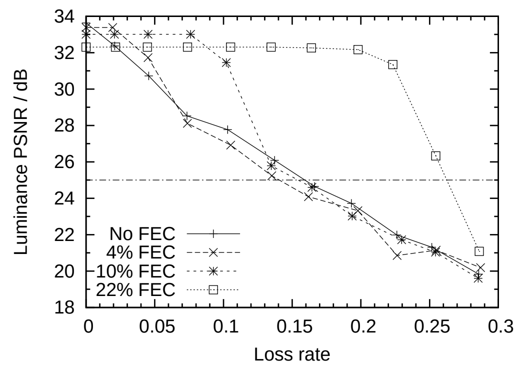


**Figure 5. Average luminance PSNR for FEC. We see the improvement brought by FEC is directly related to the redundancy level. The higher the redundancy level, the better the improvement but at the cost of a decrease in quality for low-loss receivers.**

## 5. Experimental Results

### 5.1. Multiple Description Coding

Figure 3 shows the average luminance PSNR *vs* loss rate for several redundancy levels. The 25 dB line corresponds to the minimum acceptable quality as subjectively evaluated by a human viewer. Results have been aggregated in 3% bins which corresponds on average to 60 simulations per bin and per curve.

The results show that although the use of MDC may imply a loss of 1 dB in the no loss case, it drastically improves the video quality when losses do occur. The improvement is even more important when we consider the average luminance PSNR of the worst three consecutive frames of the sequence. This information is interesting because a defect in the image becomes noticeable if it last across three consecutive frames. We see in figure 4 that the MDC coded stream allows for as much as 15% losses with an acceptable quality while the SD coded stream becomes heavily corrupted with as little as 1% losses.

We must mention that the decoder does not try to estimate missing information using prediction (in other words, it does not perform error concealment). Missing blocks are treated as zero-valued samples which causes black or colored rectangles to appear. This allows us to estimate the MDC protection alone rather than the combined effect of MDC and error concealment techniques.

We notice that the redundancy level does not seem to influence the output very much. Once the transmitted stream switches from SD coded to MD coded, we obtain an significant improvement in error resiliency. However, the curves describing the behavior of the streams targeting non-zero redundancies almost overlap. This is due to the fact that the encoder we used generates at most two descriptions. When a maximum of only two descriptions is generated, and when high bit-rates are targeted, the MDC streams generated for various non-zero redundancy levels tend to be very similar.

The non-monotonic behavior occasionally observed for the PSNR curve is due to the sensitivity of the MD codec to the loss pattern. A particular loss pattern can have a more devastating effect than a different pattern featuring the same amount of losses (or even more losses).

### 5.2. Forward Error Correction

FEC results for both average and worst three consecutive frames PSNR are shown in figures 5 and 6 respectively. We observe again a great improvement in comparison to the non protected stream. This seems to indicate that in heterogeneous networks, bandwidth is better allocated by addition of redundancy than by increasing the video bit-rate. We must however mitigate this affirmation because we used a high bit-rate. The results might be very different for lower bit-rates.

Contrary to the MDC case, the redundancy level has a more visible impact on the performance. Small level has little impact (2% FEC gives nearly no improvement) due to block effect (burst losses) while high redundancy levels allow for a very big loss rate (up to the redundancy level) without compromising the minimum quality. However, high redundancy levels penalize receivers experiencing few losses because they receive redundant data, which unnecessarily consumes bandwidth. The redundancy level

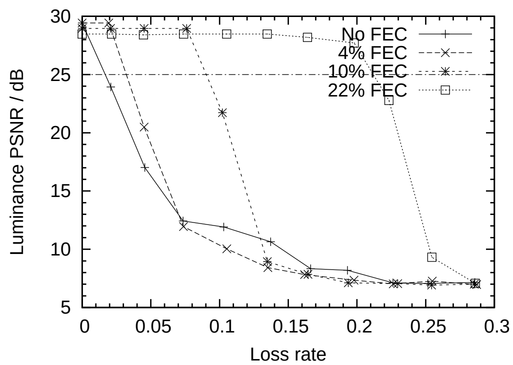


**Figure 6. Luminance PSNR of the worst three consecutive frames for FEC. We see that FEC brings a large improvement but quality still degrades sharply if the loss rate exceeds the redundancy level.**

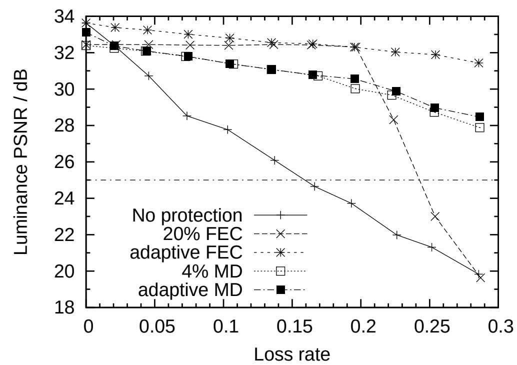


**Figure 7. Comparison of MDC and FEC for average luminance PSNR. FEC gives a better quality for low-loss and middle range receivers but MDC gives better results for high-loss receivers, when the loss rate overrun the redundancy level.**

is then more difficult to choose with the FEC method. The loss in quality when we overrun the redundancy level is also sharper than with MDC.

This sharp decrease in quality when we overrun the redundancy level is due to the all-or-nothing nature of FEC. Contrary to MDC where all the redundancy can be used to refine the quality, FEC information may not be useful. If there are few losses, most of the FEC packets are discarded because corresponding source blocks have been received. These FEC packets are purely redundant and do not bring any information if all source packets have been received. On the other hand, if too many packets are lost, FEC packets cannot be used for recovery (at least with RS codes) and then have a negative impact of performance (bandwidth is consumed without improving the decoded quality).

