June 1996 LIDS-P-2345

Design of a Transport Coding Scheme for High-Quality Video over ATM Networks[†]

V.Parthasarathy, J.W.Modestino and K.S.Vastola

Electrical, Computer and Systems Engineering Department
and
Center for Image Processing Research
Rensselaer Polytechnic Institute
Troy, New York 12180

Abstract

In this paper, we explore the design of FEC-based error concealment schemes for digital video transmission on ATM networks. In particular, we study the impact of code selection on the overall performance and provide a judicious code selection strategy. The use of FEC provides an active and powerful means of recovery from packet loss which is particularly useful when the encoded video material has high motion and scene changes. The best technique for applying FEC is to throttle the source coding rate so that the overall transmission rate after FEC application equals the original unprotected rate. However, the resulting performance then depends on the particular code selected. A well-chosen code provides good protection while allowing little sacrifice in quality and at the same time satisfies specified delay constraints. Our results show that a single code is generally insufficient to provide good performance under all operating conditions. However, a small group of codes can be pre-selected, using the efficient code-selection strategy described here, which will provide efficient and robust performance over a wide range of channel conditions. We show that this simple code selection strategy is sufficient to select codes judiciously for a wide range of operating conditions and constraints. Employing this selection strategy, we demonstrate that moderate length codes are sufficient to provide good performance while meeting the imposed delay constraint.

[†] This work was supported in part by ARPA under Contract No. F30602-92-C-0030.

This work supported in part by DAAH04-95-1-0103 (Laboratory for Information and Decision Systems, Massachusetts Institute of Technology).

I. Introduction

With the development of high-speed integrated digital networks, there has been a significant increase in the number of new and improved services available to the user. Asynchronous Transfer Mode (ATM) has rapidly emerged as the standard for Broadband ISDN. Among the various services offered in future ATM networks, packetized, variable bit-rate (VBR) video is likely to be one of the largest users of bandwidth. However, real-time transmission of VBR video creates problems such as packet loss which, due to stringent delay constraints, cannot be recovered through retransmission as in most currently used protocols. In this regard, it would appear highly beneficial to use some form of *active* recovery scheme, such as forward error-control (FEC) coding, which offers the potential benefit of improved recovery in the event of packet loss and/or errors. These active schemes would then be applied in a hybrid fashion in conjunction with *passive* concealment techniques to provide a robust form of recovery. An example of a passive scheme would be using some form of temporal interpolation for estimating pixels belonging to the missing regions.

A judicious manner of applying FEC for VBR video transmission applications has been shown to be the technique where the video coding rate is reduced, or throttled, so that, after the application of FEC, the total transmission rate is identical to the video coding rate of the unthrottled system [9]. However, the performance of such schemes is directly related to the choice of the particular FEC code used. Furthermore, due to channel coding overheads, it is desirable to minimize the amount of quality traded for channel coding operation while at the same time maximizing the protection achieved. Poorly chosen codes not only waste transmission bandwidth, but also provide little protection under congestion.

In this paper, we study a particular code selection strategy and its impact on performance, and describe techniques to improve the overall efficiency of FEC-based transport schemes. These FEC-based techniques are intended for use in the ATM Adaptation Layer (AAL) [7],[8]. The purpose of this paper is then two-fold: to design suitable AAL interfaces as well as to study the impact of code selection on the performance of a hybrid error concealment scheme employing FEC. The problem in performing an overall optimization to relate the performance of a given class of codes to the decoded video quality is that the process is extremely complicated. This is due to the fact that analysis of packet loss in such systems, including error propagation effects, has so far proven to be intractable. Moreover, the reconstructed video quality also depends on the particular error concealment strategy used, the type of video being encoded, as well as the compression scheme employed, further complicating the analysis.

Some of the quantitative objectives of a well-chosen (N, K) code can be stated as the maximization of the rate $R = K/N$, minimization of FEC coding/decoding delays and the provisioning of adequate protection under packet loss. Note that the ratio $R = K/N$ determines the fraction of the overall rate allocated to the source coding operation. Therefore, one would like to operate at a code rate as close to 1 as possible in order to minimize the throttling of the source coding rate and thus maximize reconstructed video quality under light losses. Using the class of Reed-Solomon (RS) codes, we first formulate the process of code selection as a maximization of K/N while keeping the FEC coding delay and decoded packet loss below specified thresholds.

Although this approach is relatively simple, our results show that it is sufficient in yielding codes that can provide efficient and robust performance. Furthermore, as long as the code rates are properly chosen, codes of moderate code length N yield good performance. However, at low operating rates, the use of FEC is questionable due to the degree of additional compression required to accommodate the channel coding overheads. In general, a single code is insufficient to provide good performance for all channel conditions. However, as the optimized codes are relatively insensitive to slight variations in the channel conditions for which they were designed, we suggest pre-storing a selected group of codes in the adaptation layer, one each for the more frequently occurring channel conditions.

The use of FEC has been studied previously for compensating packet loss [1]-[5]. These papers describe the use of FEC without throttling the source coder rate to accommodate bandwidth expansion. Such a scheme is inefficient as it introduces additional congestion. In other related work, an AAL for constant bit-rate applications (specifically, the AAL 1 protocol) was proposed to incorporate FEC [7]. However, a valid rationale for the code selection does not seem to exist. Among these, the work by Zhang [4] which describes a code selection strategy in the case of an unthrottled source coder-FEC scheme for voice traffic is the closest to that in this paper. However, in [4], the delays introduced by the use of FEC have not been considered. Additionally, the process depends on modeling assumptions specific to voice traffic and is not tractable for more complex traffic models.

The organization of this paper is as follows. In Section II, we describe the coder used in this work. In Section III, we describe the application of FEC codes. Section IV discusses the code selection strategy and details of the performance evaluation. In Section V, we discuss the numerical results. Finally, in Section VI, we present a summary and conclusions.

II. Preliminaries

Beginning with a broad system framework, we describe in this section the coding details as well as some adaptations to suit network transport. Figure 1 provides a general block diagram of a video coding and prioritization scheme for transmission over a packet-switched network. Although the ideas presented in this paper would be applicable to arbitrary packet-switched networks, this paper focuses on ATM due to its emergence as the standard for supporting B-ISDN. This diagram is generic in the sense that it is applicable to any source coding scheme (e.g., a subband or a DCT-based system such as MPEG) or transport coding scheme (e.g., single or multiple priorities, with or without FEC). In our work, we use an entropy-constrained, subband-based coding scheme (the ECSBC scheme to be described) due to its excellent subjective quality and the fact that its multiresolution decomposition properties are well-suited to effective prioritization for heterogenous networking environments. Details specific to the coding scheme will follow later in this section.

The output of the video coder, in the form of parallel bit streams, enters the prioritization and transport coder. For example, in the case of a subband-based coding scheme these bit streams might result from coding different subbands. In a DCT-based scheme, they could result from entropy-coding the DC and AC coefficients. These output bit streams can then be packetized individually and classified into separate priority classes by the prioritization and transport coding block in the figure. The application of FEC would also be performed in this block. This block acts as an interface between the coder and the network. In B-ISDN, the ATM adaptation layer should perform the tasks associated with the prioritization (currently limited to two levels) and transport coder. Issues related to the function of each of these blocks will be discussed in detail in this section. Alternatively, the transport coding scheme proposed here could be performed at a higher layer if the reliability of the AAL's prove inadequate.

The allocation of priority levels can be performed in a hierarchical multiresolution manner to provide scalable video at different resolutions. As an example of the use of subband coding for

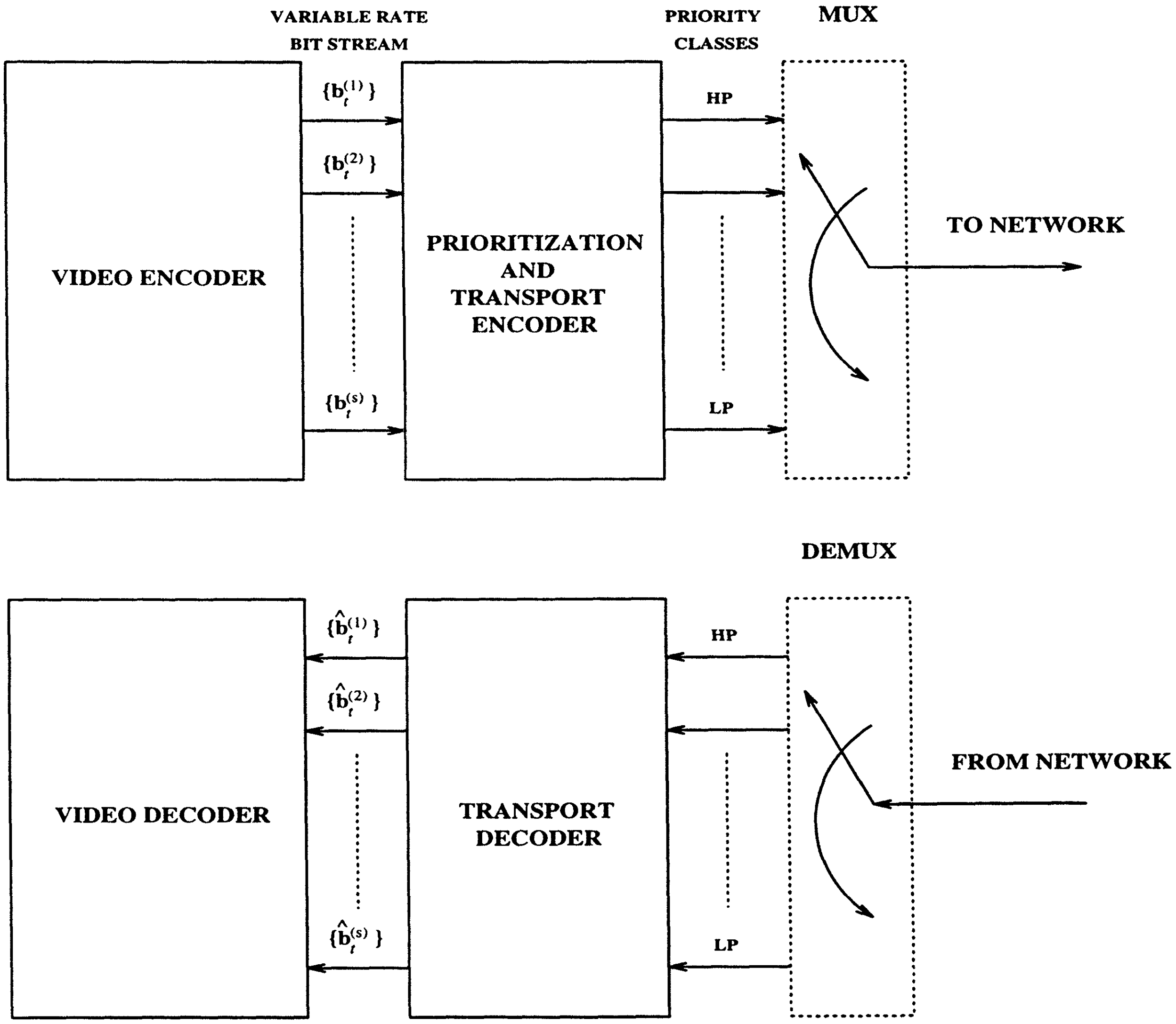


Figure 1: A Generic Block Diagram of a System for Video Transmission over a Packet-Switched Network.

prioritization, consider the case of a two-level priority allocation for a subband-based coding scheme with a uniform 16-band decomposition. The encoded bits from subbands 1-4 could be allocated the highest-priority (HP) level (priority level 1) and subbands 5-16 allocated the lowest-priority (LP) level (priority level 2). The two priority packet streams would be the resulting output of the prioritization and transport encoder block. Correct reception of subbands 1-4 would guarantee a low-resolution video sequence while correct reception of all 16 subbands would provide the highest-level video resolution. For tractability and ease of simulation, we do not consider more than two priority levels in our work. Generally, prioritized transmission mitigates the effect of packet loss. Prioritization at the source is beneficial in situations such as buffer overload at intermediate nodes of the network as it allows less important information to be dropped first.

Following the prioritization, the output packet streams are multiplexed and transmitted over the network. The multiplexer interfaces to the network at its output port and typically operates at a very high speed, e.g. 155 Mbps. Given present-day hardware capabilities and the simple task of the multiplexer (which in the block diagram is really a parallel-to-serial converter), its operation can be assumed almost instantaneous. As a result, the delay introduced by this block is assumed negligible.

A block diagram of a generic backward[1] motion-compensated predictive interframe image sequence coding system is illustrated in Fig. 2. The input frame sequence $\{\mathbf{X}_i\}$ to the encoder in Fig. 2a is motion-compensated predicted, based upon the locally reconstructed sequence $\{\hat{\mathbf{X}}_i\}$, to produce the predicted sequence $\{\hat{\mathbf{X}}_i^+\}$. The residual sequence $\{\mathbf{E}_i\}$, representing the frame differences $\mathbf{E}_i = \mathbf{X}_i - \hat{\mathbf{X}}_i^+$, is then encoded by a 2-D intraframe encoder. We assume the latter includes entropy coding of some form resulting in a variable-rate encoded output sequence $\{\mathbf{b}_i\}$. This variable-rate sequence is then packetized and transmitted. Detailed discussion of the packetization strategy will be provided later in this section. The decoder in Fig. 2b provides the inverse operations of the encoder based upon the received sequence $\{\hat{\mathbf{b}}_i\}$ which may differ from the corresponding transmitted sequence due to errors or packet losses. In one respect, packet losses are easier to deal with than bit errors because their location is assumed known through sequence numbers in the packet header.

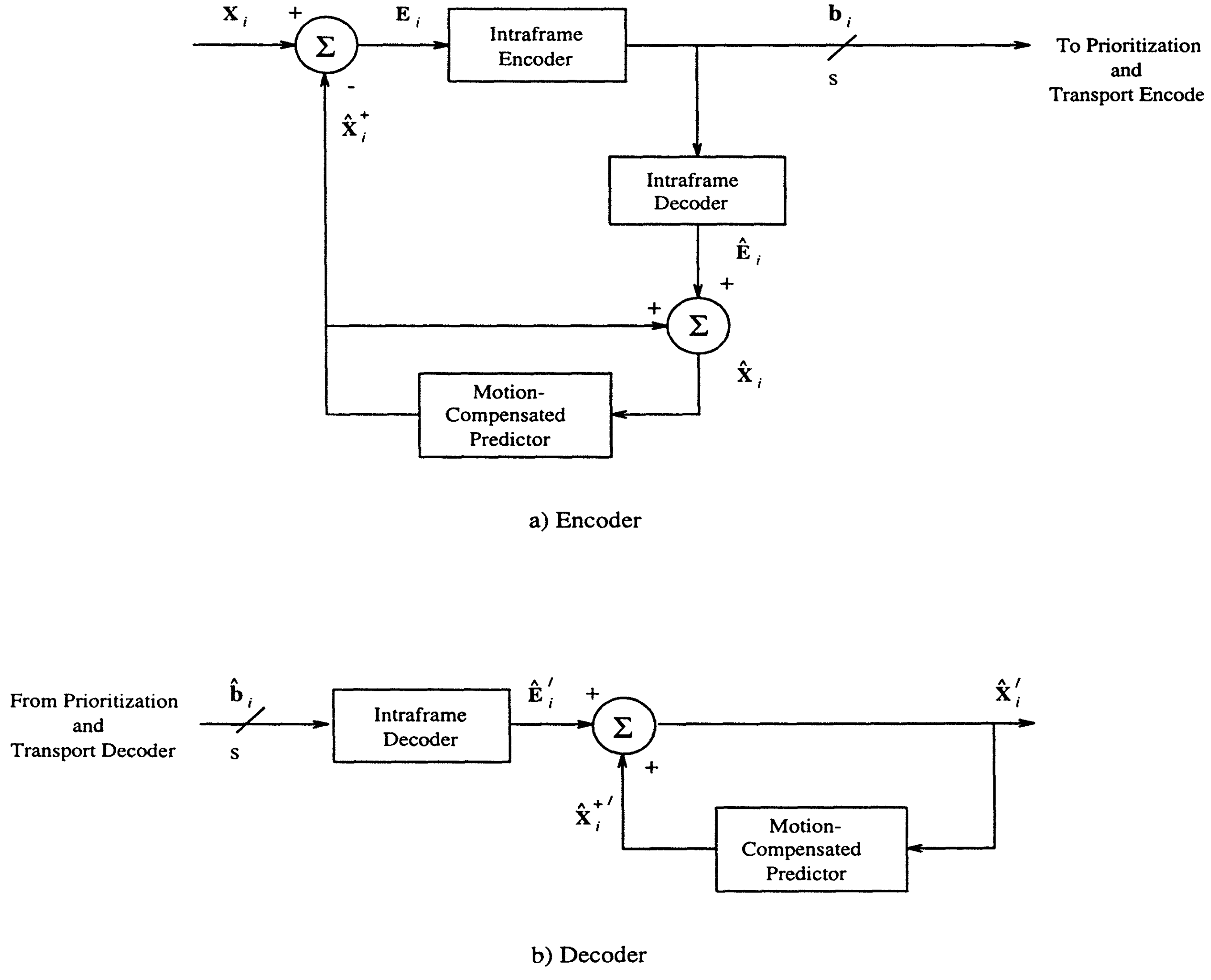
The particular motion-compensated prediction scheme employed here to illustrate our approach is a hierarchical, pyramid-based, pel-recursive approach. Specifically, crude motion estimates are first obtained at the lowest resolution of the pyramid by a pel-recursive scheme. These estimates are then used as initial estimates for a similar pel-recursive motion estimation at the next highest resolution of the pyramid. This process is then repeated to the highest resolution level representing the fullband image. This hierarchical scheme, which employs a 3-level pyramid, is described in more detail in [13]. This particular combination of pyramid-based hierarchical motion-compensated prediction (HMCP) and 2-D intraframe entropy-constrained[2] subband coding (ECSBC) is called HMCP/ECSBC [13]. Although originally intended for operation over a fixed-rate channel through use of a buffer-instrumented adaptive rate[3] mechanism, the basic HMCP/ECSBC scheme can also be used on variable-rate ATM-based packet networks as considered here. In [13], an explicit procedure is described for design of a HMCP/ECSBC scheme to minimize the distortion subject to a constraint on the overall average rate in bits/pixel.

The particular intraframe encoding scheme to be considered here is the 2-D ECSBC scheme

[1] By backward we mean that motion-compensated prediction is based upon the encoded sequence and does not require separate transmission of motion vectors as overhead information.

[2] In particular, the encoder is designed to minimize distortion subject to an overall rate or entropy constraint.

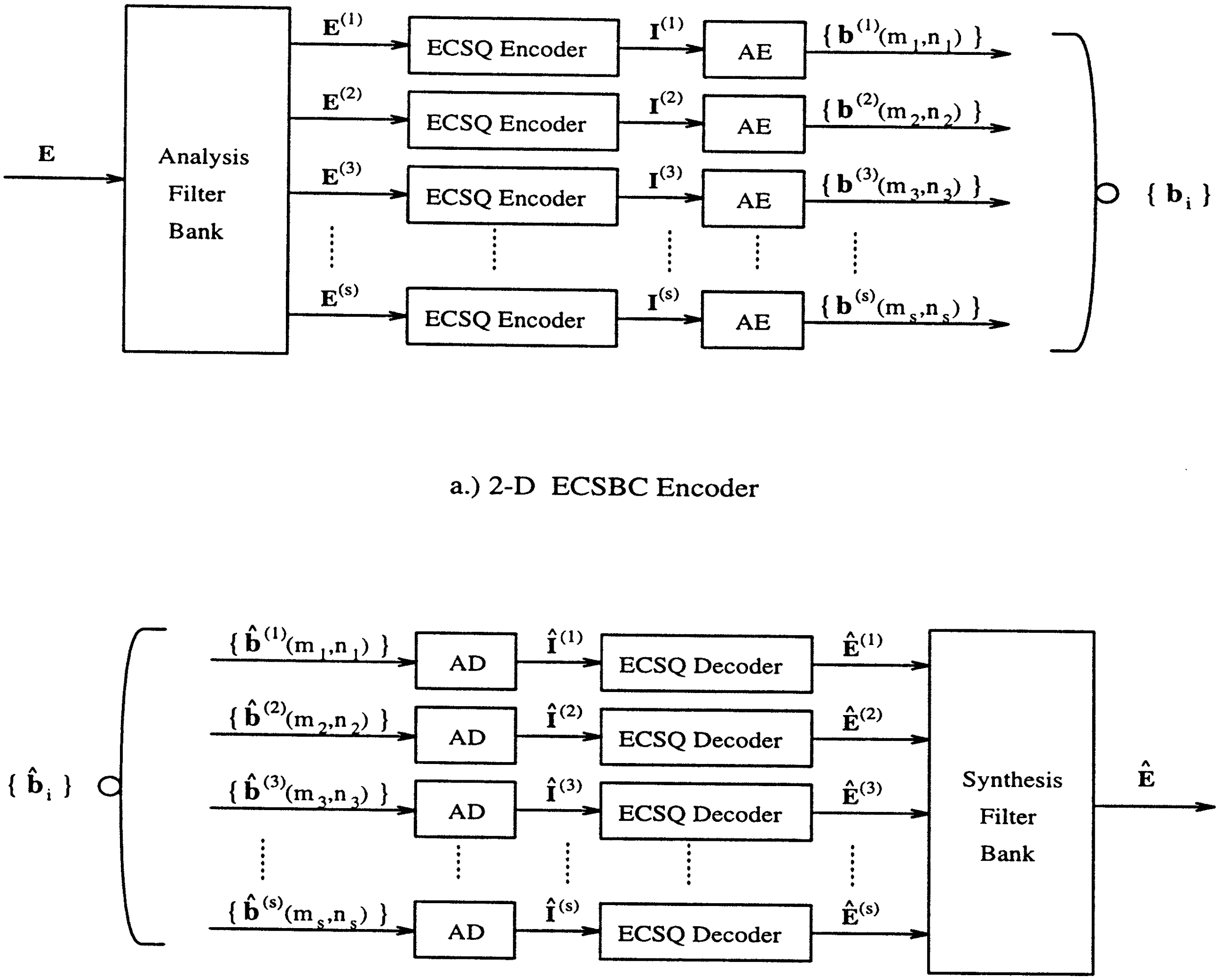
[3] Here, the term refers to the use of the encoder buffer to vary the rate through the use of an appropriate feedback mechanism.



a) Encoder

b) Decoder

Figure 2: Block Diagram of the Generic Backward Motion-Compensated Predictive Interframe Image Sequence Coding System.



a.) 2-D ECSBC Encoder

b.) 2-D ECSBC Decoder

Figure 3: Block Diagram of 2-D ECSBC System for the Residual Frame.

illustrated in Fig. 3 and described in more detail in [11]. Specifically, the residual frame is first decomposed into 16 uniform separable subbands ($s = 16$ in Fig. 3) which are then independently applied to an equal number of optimized entropy-constrained scalar quantizers (ECSQ's). An optimum ECSQ is one that minimizes the average distortion, $D(H)$, subject to a constraint, H, on the quantizer output entropy. The analysis filter bank decomposes the residual frame, $\mathbf{E}$, of dimension N_h by N_v, into s uniform subbands. For each residual subband $\mathbf{E}^{(k)} \triangleq \{E^{(k)}(m_k, n_k);\ 0 \leq m_k \leq {N_v}^{(k)} - 1,\ 0 \leq n_k \leq {N_h}^{(k)} - 1\}$, $k = 1, 2, ...16$, the corresponding ECSQ provides a quantizer index, $I^{(k)}(m_k, n_k)$, for each spatial position (m_k, n_k). Here we take ${N_h}^{(k)} = \frac{N_h}{4}$, ${N_v}^{(k)} = \frac{N_v}{4}$, $k = 1, 2, ...16$. As in [11, 12], following the analysis filter bank various scanning and formatting techniques can be used for encoding the subbands so that a variety of SBC schemes can be developed within the same generic 2D-ECSBC framework. However, in our case, each subband is processed in a row-by-row raster scan order and a memoryless scalar quantization scheme is adopted for simplicity and to facilitate easy recovery under packet loss. The quantizer output indices are then entropy encoded, using an arithmetic coding scheme, to produce the variable-rate outputs denoted as a vector with variable dimension, $\{\mathbf{b}^{(k)}(m_k, n_k)\}$, $k = 1, 2, ..., 16$. An appropriate means of classifying these bit streams into priority classes based on the provision of scalable video was described earlier. In the most general case, each of these 16 output streams can be packetized individually and classified into separate priority levels before transmission over the network. The reverse operation takes place at the decoder illustrated in Fig. 3b. In our work, where we restrict attention to at most two priority[4] classes, the bit-stream from encoding subbands 1-4 forms the highest priority class providing the base resolution, while the bit-stream from subbands 5-16 forms the lower priority class and provides enhanced resolution.

To operate over a network with packet loss, the entropy coder operation needs to be modified. Our entropy coder modification follows some of the work in [14] where a run-length coder was modified to be able to function under packet losses. As used in [11, 12], the arithmetic coder (which is the entropy coder used in our work) functions by encoding the complete set of quantizer indices from all the decomposed subbands as a single message. Although appropriate for use over noiseless channels, direct use of such an arithmetic coding scheme could potentially be catastrophic under packet loss. With missing packets, synchronization is completely lost at the arithmetic decoder and the decoding process cannot be continued beyond that point. However, if some resynchronization capability were provided after entropy coding a fixed number of quantizer index values, the decoder would contain specific points in the bit stream allowing it to re-enter the decoding process under packet loss. Suppose L_I were to represent the number of source symbols (quantizer index values) which are entropy coded after which resynchronization capability is provided in the form of flags. The exact technique of resynchronization will be discussed later. For reasonable values of L_I, hardly any encoding efficiency is sacrificed in block arithmetic coding as compared to arithmetic coding the entire set of quantizer indices as a single message (which essentially means that L_I = number of quantizer indices in the frame).

Briefly then, in the case of a lost packet, the entropy coder synchronization is maintained by continuing to decode from the earliest available starting point. Such a starting point in the bit stream is indicated by the end-of-block (EOB) symbol in the arithmetic coder (which in turn means that L_I symbols have been encoded) and can be implemented through the provision of unique flags. Uniqueness of the flag is preserved by using the technique of bit stuffing to prevent the occurrence of the flag in the coded information bit stream. In our work, we use a string of N_{EOB} consecutive $1's$ to specify a flag. Additionally, the flag is followed by a fixed-length flag counter which indicates the corresponding flag number, thereby providing information about the number of flags lost during transport. Hence, correct decoder initialization is possible in the case of missing packets. Ideally, the number of flags used should be kept to a minimum. On the other hand, using large L_I values leads to an increased amount of missing information before entropy coder resynchronization. Also, in some higher subbands, the amount of overhead due to the use of flags is comparable to the information generated in these subbands. Therefore, we minimize the number of flags used by positioning the flags based on the specific subband as well as the average rate of operation. The actual positioning of the flags, or equivalently the choice of L_I, was optimized empirically. This was performed based on the video sequence used, the rate of operation and the subband being encoded so as to minimize the overheads while at the same time maintaining the ability to resynchronize. For example, in our experiments with the 16-band subband coding scheme at a source coding rate of approximately 0.85 bits/pixel, we found that an L_I of 176 for the high-priority bands (subbands 1-4) and 352

[4] Current ATM standards allow only two priority classes.

(subbands 5-16) for the low-priority bands and N_{EOB} of 8 bits was a reasonable tradeoff between resynchronization ability and extra overheads. These values are therefore used in the course of our simulations performed in Section V. Further details can be found in [23].

Since the motion compensation scheme used in our work is a backward motion compensation scheme, there is no explicit transport of motion vector information (in contrast to a forward motion compensation scheme). Thus, the packetized and transported information is comprised only of the entropy-coded residual error frame data which leads to a more efficient packetization of the encoded bit stream. However, there is a penalty in the form of slightly higher error propagation as the transmitted information is relatively more sensitive to packet loss.

The bit stream produced by entropy coding each subband is packetized separately. The same is true for bits from the current and successive frame. In other words, if the bit stream produced by entropy coding a specific subband does not fill an integral number of packets, then padding bits are added to complete the last packet. Recently, Ghanbari and Hughes [15] discussed packetization strategies for a H.261/MPEG coder. They employed two methods of packing bits into cells which they term as *close packing* and *loose packing.* In close packed cells, macroblocks (in H.261/MPEG coding takes place by subdividing the image into groups of pixels known as macroblocks) may be split and placed in different cells while in loose packing, a cell may only contain an integral number of macroblocks. They show that using close packing of cells gives marginally better performance in an ATM network. Our method of packing of cells is in a sense similar to their technique of loose packing of cells. Though close packing might be more efficient, in our case, a loose packing of cells reduces delays at the prioritization and transport encoder. Indeed, additional delay would otherwise be introduced due to the packetization process. To a certain extent, this would depend on the hardware used for encoding. However, note that excessive packetization delays are avoided since packets not completely filled in the current subband are not held up waiting for bits obtained through entropy coding the next subband. Delay issues will be discussed in detail in Section III.3. Finally, we should note that, in general, a separate encoder/decoder would be provided for the luminance and the chrominance components. In this work we consider only monochrome image sequences so that only one encoder/decoder is required for operation on the luminance component. In the next section, we discuss in detail the application of FEC codes and the corresponding delay implications.

III. FEC codes and their application

In this section, we discuss the interlaced application of FEC codes to ATM cells and the FEC coding delays involved in this operation. Reed-Solomon (RS), Bose-Chaudhuri-Hocquenghem (BCH) and many other classes of codes can be used for FEC [18]-[20]. RS codes make use of the generated overheads efficiently and have attractive minimum distance properties. Accordingly, they can be used effectively for burst erasure recovery which will prove valuable in the face of correlated cell loss. Therefore, although our methodology extends easily to other codes as well, we focus on the use of RS codes.