### 5.3. FEC or MDC?

Comparison plots for MDC and FEC are given in figure 7. We plotted curves for both fixed and adaptive redundancy levels. We choose a redundancy level of 20% for the fixed case for FEC and 4% for MDC (as we saw in section 5.1, the quality increase at high loss rates does not compensate for the loss of quality at low loss rate motivating the choice of a lower redundancy level for MDC).

Adaptive curves are hypothetic curves that would result of a constant adaptation of the redundancy level to the receiver's conditions optimizing the luminance PSNR. The curves are obtained by choosing the redundancy level which gives the higher average luminance PSNR at each loss rate.

The fixed case is the easier to deploy, as it concerns only the streaming server. The server adds the redundancy once for all and transmits the protected stream to all receivers. The more losses are expected, the higher the redundancy level.

For fixed rates, figure 7 shows that FEC generally performs better for low-loss and middle-range receivers while MDC gives better results for high-loss receivers. Difference in quality between the two schemes is rather small for users in the first part of the loss rate range (between 0% and 15%), but it increases when we approach or overrun the redundancy level (between 15% and 30%).

The adaptive case generally gives better results than the fixed one. However, it has the drawback that, for heterogeneous networks, transcoding at edge nodes may be necessary. In fact, the addition of redundancy must occur near the receivers. The video could be transmitted with high bit-rate and low redundancy by the server. On edge nodes where the loss rate increases (*e.g.* at the access point of a wireless network), the video can be transcoded by sacrificing bit-rate for redundancy. The video stream can also be augmented with redundancy, increasing the total bandwidth consumed by the video stream. Adding sufficient redundancy is the only option for a FEC-based system if we want full transparency and generality (which does not involve video transcoding), but FEC can also be used with transcoding.

## 6. Conclusions

The transmission of video in very large and heterogeneous networks can use multicast but should avoid any feedback from receivers to source as the feedback traffic grows with the number of receivers and can exacerbate the network condition. To alleviate or suppress the need of feedback, the video streams must be resilient to transmission errors. This resiliency can be obtained by adding redundancy to the transmitted video stream. This can be achieved at the application level using MDC or at the transport level using FEC.

The choice of MDC or FEC for resilient video transmission is not easy. It will largely depend of the goal you want to achieve and what you privilege most as both have advantages and drawbacks. MDC makes a good use of redundancy for low-loss receivers. It can be parametrized robustly and supports both transcoding (keeping the same bit-rate) and augmentation (addition of redundancy). FEC is a more general technique, especially if we restrict ourselves to addition of redundancy without the need to understand the underlying video encoding process. It gives good results as long as you do not overrun the loss rate corresponding to the redundancy level. However, it must be parametrized with care (maximum quality *vs* accepted loss rate range). Video quality degrades also more sharply with FEC than with MDC with which the transition is smoother.

Another key advantage of MDC over FEC is its inherent support for splitting the video stream over multiple layers for multicast streaming (stacking the different descriptions). As the descriptions may be decoded independently and may be of different sizes, it gives a very flexible and attractive scheme for multicast distribution. FEC can also be used to produce multiple descriptions [10] but this is less flexible than EMDSQ.

To conclude, MDC and FEC both have their advantages and drawbacks for resilient video transmission and the choice of one or the other must be driven by our own design constraints and priorities. Although MDC is in general less efficient than FEC in our experiments, it is still performing largely better than the unprotected stream while providing an unmatched level of flexibility for multicast distribution.

Future research directions could be to combine MDC and FEC in an hybrid scheme and to expand our analysis to the case of video distribution with multiple layers using multicast. It is possible that a cross-layer solution could provide flexibility and efficiency at the same time. Finally, we should extend our investigations to lower bit-rate video transmissions and study the impact of the loss pattern on the results.

## References


[1] B. Whetten, D. M. Chiu, M. Kadansky, S. J. Koh, and G. Taskale, "Reliable Multicast Transport Building Block for TRACK," IETF draft draft-ietf-rmt-bb-track-02.txt, November 2002, work in progress.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-Level Measurements from an 802.11b Mesh Network," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 121–132, October 2004.

[3] V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, September 2001.

[4] F. Verdicchio, A. Munteanu, A. Gavrilescu, J. Cornelis, and P. Schelkens, "Error-Resilient Video Coding Using Motion Compensated Temporal Filtering and Embedded Multiple Description Scalar Quantizers," in *IEEE Int. Conf. on Image Processing*, vol. 3, Italy, October 2005, pp. 933–935.

[5] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "Embedded Multiple Description Scalar Quantizers," *IEEE Electronics Letters*, vol. 39, no. 13, pp. 979–980, 2003.

[6] A. I. Gavrilescu, A. Munteanu, J. Cornelis, and P. Schelkens, "High-Redundancy Embedded Multiple Description Scalar Quantizers," in *IEEE Benelux Signal Processing Symposium*, The Nederlands, 2004, pp. 21–24.

[7] D. S. Turaga, M. van der Schaar, Y. Andreopoulos, A. Munteanu, and P. Schelkens, "Unconstrained Motion Compensated Temporal Filtering (UMCTF) for Efficient and Flexible Interframe Wavelet Video Coding," *Signal Processing: Image Communication*, vol. 20, no. 1, pp. 1–19, April 2005.

[8] C. Neuman and V. Roca, "Analysis of FEC Codes for Partially Reliable Media Broadcasting Schemes," in *International Workshop on Multimedia Interactive Protocols and Systems*, France, November 2004.

[9] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, April 1997.

[10] R. Puri, K. Ramchandran, K. W. Lee, and V. Bharghavan, "Application of FEC Based Multiple Description Coding to Internet Video Streaming and Multicast," *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 745–762, May 2001.