III.1. Application of FEC codes

As illustrated in Fig. 4 for the specific case of an RS code, FEC is applied through interlaced coding across packets by grouping the information bits in the packet into q-bit symbols. The technique used here is the same as the approach in [1]-[2]. The packet size we consider to illustrate

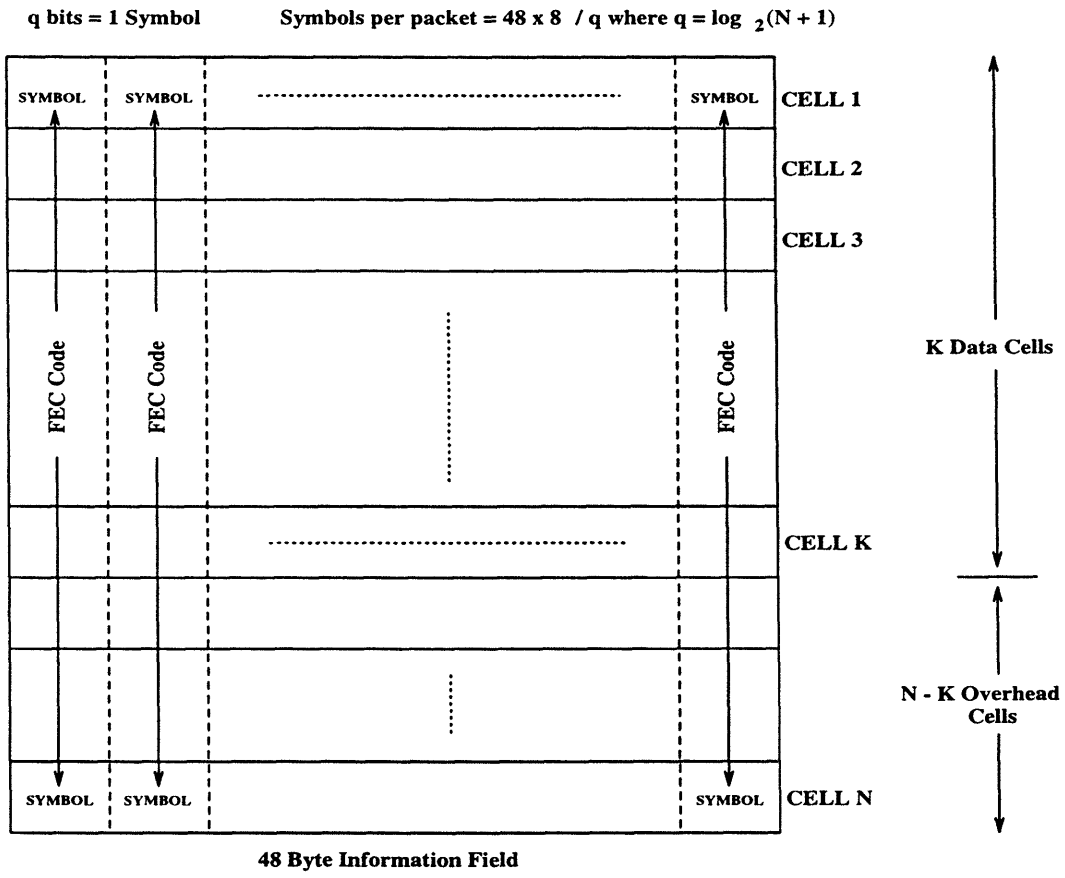


Figure 4: Interlaced application of RS codes on ATM cells.

our approach is 48 bytes which corresponds to a standard ATM cell payload. Use of interlaced FEC coding in an ATM environment is necessitated by the following: Consider a non-interlaced FEC approach. There are three possible cases which we will now discuss. Nq could be smaller than the ATM cell size, a few times the cell size or very much larger than the cell size.

First, we consider the case where Nq is chosen to be very much larger than the ATM cell size. For example, suppose Nq is 50 times the ATM cell size. Then a code length of $N = 2047$ (This is obtained by solving for the symbol size q from $Nq = 50 \times 48 \times 8$ and using the fact that $N = 2^q - 1$) is required. In such a case, considerable coding delay and complexity is introduced because of the use of such long code lengths making the scheme impractical.

Now suppose Nq is chosen to be a few times the cell size. For example, suppose Nq is 5 times the ATM cell size and the symbol size is $q = 8$ resulting in a reasonable code length. Then, the loss of a single packet causes $384/8 = 48$ symbols to be lost. This could be even higher when the symbol size is smaller. To recover the loss of so many symbols, an extremely powerful code would be required. In other words, the code rate would have to be low making poor use of the transmission bandwidth.

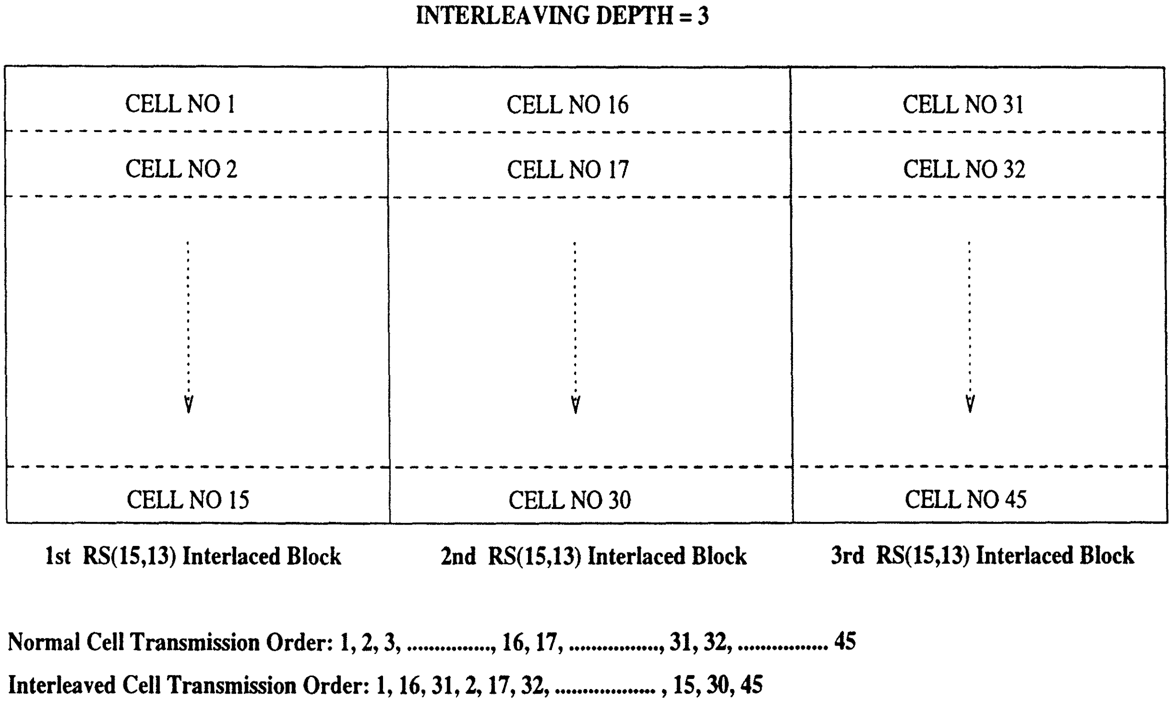
Finally, if Nq is smaller than the cell size then with the loss of a single packet all the N symbols in a codeword are lost making their recovery impossible. Consequently, loss of even a single packet would lead to all N symbols being erased. Thus, we see that non-interlaced coding would generally be ineffective.

III.2. Interleaving to improve FEC efficacy

Interlaced FEC performs best when packet losses are independent. In other words, correlation in packet losses reduces the advantage in using FEC. Unfortunately, packet losses in networks are typically correlated. To reduce the effect of the correlation, interleaving (typically, an interleaver permutes the ordering of a sequence of symbols in a deterministic manner) at the packet level can be used to randomize the loss process, making it appear almost independent at the receiver. One simple way of understanding interleaving is viewing it as packets being filled into a buffer (visualize the buffer as a two dimensional array, each element of the array capable of accommodating a packet) row-wise and transmitted column-wise. Figure 5 illustrates this concept in detail for the RS(15,13) code. This operation would be performed prior to transmitting the packets over the network.

To briefly summarize the overall system operation for the most general case, the coded bit-stream would be first classified and packetized into separate priority classes. For example, in Section II we discussed a possible priority allocation criterion for multiresolution delivery. Following this operation, an appropriate code is applied to each individual priority class. These operations would be performed by the prioritization and transport coding block. Subsequently, the priority classes are multiplexed into a serial stream, interleaved to decorrelate losses, and then transmitted over the network. In the most general case, a separate interleaver would be applied to each of the priority classes due to the use of different FEC codes for each priority class. However, if the same code is used for both priority classes, then a single interleaver would suffice.

In [9] and [23], complexity issues associated with the interleaving operation are evaluated and shown to be sufficiently small for small interleaving depths to allow real-time operation. Delay issues associated with the interlacing and interleaving operation will be evaluated in the next section.



INTERLEAVING DEPTH = 3

CELL NO 1	CELL NO 16	CELL NO 31
CELL NO 2	CELL NO 17	CELL NO 32
⋮	⋮	⋮
CELL NO 15	CELL NO 30	CELL NO 45
1st RS(15,13) Interlaced Block	**2nd RS(15,13) Interlaced Block**	**3rd RS(15,13) Interlaced Block**

Normal Cell Transmission Order: 1, 2, 3, ..., 16, 17, ..., 31, 32, ... 45

Interleaved Cell Transmission Order: 1, 16, 31, 2, 17, 32, ... , 15, 30, 45

Figure 5: Illustration of the operation of a depth 3 packet-level interleaver used with the RS(15,13) code.

III.3. FEC coding delay

As mentioned earlier, FEC coding delay is an important parameter to be considered for practical operation. This coding delay depends on the particular code used, the stochastic nature of traffic and the processing speed. In this section we incorporate FEC coding delay as a constraint in an objective design criterion. We begin by providing an example of typical overall FEC delays and its relevance to the feasibility of using a given code. Further details can be found in [9]. Consider first the evaluation of the basic FEC coding delay. Since the RS code is systematic it is not necessary to buffer at the interlacer to form RS codewords. Rather, the information symbols can be transmitted directly with a local copy retained to form the parity check symbols. These computed parity check symbols can then be sent immediately after the information symbols thereby eliminating interlacing delay at the transmitter. Then the only contribution to the net FEC coding delay (due to interlacing and interleaving) at the transmitting end is due to the interleaving operation. At the receiving end, the contribution to the net delay is due to both the de-interlacing and de-interleaving operations. The average[5] interleaving delay is the average delay introduced in accumulating the required number of packets in the interleaver prior to transmission. A similar argument holds for the de-interleaving delay.

We experimentally determined the overall FEC delay by encoding 150 frames of the *Susie* sequence (CCIR 601) at different rates. Although the simulations provide the number of packets generated in each frame, the interarrival time of packets within a frame cannot be determined as the software simulation does not work in real-time. As a result, we assume a particular model on the interarrival time of packets generated within a frame. In particular, we assume packets generated

[5] Note that the delay introduced due to the FEC application is stochastic due to the VBR output of the video coder.

in a frame are spaced out uniformly. For example, if the frame rate were 30 frames per second and 40 packets are generated in a particular frame, the interarrival time for the packets in that frame is taken as $\frac{1}{(40\times30)}$ seconds. This would then correspond to a packetization and processing delay of $\frac{1}{(40\times30)} \approx 0.8$ msec. Our discussion is based on the assumption of a 20 msec bound on overall delay which provides a benchmark in assessing the tolerable interlacing and interleaving delay. Generally, 20 msec represents the tolerable end-to-end delay in interactive audio communication [16]. In interactive video, since audio information usually accompanies it, 20 msec should likewise represent a benchmark for the tolerable end-to-end delay. However, it should be noted that the 20 msec bound is an upper bound on the end-to-end delay, not just on the interlacing or interleaving delay. We use it solely as a benchmark to judge the FEC delays in relation to this bound. In real-world situations, the end-to-end delay is the determining factor for stable operation.

As seen in Fig. 6, for an interleaving depth of 2, even under low coding rates, the net delay introduced by the RS(7,5) code due to interlacing and interleaving operations is much less than the 20 msec bound. In fact, at rates as low as 0.25 bits/pixel, the delay introduced by the RS(15,13) code is almost an order of magnitude lower than the 20 msec bound. However, below this rate, the use of the code is governed by the expected queueing delays in the network. Large FEC coding delays place critical restrictions on acceptable queueing delays. In short, except under very low coding rates, shorter length RS codes can be successfully interleaved up to depths of $2-3$ with small net FEC coding delay due to interlacing and interleaving operations. In the case of the RS(63,57) code, even for interleaving depths of 2, the net FEC coding delay is nearly $4-5$ times that of the RS(15,13) code. Also, using the RS(63,57) code would involve reducing the tolerable queueing delays by nearly half to meet the 20 msec benchmark. Such a performance restriction is generally difficult to achieve, especially for interactive applications, thereby precluding the possibility of interleaving the RS(63,57) code. This example demonstrates that the FEC delay places severe restrictions on the feasibility of using a particular code. More detailed evaluations for other interleaving depths can be found in [9] and [23].

The evaluation of coding delay illustrated in Fig. 6 was experimentally determined for a CCIR 601 sequence. Indeed, for any general video sequence, the delay introduced is a function of the image resolution, the frame rate and coder operating rate. Experimental determination of the FEC coding delay would not be possible in most real-world situations where the encoded video material is not available prior to the start of transmission. In such cases, it is necessary to have apriori estimates of the FEC delay when some information on the *average*[6] coding rate is known. We now provide an expression for an approximate calculation[7] of the FEC delay which in later sections will prove useful in obtaining apriori estimates of the overall FEC delay for sequences coded at any rate. Note that this delay includes the delay associated with the interleaving operation. Let N_p represent the *average* number of packets generated in a frame, f be the frame rate in frames per second and M be the interleaving depth. Then the average FEC delay D is given by

$$D = \begin{cases} \frac{N}{fN_p}; & M = 1 \\ 2 \times \frac{MN}{fN_p}; & M > 1. \end{cases} \tag{1}$$

[6] Here, the term average refers to the coding rate averaged over all the transmitted frames. Arguably, such information too may not be exactly known while employing VBR coders prior to encoding the sequence, especially since the video material characteristics could be arbitrary. However, knowing the target quality intended to be achieved, some idea about the nature of the encoded information from past experience and the typical operational rate-distortion characteristics of the video coder, one can arrive at intelligent estimates of the average rate it would take to encode the source material.

[7] The expression for analytically evaluating the FEC delay is an approximation due to the fact that it employs the *average* rate for its computation. In other words, the frame-to-frame variation in the rate is ignored.

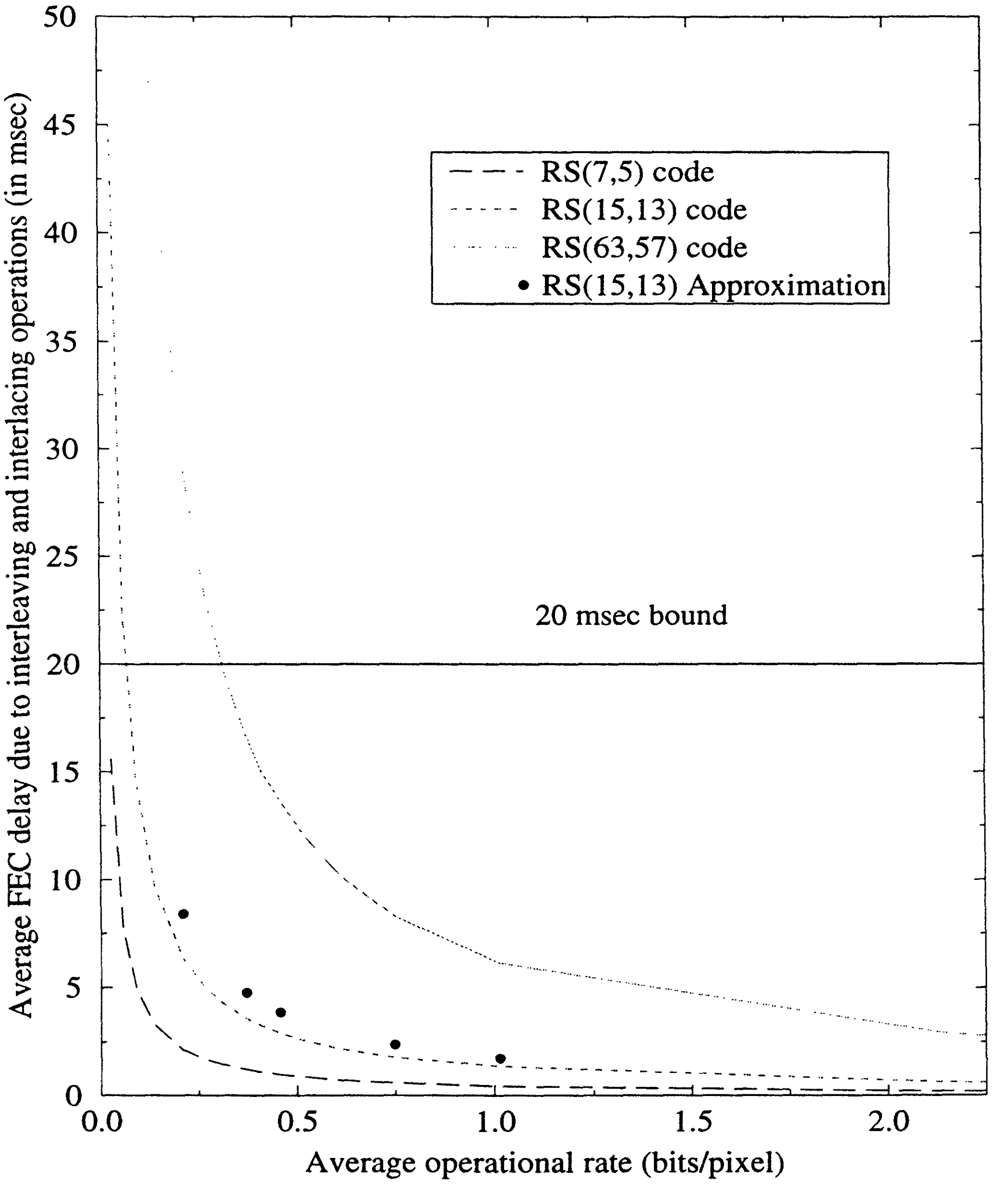


Figure 6: Net interlacing and interleaving delay for an interleaving depth of 2 at different operational rates computed by coding the *Susie* sequence.

As illustrated in Fig. 6, for the RS(15,13) code, this approximation compares favorably with simulation results. The derivation of this approximation for the average interlacing delay is provided in Appendix A. This approximation also proves helpful since the experimental FEC delay determination is tedious in most situations, especially since the video source material can be coded at various rates or resolution levels.

A major issue then concerns the validity of the approximation. Notice that the approximation uses the average number of packets generated in predicting the average FEC delay. Hence, this makes it independent of the particular sequence being encoded as long as information about the average coding rate is available. Suppose there is very little variance in the output rate produced by the coder. In that case the approximation is extremely tight. On the other hand, if the variance in the output rate is very high, then the approximation is not very good.

IV. Performance Analysis Methodology

In this section we first discuss the code selection policy employed in our work. We then discuss the channel model used to capture the packet loss behavior. Finally, we explain the calculation of the fidelity metrics.

IV.1. Code selection principle

Since the FEC application is performed by throttling the coding rate to accommodate the FEC overheads, the throttling operation is to be minimized to prevent sacrifice in quality under light loads resulting in small packet loss rates. In other words, a critical parameter is the *code rate* $R = K/N$ which determines the fraction of the overall rate allocated to the source coding operation. Therefore, it is desired to make K/N as close to 1 as possible, which is the ideal code rate under lossless conditions. At the same time, to provide good protection with reasonable delay, it is desired to maintain the FEC coding delay and the decoded loss probability below appropriate thresholds. Therefore, we choose the code which solves the constrained optimization problem:

$$\begin{array}{ll} \text{maximize:} & R = \frac{K}{N} \\ \text{subject to:} & \text{Decoded loss prob} \leq L_{threshold} \\ & \text{FEC coding delay} \leq D_{threshold}. \end{array} \tag{2}$$

Here, $L_{threshold}$ and $D_{threshold}$ are the quality-of-service (QOS) constraints which refer to the thresholds below which the decoded loss probability P_{dec} and FEC coding delay are to be maintained. P_{dec} refers to the packet loss probability after the FEC decoding operation is performed. In such a formulation, we assume the environment to be jitter tolerant. In other words, the jitter is not included as a constraint. We adopt a brute force method of enumeration in order to find the required code. The maximization of K/N is over all RS codes including the shortened and extended codes. The steps in the search can be outlined as follows.

1. Select a symbol size (start from the smallest symbol size)

2. Construct a RS code for this symbol size

3. Select the interleaving depth (start from 1)

4. Evaluate FEC delay D for this particular code using (1) and compare with $D_{threshold}$. If threshold is met, go to 5, else go to 1.

5. Evaluate loss probability achieved using this code and compare with $L_{threshold}$. If threshold is met, go to 6, else go to 2. Computation of the loss probability for a particular congestion model is described in the next section.

6. Compare the code rate K/N with current maximum. If higher, update current best code rate.

7. Evaluate the above steps for all RS codes possible for that symbol size.

8. Stop when the symbol size is equal to the largest symbol size.

Note that cycling through steps 4 and 5 also yields the smallest interleaving depth which meets all the thresholds specified. A maximum symbol size of 8 and interleaving depth of 3 is sufficient to cover all cases of interest[8] in our work. Although it is somewhat tedious, such an enumeration needs to be performed only once for a given set of parameters. Therefore, by pre-storing a set of good codes for the set of frequently occurring channel conditions, the search can be limited. Furthermore, as we will demonstrate later, moderate codelengths are sufficient to obtain codes that provide efficient and robust performance. Hence the search space can be restricted.

In the next section, we present a Markov model used to capture the behavior of the packet loss process.

IV.2. Model for representing congestion

Modeling packet loss in high-speed integrated networks is a challenging problem. While a variety of sophisticated techniques and models have been developed, our interest in modeling packet loss and congestion is only one component in our end-to-end view of packet video. Thus, it is essential that our model be simple and tractable.

The most important characteristic of packet loss in networks which will challenge recovery schemes for video is the correlation between losses caused by buffer overflow and cell dropping by the network (as a form of congestion control). The effect of correlated losses is normally[9] more severe than the effect of independent losses. If the output of the video coder were ideally interleaved, the packet losses would be independent. To represent the packet loss behavior under non-ideal interleaving (which is a practical requirement) in our simulations using real-world sequences, we use a Markovian loss representation. Such a generic model emulates the basic end-to-end loss characteristics in the entire network without requiring detailed explicit knowledge of the particular topology. A more finely tuned representation of the end-to-end packet loss behavior is a difficult task. Complex models do not allow flexibility or tractability in analysis. Furthermore, the number of parameters that need to be selected is quite large making it difficult to interpret the results. In our work, in order to account for the correlated packet loss properties, we have modeled the

[8] Note that higher codelengths and interleaving depths lead to FEC coding delay in excess of the tolerable end-to-end delays for real-time interactive communication. Furthermore, the FEC coding/decoding complexity also increases with the codelength.

[9] However, in the case of TCP/IP for data transport over ATM networks, correlated cell loss might be preferred. In this case, many ATM cells would normally be used to transport a single IP datagram. Loss of an ATM cell would then require retransmission of the entire datagram. In such a case, it would be preferable to have correlated cell losses in order to minimize the number of corrupted datagrams and consequently the number of retransmissions.

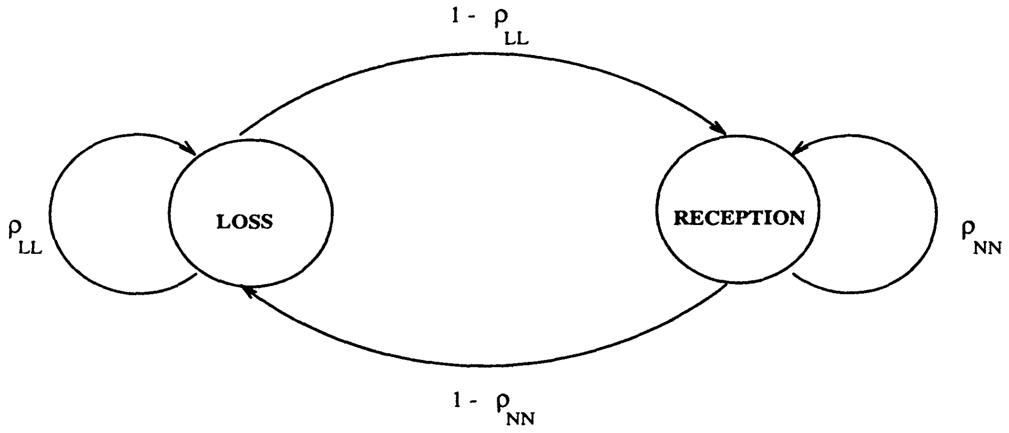


Figure 7: Markov model representation of the packet loss behavior.

packet loss behavior by a simple two-state Markov chain as illustrated in Fig. 7. In the particular case of an ATM multiplexer, a two-state Markov model can characterize the multiplexer buffer overflow-underflow process [17].

In Fig. 7, ρ_{LL} is the one-step transition probability for remaining in the packet or cell loss state while ρ_{NN} is the corresponding one-step transition probability for remaining in the correct packet reception state. The steady-state probabilities of losing and correctly receiving a packet are then given, respectively, by

$$P_L = \frac{1-\rho_{NN}}{2-\rho_{LL}-\rho_{NN}}, \tag{3}$$

and

$$P_N = \frac{1-\rho_{LL}}{2-\rho_{LL}-\rho_{NN}}. \tag{4}$$

Given a desired value of the steady-state loss probability, P_L, once ρ_{LL} is chosen the other parameter ρ_{NN} can be readily calculated from (3), (4) and thus the model parameters become completely specified.

In general, the behavior of packet loss in a network would be time-varying. Among the many quantities it depends on are the load, processing speeds, buffer sizes and number of intermediate links. However, we begin with a two-state model whose parameters are fixed throughout the transmission. Such a situation would be valid over small segments of transmission where stationarity can be assumed. Also, in the case where network feedback is available, such a model would be valid in between feedback instants. Furthermore, assuming such a stationary model allows us to freely vary the parameters and understand their performance implications. The results so obtained would then be useful in designing for more realistic situations.

In Appendix B, we provide a method for calculating the decoded packet loss probability for this channel model. This computation is required in step 5 of Section IV.1.

IV.3. Computation of the fidelity metric

In this section, we explain the computation of the objective fidelity metric used in our examples. Though subjective measures are more appropriate, objective measures are easier to compute especially in cases where extensive simulation is performed. To accurately capture the performance and yield sufficient statistical confidence, a group of N_f input frames is first coded, and the packet loss is emulated by the Markov loss model (as described in Section IV.2) using a particular seed for the random number generator. Next, the input frames are updated by N_f frames to a new set of N_f frames in the sequence and a new seed is used for the random number generator. To compute our fidelity metric, we evaluate the distortion between the reconstructed and input frames. This is then averaged over all the frames in a run and then over all the runs. More specifically, the distortion[10] is taken as

$$D \triangleq \frac{1}{N_t N_f N_v N_h} \sum_{l=1}^{N_t} \sum_{i=0}^{N_f-1} \frac{1}{{\sigma_{i,l}}^2} \sum_{m=0}^{N_v-1} \sum_{n=0}^{N_h-1} [X_{i,l}(m,n) - \hat{X}'_{i,l}(m,n)]^2, \tag{5}$$

where N_t is the number of runs, each of N_f frames, and N_h and N_v represent the horizontal and vertical resolution of a frame, respectively. Here, $X_{i,l}(m,n)$ denotes the intensity at pixel location (m,n) in the l^{th} run within the i^{th} encoder input frame $\mathbf{X}_{i,l}$ while $\hat{X}'_{i,l}(m,n)$ denotes the corresponding intensity at the same pixel location in the final reconstructed frame $\hat{\mathbf{X}}'_{i,l}$ at the decoder. Also, ${\sigma_{i,l}}^2$ denotes the variance of the i^{th} input frame in the l^{th} run. Thus, by making N_t very large, an accurate evaluation of distortion can be made taking into account both the statistical idiosyncrasies in computing relatively short-term statistics as well as the input frame characteristics.

V. Results and Discussion

In this section, we use the code selection strategy and performance evaluation technique described in previous sections in our numerical results and provide interpretations. For clarity, this section is further divided into 3 sections. First, we study some tradeoffs in the codes selected using this principle. Following this, we compare the optimized performance to theoretical bounds. Finally, we provide extensions of this code selection principle to multi-priority classes.

V.1. Preliminary evaluation of code selection principle

Figures 8 and 9 show the optimized code rates for two examples of the Markov channel model. We will first provide some comments about the choice of parameter values used in the example before discussing the performance trends depicted. In the first example, the loss probability P_L is 5×10^{-3} and in the other example, the loss probability is 1×10^{-2}. Both examples have $D_{threshold} =$ 5 msec and $L_{threshold} = 10^{-4}$. As mentioned earlier, we do not include the jitter as a constraint to simplify our evaluation. However, including the jitter in the evaluation is relatively straightforward. Although 20 msec typically represents the tolerable end-to-end delay for interactive applications, a value of 5 msec is chosen for $D_{threshold}$ to allow for queueing delays which typically dominate end-to-end delay. We also assume in these examples a CCIR 601 video resolution. Recall from Section III.3 that the delay introduced by the FEC application can be either computed experimentally by encoding the source material or analytically approximated through a formula. As described in Section III.3, the analytical expression provided in (2) yields a relatively good estimate and can be easily calculated in real-world scenarios. Therefore, in these examples, the approximation provided in (2) was used in the computation of the FEC coding delay. However, simulations indicate that

[10] The distortion is the normalized mean-squared distortion where the normalization factor is the input frame variance. This allows a more meaningful representation of the distortion metric.

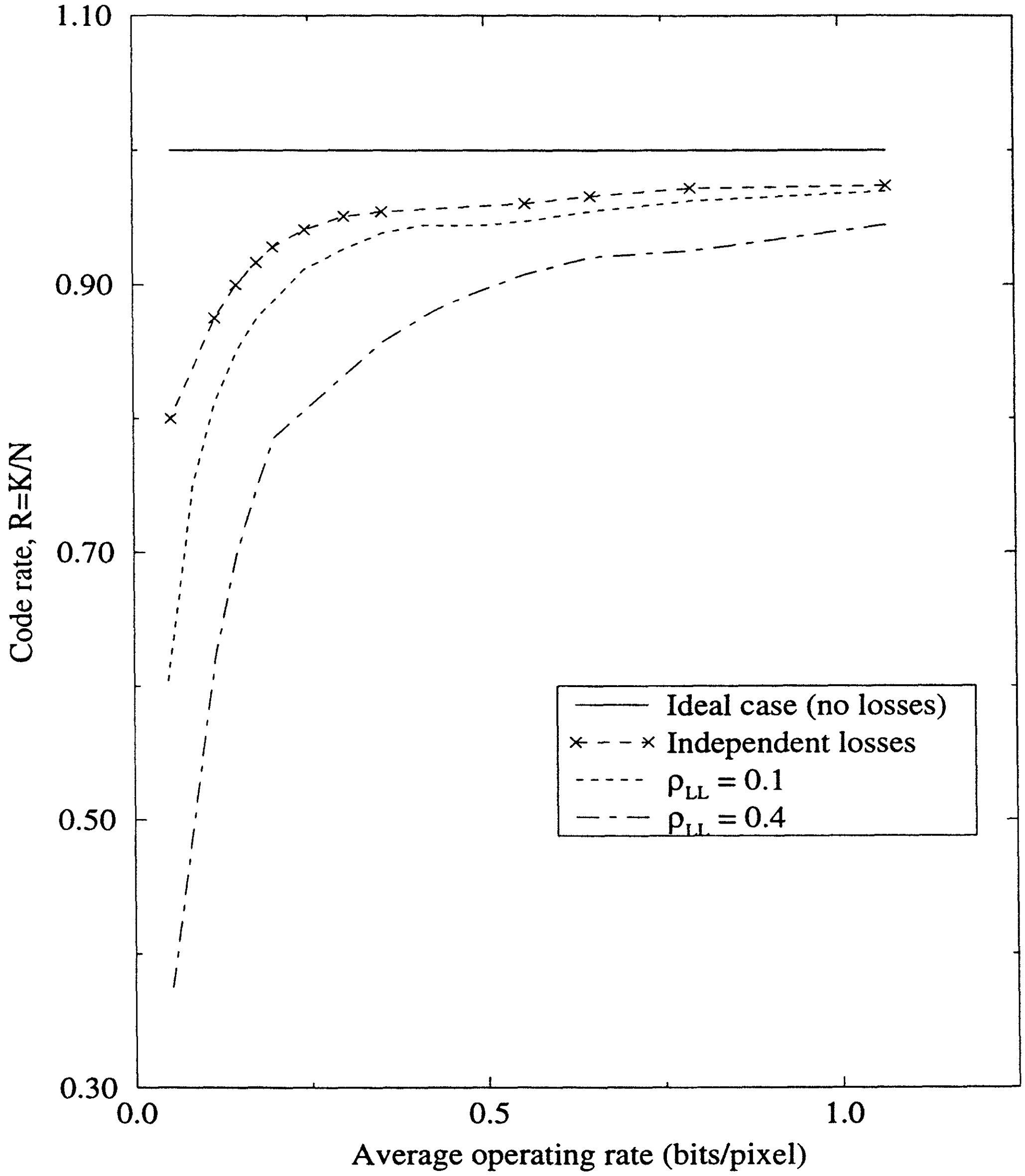


Figure 8: Typical profile of optimized code rate (vs.) transmission rate. In this example, the loss probability $P_L = 5 \times 10^{-3}$ and $D_{threshold} = 5$ msec while $L_{threshold} = 10^{-4}$.

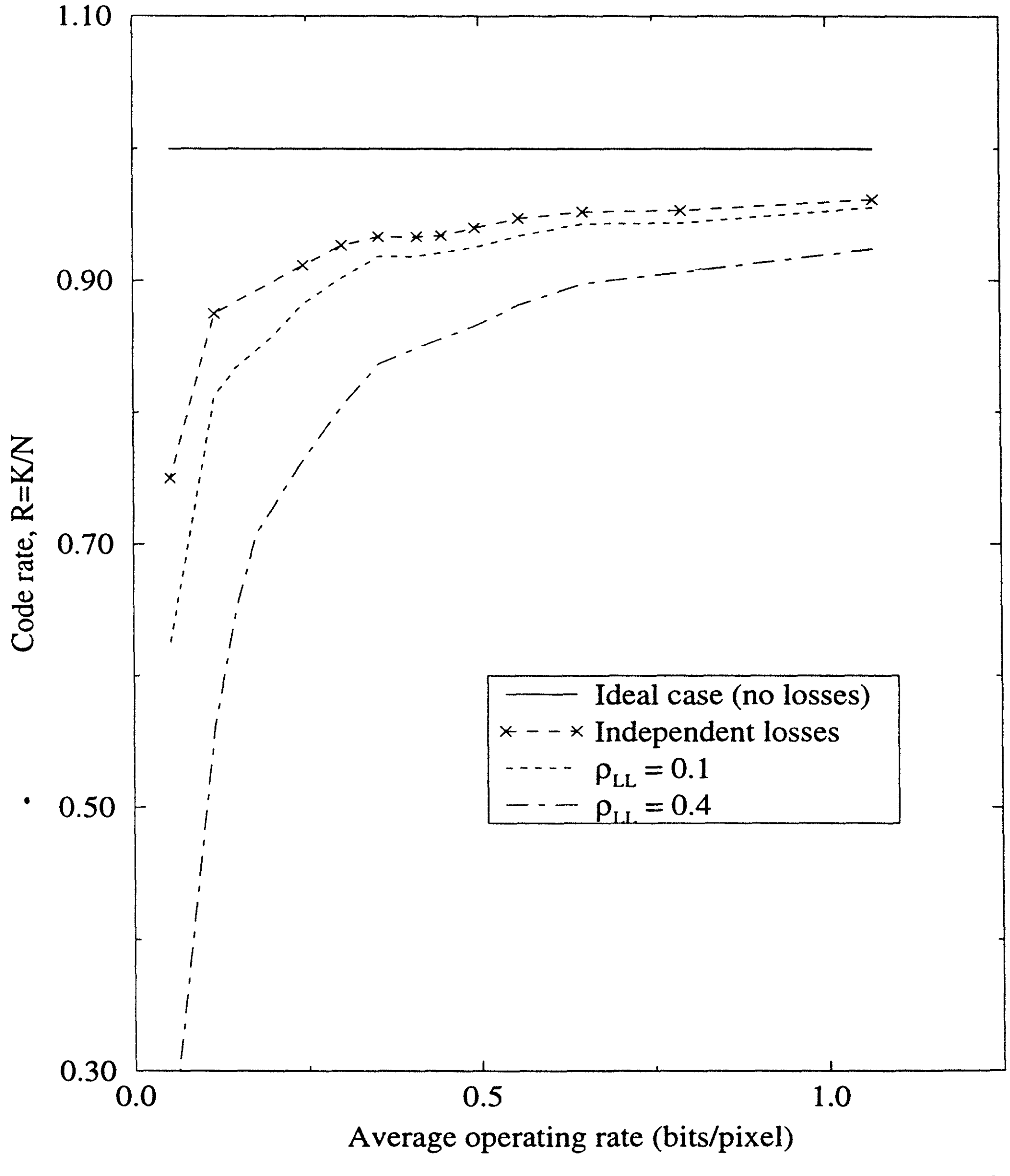


Figure 9: Typical profile of optimized code rate (vs.) transmission rate. In this example, the loss probability $P_L = 1 \times 10^{-2}$ and $D_{threshold} = 5$ msec while $L_{threshold} = 10^{-4}$.

Operating rate (bits/pixel)	Code selected	
	$\rho_{LL} = 0.10$	$\rho_{LL} = 0.40$
0.75	RS(102,98)	RS(90,83)
0.60	RS(82,78)	RS(82,75)
0.40	RS(54,51)	RS(55,48)
0.10	RS(14,11)	RS(14,8)

Table 1: Codes selected at different operating rates for the same example as that in Fig. 8.

Operating rate (bits/pixel)	Code selected	
	$\rho_{LL} = 0.10$	$\rho_{LL} = 0.40$
0.75	RS(89,84)	RS(102,92)
0.60	RS(82,77)	RS(82,73)
0.40	RS(49,45)	RS(53,45)
0.10	RS(14,11)	RS(14,7)

Table 2: Codes selected at different operating rates for the same example as that in Fig. 9.

the results are not noticeably different when using the actual FEC delay-operating rate curves experimentally determined for the same sequence.

To save computation time, the code rate evaluation has been performed over selected operating rates. Notice that low optimized FEC code rates are obtained when the operating rate is quite small. This means that substantial throttling of the source coding rate, and hence reduced transmission quality, is required to accommodate FEC overheads. Furthermore, observe that the selected code depends on the operating rate. This occurs because the FEC coding delay depends on the operating rate according to (1). Hence, a code which satisfies the delay threshold at a given operating rate may not do so at a lower operating rate. Consequently, the philosophy of using only one code independent of the operating rate is questionable. A better approach would be to pre-determine a number of codes, one for each operating rate for fixed channel conditions as in Table 1 and Table 2. These tables provide the particular optimized codes for selected operating rates as plotted in Fig.'s 8 and 9. Once again, notice from Tables 1 and 2 that for the selected codes, the codelengths and code rates are distinctly different at each operating rate and channel condition. This is not surprising given the nature of the code selection process which depends on the model parameters. To achieve adequate protection under overload periods (which could potentially result in loss rates of 1%) at very low transmission rates (e.g., 0.10 bits/pixel) would force the use of as much as 50% of the bandwidth for channel coding. This makes the use of these codes questionable for low-bit rate applications[11]. By predetermining codes, the AAL layer can then use the appropriate code based

[11] Interestingly, this could have some consequence in the manner of applying FEC in random-access channels for packetized video transmission where, to begin with, the available bandwidth to the transmitting application is relatively low. Perhaps, it may be possible to modify the interlacing structure to reduce the delay introduced by the FEC application and improve the code rate.

on operating rate. The selected code can be indicated to the receiving node during the session setup phase at the start of transmission.

The loss probability values used are in general higher than those indicated ($\leq 10^{-4}$) by the ATM specifications for good quality of service. However, transport schemes for VBR traffic that will ensure such quality of service irrespective of congestion and network loads are still under study. Furthermore, the values indicated in the ATM specifications are average values over the entire session duration. The average values do not reflect the short-term characteristics of the loss [21]. Over short time periods the loss probability could increase much beyond the ATM specification values due to temporary network overloads. As a result, we use these higher loss probability values to reflect the transient network behavior. Additionally, the computational time involved to achieve sufficient statistical confidence while performing evaluations with the coder are also drastically reduced for these higher probability values.

V.2. End-to-end evaluation of code selection principle using fidelity metrics

We now present examples using the ECSBC/HMCP coder to highlight the importance of code selection and the performance of a well-chosen code. Concealment is applied in two steps when packet loss occurs. First, the active error recovery technique employing FEC attempts to recover the missing packets. If the active recovery fails, as a second step, a passive error concealment scheme is used. The passive error concealment scheme used is that of temporal interpolation where every lost pixel is replaced by the value obtained from interpolating the values of the pixels in the same spatial location of the previous two frames. This error concealment scheme is simple and requires very little processing at the receiver. As a result, given the relatively complex processing/overheads involved in the decoding operation, its implementation would be quite straightforward. Furthermore, due to the correlated nature of packet loss, the loss of consecutive packets would typically cause the spatially adjoining neighbors of a lost pixel in a particular frame to be missing as well. This thereby necessitates the use of the corresponding information from previous frames in the replacement process.

Before presenting results, using the ECSBC/HMCP coder, we first depict in Fig. 10 the decoded loss probability after the FEC application. This will prove useful in the subsequent interpretation of the important results of this section. Figure 10 shows the decoded loss probability as the code rate $R = K/N$ is varied. In this example, the codelength N is taken to be 63 which represents a code of moderate codelength. Two examples are considered where the loss probability is 5×10^{-3} and 1×10^{-2}. Note that the decoded loss probability decreases quite rapidly with decreasing code rate. Around a code rate of 0.9, the decoded loss probability drops about two orders of magnitude lower than the unprotected case (i.e., $R = 1$). We use the same loss probability values for both priority classes in our example in Fig. 10 so that a two-dimensional depiction of the post-FEC decoded packet loss probability is feasible.

Now, consider the case where the code selection is described by maximizing the code rate K/N subject to constraints as described in (2). Figure 11 demonstrates the effectiveness of the code selection policy for the *Football* sequence. This particular sequence is of SIF resolution (360×288). Though a CCIR 601 sequence would be more appropriate, due to ease of computations as well as drastically reduced simulation times, we restrict attention to a sequence of smaller resolution. The overall transmission rate is 0.85 bits/pixel as it is sufficient to provide reasonable coded quality for this particular sequence. The interleaving depth in this example is 1. In the figure, the normalized distortion computed through (5) is plotted as a function of code rate. The loss probability is 1×10^{-2}

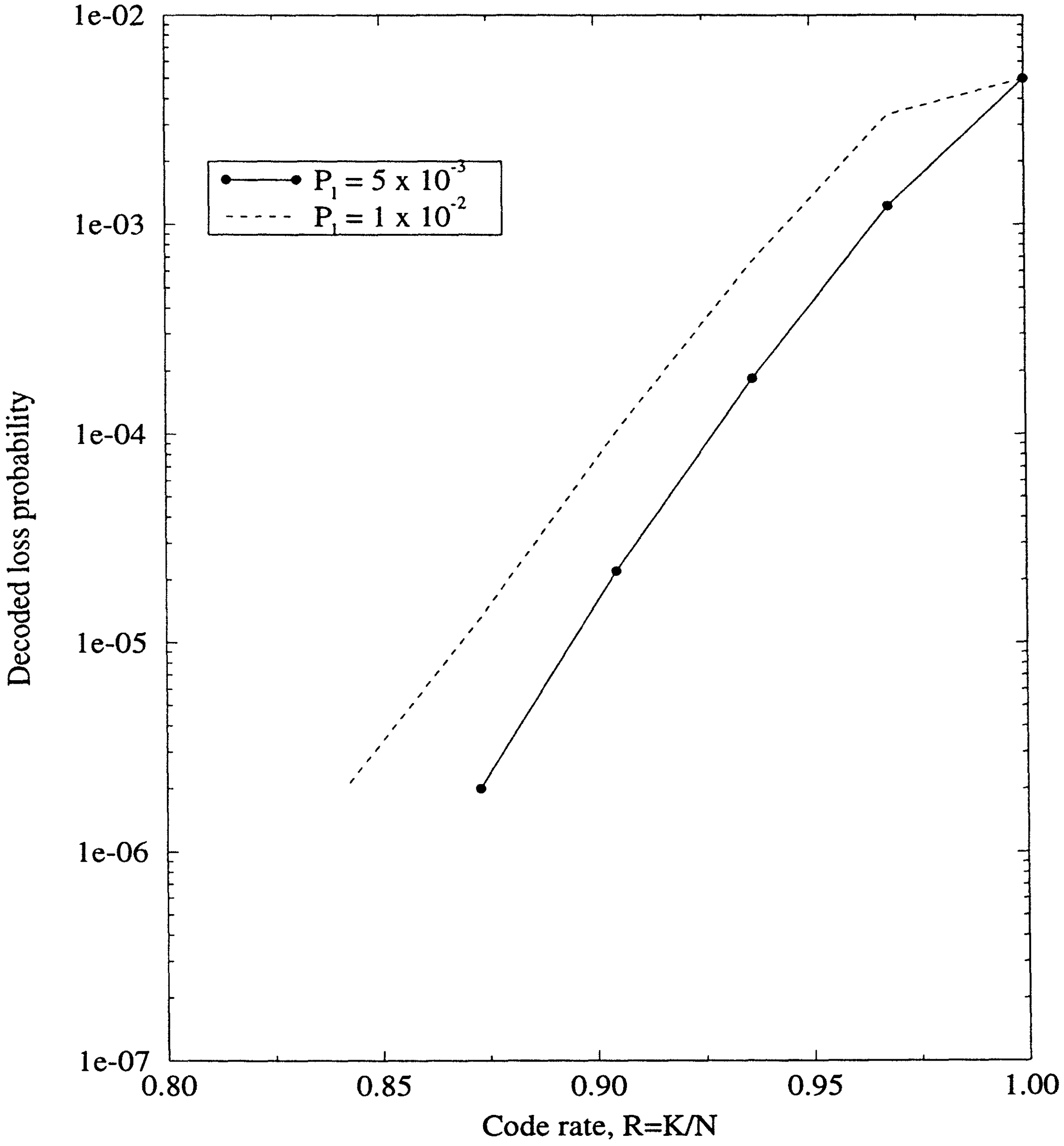


Figure 10: Decoded packet loss performance with a code length of 63 with different code rates under two different values of the loss probability P_L. In this example $\rho_{LL} = 0.25$ in both cases and the loss probabilities are 5×10^{-3} and 1×10^{-2} for both priority classes.

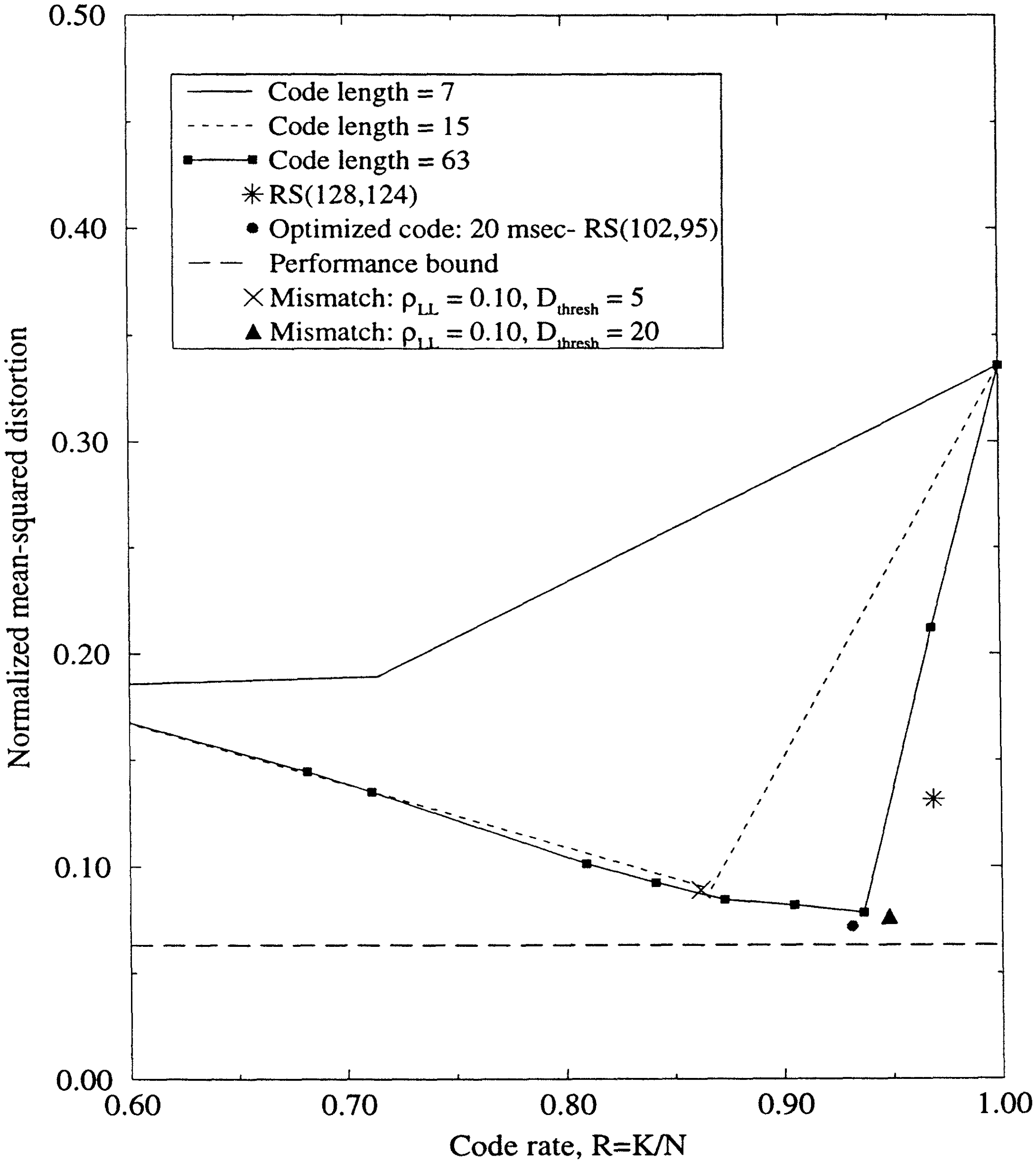


Figure 11: Comparative performance of a hybrid error concealment scheme at different code rates. Loss probability $P_L = 1 \times 10^{-2}$ for both priority classes, $\rho_{LL} = 0.25$, sequence is the *Football* sequence, $N_f = 12$ and the overall average transmission rate is 0.85 bits/pixel.

while $\rho_{LL} = 0.25$. Due to the use of the same code on both priority classes, a single interleaver would be sufficient if interleaving is used. First, the performance is depicted for different code lengths ($N = 7, 15, 63$) as the code rate given by $R = K/N$ is varied. In this particular example, all the depicted codelengths meet the constraint on the FEC coding delay. A code rate of $R = 1$ indicates no FEC is used and hence no throttling is performed. On the other extreme, a code rate of $R = 0$ would indicate that there is complete throttling and hence no information would be transmitted. With a codelength of 7, the set of available RS$(7, K)$ codes is limited. However, with increasing codelength, the set of available codes also increases.

Observe that for any given code length, the performance is maximized at a certain code rate. Additionally, the code rate which maximizes the performance is different for different code lengths. This highlights the importance of code selection and code rate selection in achieving the optimal performance.

Not surprisingly, notice in Fig. 11 that for each of the depicted codelengths, the performance improves with decreasing code rate up to a point because most losses are recovered by the code. At the same time, beyond a certain code rate, too much quality is sacrificed for the channel coding operation. As a result, the quality lost due to the reduced source coding rate is much more than that incurred by packet losses and the resulting imperfect recovery. As seen in Fig. 10, for codes of moderate codelength (≤ 63), the packet loss becomes exceedingly small around code rates of 0.9. Therefore, a point of diminishing returns is seen in Fig. 11 due to the fact that, beyond a certain code rate, the packet losses are reduced to a very small value. Beyond this point, the quality lost by throttling the video coder to accommodate the FEC overheads overrides any performance improvements due to further reduction in losses.

Notice, however, that for any of the depicted codelengths, there is a rapid decrease in the distortion up to the point where the performance is maximized (minimum distortion). After this point, with any further decrease in code rate (which corresponds to the application of more powerful codes), the distortion increases but the increase judging from the slope of the curve is not very steep. The rate at which the distortion either increases or decreases depends upon the sensitivity of the coder towards loss. Therefore, a slight over-estimation of the optimizing code rate is not as penalizing compared to under-estimating it. Beyond the optimal point for each codelength, the increase in distortion is due to source coding quality lost due to the throttling operation. This quality loss follows the slope of the operational rate-distortion characteristics of the source coder operating under lossless conditions. As the distortion does not increase steeply with a slight decrease in the average operating rate, the slope beyond the optimal point is not too steep.

We also depict the code which optimizes the performance for the particular example. This code was chosen in accordance with (2) using a 20 msec constraint on $D_{threshold}$. Though 20 msec may be a little too relaxed for use in real-world situations, it enables us to study the performance when the set of available codes is large. Evaluation of the performance of codes chosen with a more stringent delay constraint is also studied in the figure and will be described in detail in the ensuing paragraphs. For this particular example, the optimized code is the RS(102,95) whose code rate $R = 0.93$. The interleaving depth is 1. Also shown in the figure is the performance of the RS(128,124) code which has been proposed for the AAL 1 layer [7]. The code optimized according to the selection policy performs much better than the RS(128,124) code. In fact, the normalized distortion for the optimizing code is about half that of the RS(128,124) code. This again shows how well-chosen codes can provide good performance.

The figure also depicts an information-theoretic upper bound developed in [9],[23] on the performance. This bound was developed by modeling the packet loss channel as a special case of a *block interference* channel [22]. Notice that the optimized code is very close to the information-theoretic bound in terms of performance. To achieve performance arbitrarily close to the bound is possible but would require extremely long codelengths. This, consequently, would introduce much delay into the system thus affecting its feasibility for real-time communication.

As discussed earlier, with increasing codelength N, the best performance that is achieved also improves. However, increasing codelength is also accompanied by increased delay. Notice that the performance of the code of codelength 63 is very close to the optimizing code rate. Additionally, the best performance that can be achieved with a code length of 63 is also close to the information-theoretic bound on performance. Consequently, as long as the code rate is properly chosen, a codelength of 63 is sufficient in providing good performance. This is particularly interesting since it indicates that codes of relatively small codelength (the FEC decoding complexity as well as the net FEC delay introduced depend on the codelength) are sufficient to yield good performance.

Though the optimized code was selected here for a particular value of the Markov chain parameters, such an accurate description of the loss behavior in the network is seldom available. As a result, it is important to consider the behavior of the optimized codes for slightly different values of the Markov chain parameters so that their behavior can be studied under mismatched conditions. For example, consider the behavior of the network when there is a temporary overload. In that case, the Markov model parameters could fluctuate from their long-term average values. This behavior is illustrated in the figure for two cases, one of which was selected with a relatively strict 5 msec constraint on the delay and the other a more relaxed 20 msec constraint. The 5 msec constraint limits the number of available codes that meet the threshold on the loss. As a result, much lower code rates are required to achieve good performance. Even under mismatched conditions, observe that the optimized codes perform very well indicating the robustness of the code selection strategy.

Figures 12 and 13 demonstrate the same for the case where the loss probability P_L and ρ_{LL} are different. Figure 12 illustrates the case where $P_L = 1 \times 10^{-2}$ and $\rho_{LL} = 0.25$ while Fig. 13 illustrates the case where $P_L = 5 \times 10^{-3}$ and $\rho_{LL} = 0.40$. The optimizing code for the example in Fig. 12 is RS(116,106) whose code rate is 0.91. For the example in Fig. 13, the optimizing code is RS(116,108) with a code rate of 0.93. As might be expected, the distortion is higher for the example in Fig. 12 compared to that in Fig. 13 due to the fact that the loss probability is higher. Due to the very same reason, the optimizing code rate for the example in Fig. 12 is lower than that for the example in Fig. 13. In each case, notice that a similar trend as in Fig. 11 is seen in terms of performance. More specifically, notice that a codelength of 63 performs close to the optimized code provided the code rate R is properly chosen. Even under slight deviation from the channel parameters for which the codes were selected, the optimized codes yield robust performance. Furthermore, the performance was far better than the RS(128,124) code.

We now illustrate in Fig's 14 and 15 a set of images from a particular run. In this example, the loss probability is $P_L = 1 \times 10^{-2}$ while $\rho_{LL} = 0.25$ and the sequence used is the *Football* sequence. The overall transmission rate is 0.84 bits/pixel. Here, the distortion is due to the loss in the particular frame as well as the cumulative effects of the loss in the previous 8 frames.

To begin, Fig. 14.a depicts the original frame while Fig. 14.b shows the reconstructed image under conditions of no packet loss for the unprotected scheme. Figure 14.c illustrates effects of loss for the unprotected scheme. Notice the distortion seen around the jerseys of the players in the image. Due to the relatively high motion present in the sequence, the passive concealment fails in

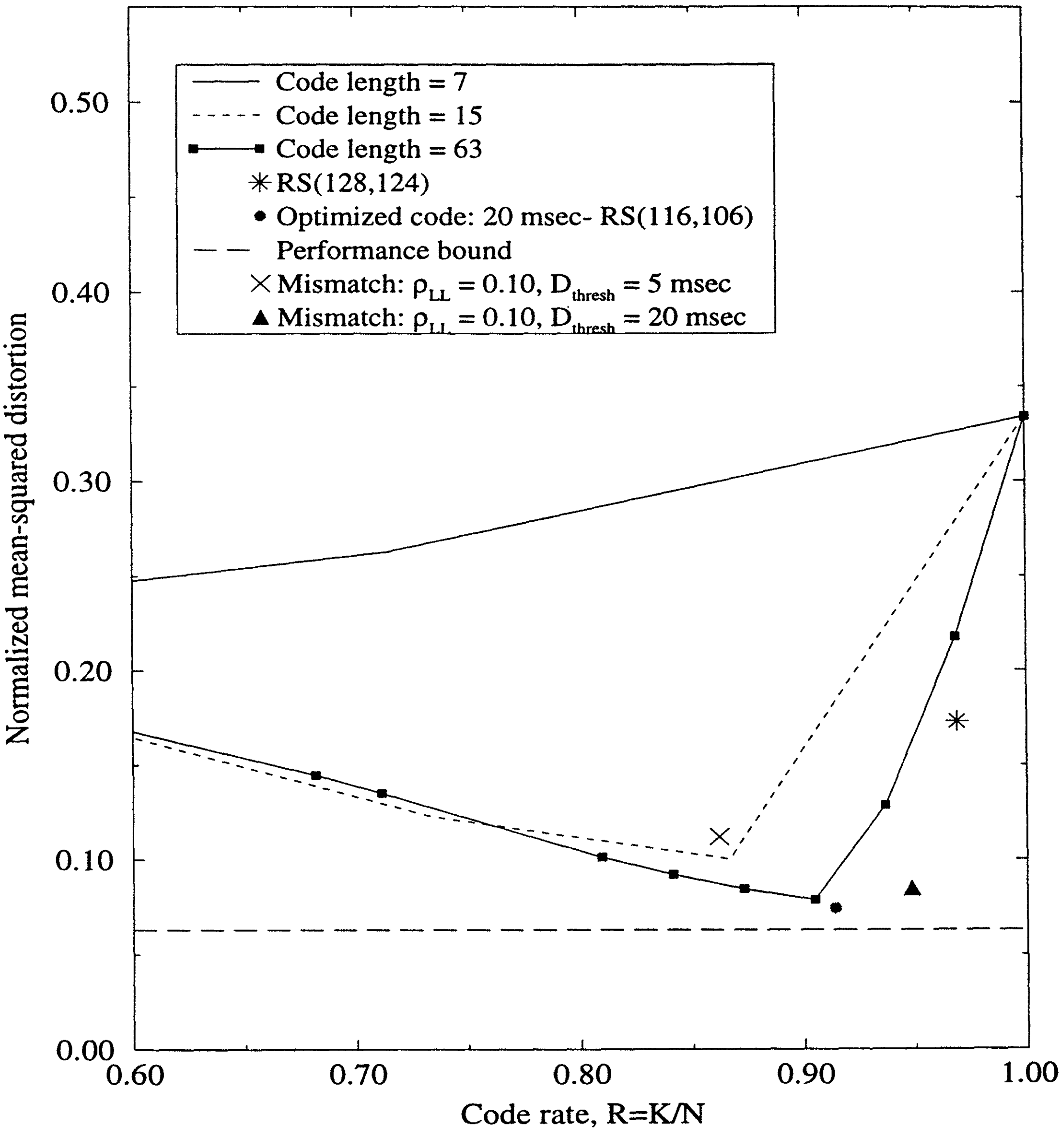


Figure 12: Comparative performance of a hybrid error concealment scheme at different code rates. Loss probability $P_L = 1 \times 10^{-2}$ for both priority classes, $\rho_{LL} = 0.40$, sequence is the *Football* sequence, $N_f = 12$ and the overall average transmission rate is 0.85 bits/pixel.

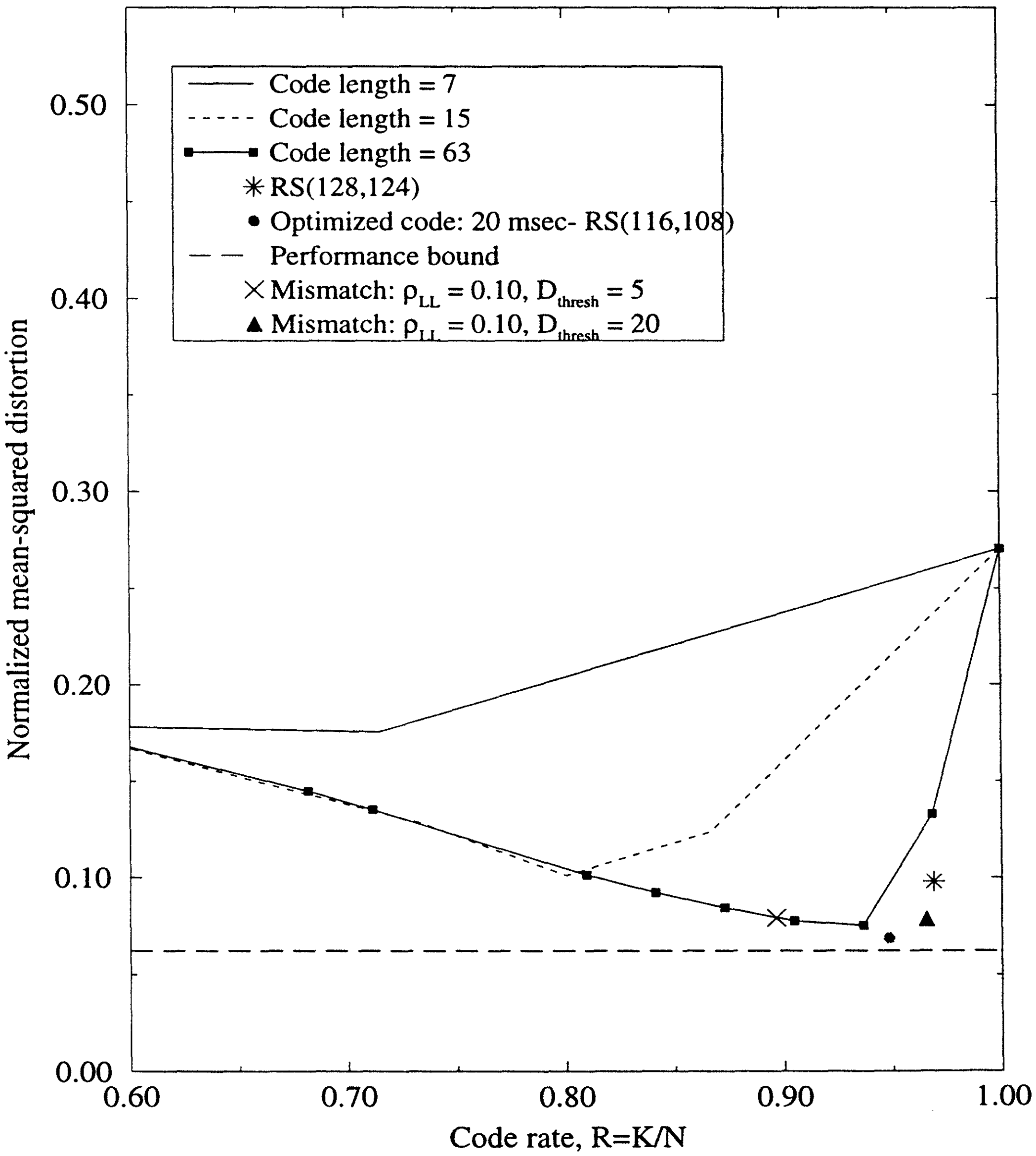
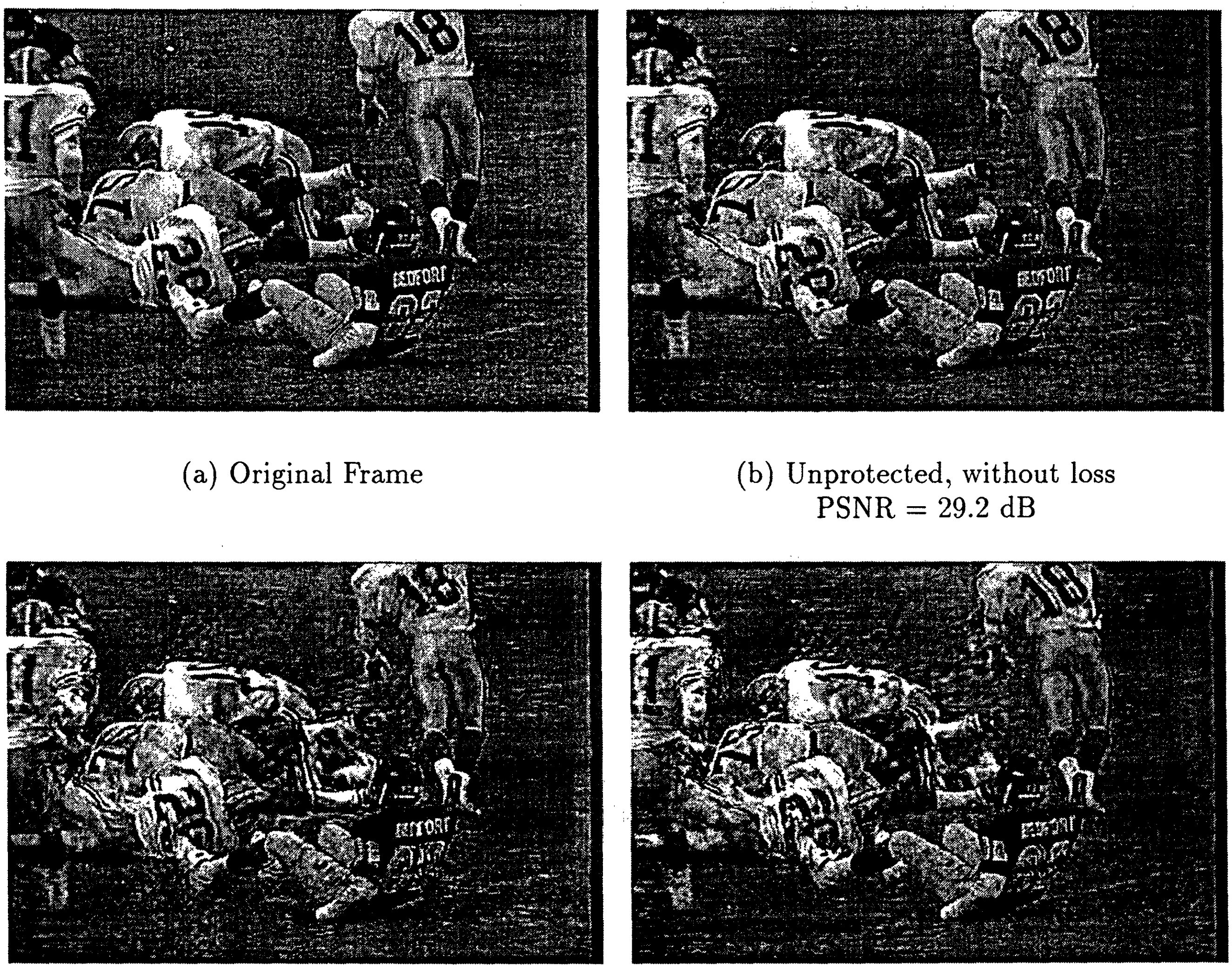


Figure 13: Comparative performance of a hybrid error concealment scheme at different code rates. Loss probability $P_L = 5 \times 10^{-3}$ for both priority classes, $\rho_{LL} = 0.40$, sequence is the *Football* sequence, $N_f = 12$ and the overall average transmission rate is 0.85 bits/pixel.



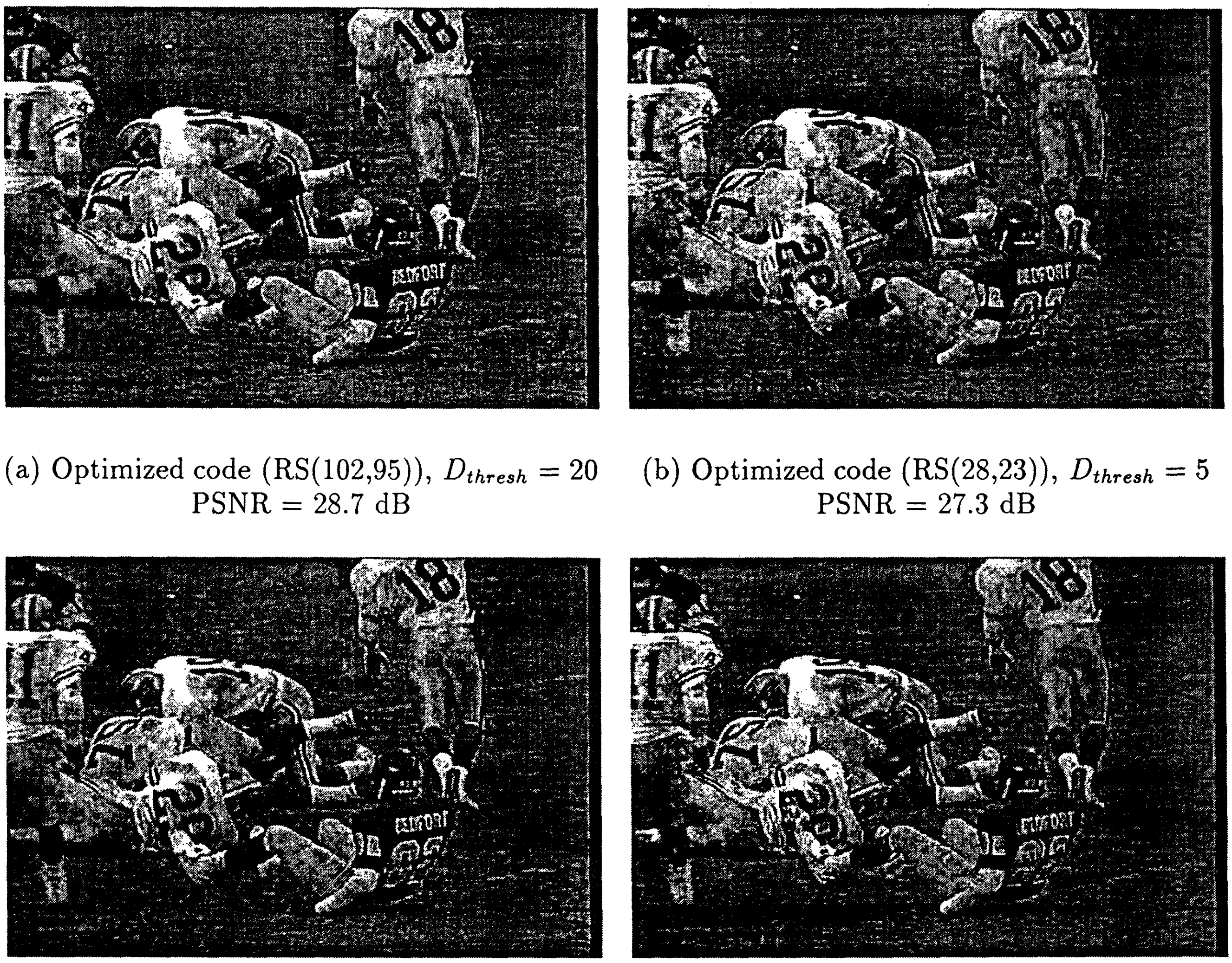
(a) Original Frame

(b) Unprotected, without loss
PSNR = 29.2 dB

(c) Unprotected, under loss
PSNR = 22.1 dB

(d) With RS(15,14)
PSNR = 23.4 dB

Figure 14: Comparison of images in the 9^{th} frame over a run of $N_f = 12$ frames under packet loss. Both high priority loss and low priority loss are 1×10^{-2}, loss correlation $\rho_{LL} = 0.25$, overall rate = 0.84 bits/pixel and interleaving depth = 1 in all cases.



(a) Optimized code (RS(102,95)), $D_{thresh} = 20$
PSNR = 28.7 dB

(b) Optimized code (RS(28,23)), $D_{thresh} = 5$
PSNR = 27.3 dB

(c) With RS(63,57)
PSNR = 28.3 dB

(d) With RS(128,124)
PSNR = 26.6 dB

Figure 15: Illustration of reconstructed images using other codes for the same conditions as in Figure 14.

providing adequate protection at such loss probabilities. To emphasize the requirement for a good code selection strategy notice that the quality resulting from using the RS(15,14) code (which is chosen arbitrarily in this example) shown in Fig. 14.d demonstrates the poor protection provided by this particular code. Furthermore, apart from the poor protection, transmission bandwidth is wasted due to the throttling quite apart from introducing FEC coding delays. By employing an optimally selected code, far better performance is achieved as seen in Fig. 15.a and Fig. 15.b. These two codes are selected using two different thresholds on the the delay, one a 20 msec. threshold and the other a 5 msec. threshold. Not surprisingly, the code which satisfies the 5 msec bound has a lower code rate and consequently lower objective quality as seen in Fig. 15.b. The reconstructed images in both Fig. 15.a and Fig. 15.b are far better than those in either Fig. 14.c or Fig. 14.d. Figure 15.c depicts the performance obtained by using the RS(63,57) code. Notice that the RS(63,57) code performs nearly as well as the optimum RS(102,95) code. The objective quality in this case is only slightly lower. It is difficult to perceptually distinguish the difference in quality between Fig. 15.a and Fig. 15.c. Normally, it is difficult to distinguish between two images when their SNR's differ slightly as the perceptual visual response works on a threshold basis. More importantly, the delay and complexity in employing the RS(63,57) code is far lower than for the RS(102,95) code. Finally, we show the performance provided by the RS(128,124) code for the same frame in the sequence as depicted by Fig. 15.d. The RS(128,124) code does not provide as much protection as the codes illustrated earlier in this example. In fact, codes of smaller code length with only a slightly lower code rate can be used to provide good performance. These results become even more pronounced under situations of scene change in the video source material as well as in cases where there are a large number of frames to which the error can propogate.

V.3. Extension to multiple priority classes

Normally, in transmission in the presence of bit errors, an *unequal* error protection scheme is adopted to provide more protection for the information which is relatively more sensitive to loss and less protection to information which is relatively less sensitive to loss. Typically, by redistributing the bits allocated for the channel coding operation by first classifying the information into more and less important and subsequently applying an appropriate code in accordance with the importance of the information, improved performance can be achieved. This is in contrast to a system using a single code on all classes.

The discussion above also extends to the ATM environment where up to two priority classes are allowed by the ATM specification. Consider the example shown in Fig. 16 where the loss probability on the high-priority class is 5×10^{-3} and on the low-priority class is 1×10^{-2}, both with $\rho_{LL} = 0.25$. In this case, a different code is used for each priority class optimized independently for that class in accordance with the respective values of the loss parameters. The data from encoding the first 4 subbands is packetized and assigned the high-priority class while that from 5-16 is assigned the low-priority class. The codelengths illustrated ($N = 7, 15, 63$) represent the performance when the corresponding code is applied to both priority classes. The optimized code for the high-priority class is RS(61,56) and for the low-priority class is RS(56,50). Though it seems that the code rate which maximizes the performance for the codelength of 63 does better than the optimizing code, it should be noted that the particular code does not satisfy the delay threshold of 5 msec. in this particular example. Notice that the results in the example illustrated in Fig. 16 are similar to those in Fig's 11-13. Therefore the conclusions for the examples in Fig's 11-13 apply here as well.

As described earlier, typically in an environment with bit errors, a more powerful code is applied

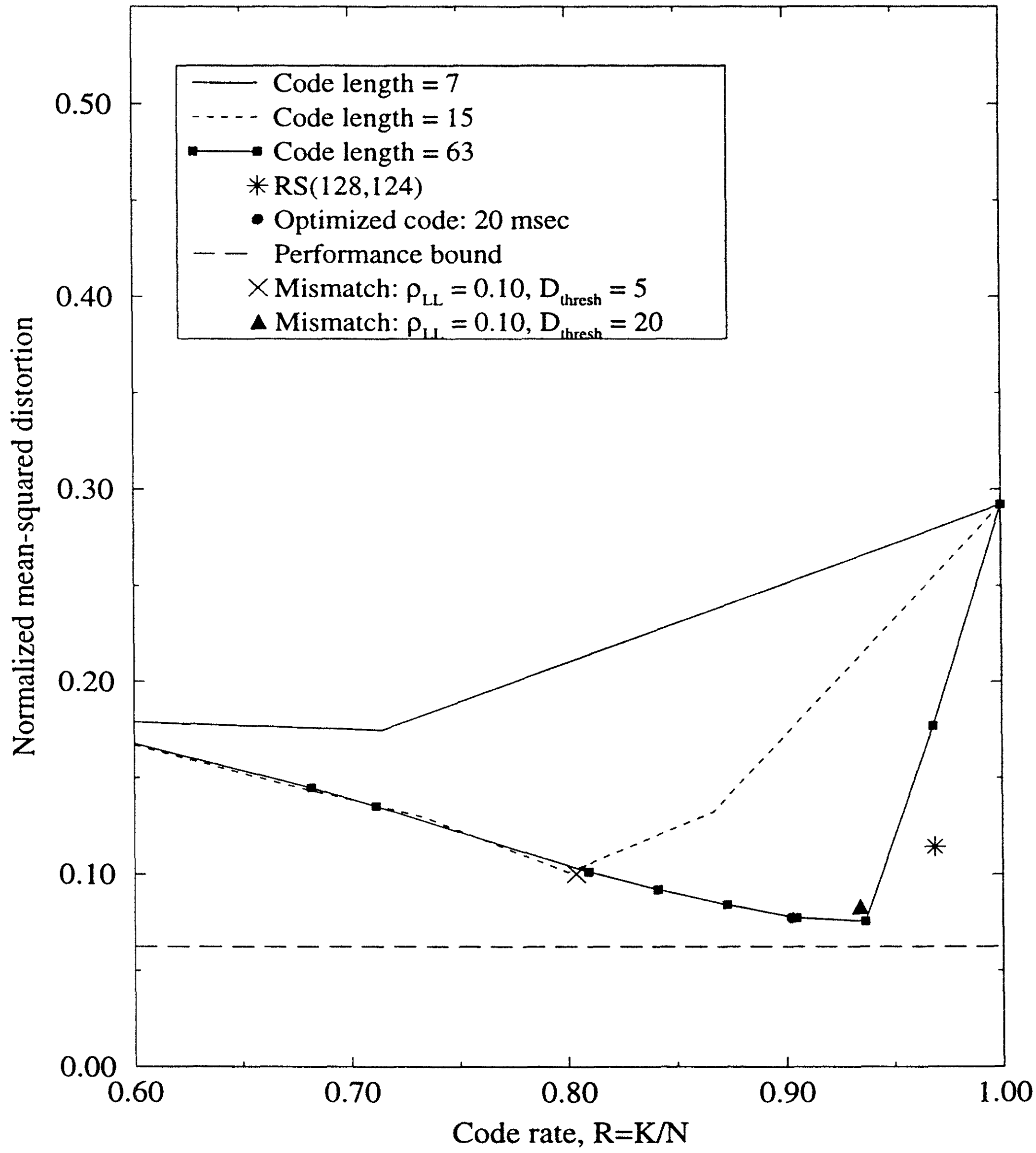


Figure 16: Comparative performance of a hybrid error concealment scheme at different code rates. Loss probability $P_L = 5 \times 10^{-3}$ for the high-priority class, 1×10^{-2} for low-priority class, $\rho_{LL} = 0.40$, sequence is the *Football* sequence, $N_f = 12$ and the overall average transmission rate is 0.85 bits/pixel.

on the more valuable information thereby spending more channel coding bits on it. This situation is true when both the classes undergo the same channel degradation (either in terms of bit-errors or loss probability). However, such an allocation is questionable in an environment like the ATM network in situations when there is a constraint on the code rate. This is due to the fact that typically the high-priority information has much lower loss than the low-priority information. The ensuing discussion will elaborate on this effect in a more detailed manner. The previous examples (Fig.'s 11-16) were unconstrained in regards to the fraction of the overall rate allocated to FEC application. In other words, an unconstrained allocation has the drawback that there could potentially be too much sacrifice in transmission quality due to the throttling when there are few or no losses. Therefore, when there is a constraint on the overall code rate, we investigate whether an unequal application of FEC (applying a less powerful code on the low-priority information and vice-versa on the high-priority information) proves beneficial as it may potentially be insufficient to recover the missing packets.

Figure 17 illustrates an example to demonstrate this situation again using the *Football* sequence. In this example the fraction of the overall rate allocated to the channel coding operation is no more than 10%. The total transmission rate is once again 0.85 bits/pixel. Let the fraction of the 10% of channel coding bits spent on the high-priority information be β. So the fraction spent on the low-priority information is $1 - \beta$. A code is selected for each class based on the condition (similar to that in (2)) that

$$\begin{array}{ll} \text{minimize:} & \text{Decoded loss probability} \\ \text{subject to:} & \frac{K}{N} \geq R_{threshold} \\ & \text{FEC coding delay} \leq D_{threshold}. \end{array} \tag{6}$$

Here $R_{threshold}$ represents the constraint on the code rate. For the particular example, $R_{threshold} = .90$ and $D_{threshold} = 20$ msec. This particular formulation is chosen rather than (2) since, given that the fraction spent on channel coding is limited by $R_{threshold}$, it can be split among the two-priority classes in any desired manner. Two examples are chosen for the channel parameters. In one case, the high-priority loss is an order of magnitude lower than the low-priority loss. The values chosen for the low and high-priority loss probability are 1×10^{-2} and 1×10^{-3}, respectively. Such a situation is quite normal since the very idea behind splitting into priorities is to reduce the loss on the high-priority by making use of appropriate buffer management policies. In another example shown in the same figure, there is a lower difference between the high and low-priority loss probability where the values are taken as 1×10^{-2} and 5×10^{-3}, respectively. Also shown in the figure are the respective information-theoretic performance bounds for the two examples. In both examples, the codes which maximize the performance perform reasonably close to the bound. Notice from Fig. 17 that, as would be normally expected, the performance is better for the example in which the high-priority loss is lower. When the high-priority loss is 5×10^{-3}, the distortion is minimized when the total bits allocated for FEC are split nearly equally between the high-priority and low-priority class. However, when the high-priority loss is reduced to 1×10^{-3}, the best performance is achieved when more bits are spent on the low-priority class. This result indicates that the best allocation of FEC between the high and low-priority class is when a higher fraction of bits is spent on the low-priority class due to the loss characteristics of each priority class. Though this seems counter-intuitive, it may be accounted for as follows. As the loss probability on the high-priority class improves (or is far lower than that on the low-priority class), less protection is required. Therefore to optimize overall performance, a greater percentage of the overhead is allocated to protect the low-priority class.

In the next section, we summarize the main results and provide some direction for future re-

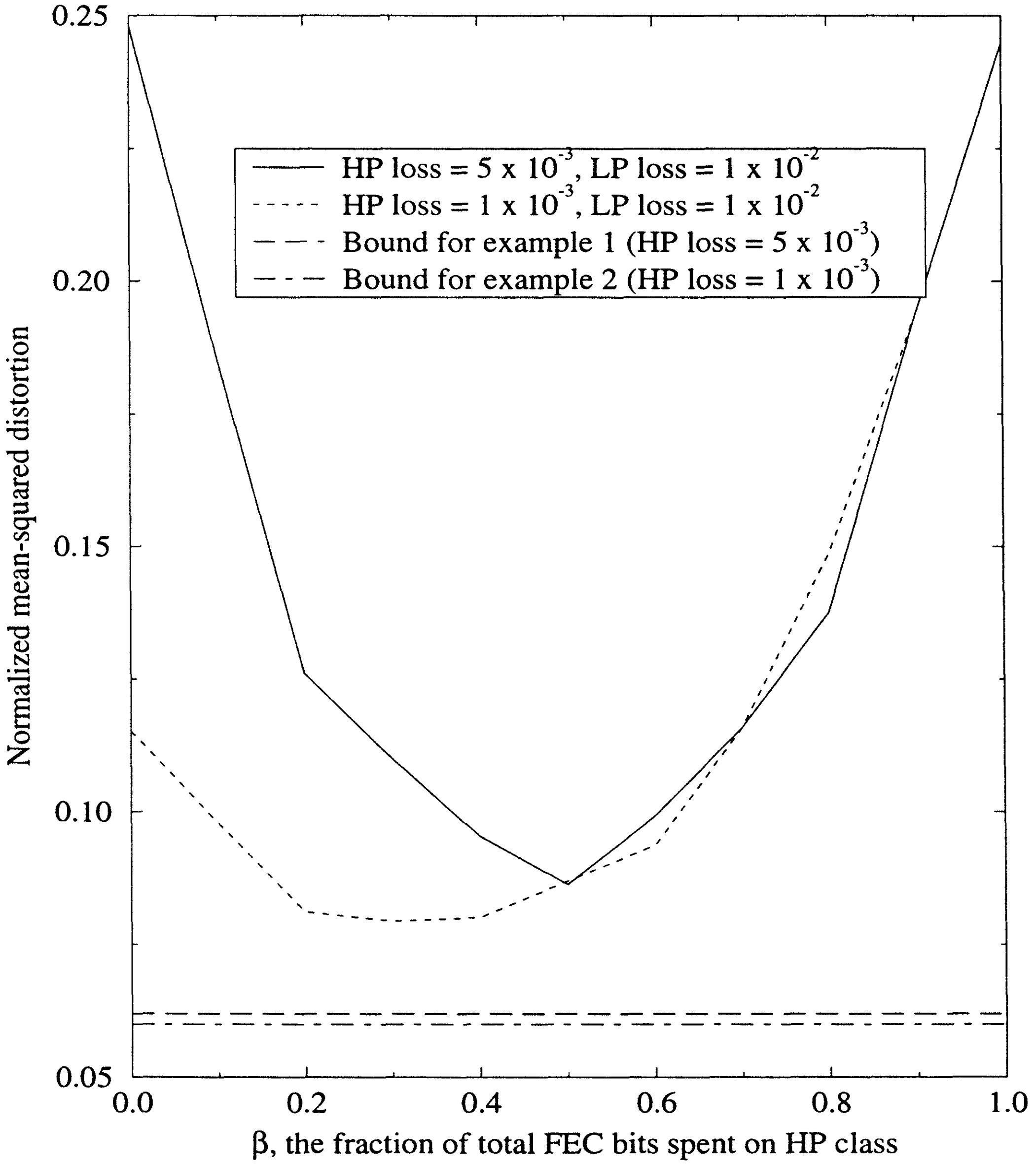


Figure 17: Performance as a function of the fraction of total bits allocated to the high-priority class. $\rho_{LL} = 0.40$, $R_{threshold} = 0.90$, $N_f = 12$, sequence is the *Football* sequence and the overall average transmission rate is 0.85 bits/pixel.

search.

VI. Conclusions and Future Directions

In this paper, various aspects of an FEC-based transport scheme for packetized video have been considered. First, issues associated with code selection were studied. The performance of the scheme is closely related to the code selected. As a result, it is important to devise a clever code selection strategy. Due to the huge parameter space and the lack of tractability, traditional methods of optimizing the code selection fail. A simple strategy of selecting codes based on a constrained optimization technique was outlined and its performance studied. The results indicate that the selection of a single code for all operating rates is questionable. Rather, it would be better to pre-store a fixed group of good codes, for a selected range of operating parameters. At low operating rates, the use of FEC may be accompanied by excessive throttling which would translate to excessive sacrifice in coding quality to accommodate the FEC bandwidth expansion. For most cases, codes of small-to-moderate codelength (≤ 63) performed very well as long as the code rates were properly chosen. The code selection strategy provided robust performance even under conditions of mismatch in choice of the channel parameters for which the codes were selected. It would be interesting to study the performance of this scheme for MPEG coded video. This is currently under study and the corresponding results will be presented in future work.

A possible direction of future research would be to study the application of FEC taking into account the network feedback. It would be interesting to study the performance of the code selection strategy in such a case where the goal would be to optimize the end-to-end system performance.

Another related extension could be to attempt a more rigorous analysis to quantify some of this work and use it in designing a code selection strategy. This would relieve some of the ad hoc techniques used in this work. In this regard it would be interesting to see how such a code selection strategy would compare with the one used in this work. However, such a method is at present unclear.

Appendix A
Evaluation of the FEC Coding Delay

Let the frame rate in frames per second be denoted by f and the packets generated in a frame be p_{ave}. Consider first the case when there is no interleaving. Therefore the FEC delay introduced is due to the interlacing application alone. As discussed in Section III.3, due to the systematic nature of RS codes, the interlacing delay can be avoided at the transmitting end by suitably copying the cells for FEC application prior to transmission. Then the FEC delay introduced is due to the delay at the receiving end alone.

The mean interarrival time of the packets is $\frac{1}{f.p_{ave}}$ seconds assuming uniform interarrival of packets within a frame. For the RS(N, K) code, the interlacing delay D (in seconds) in accumulating N packets at the receiver is then

$$D = N \times \frac{1}{f.p_{ave}} \tag{A.1}$$

Consider now the case with interleaving. In this case there is delay at both the transmitting and receiving site. Suppose the interleaving depth were given by M. The delay in accumulating MN packets is then given by

$$D = MN \times \frac{1}{f.p_{ave}} \tag{A.2}$$

Since the delay now is both at the transmitting end as well as the receiving end, the total delay due to both the interlacing and interleaving operations is given by

$$D = 2 \times MN \times \frac{1}{f.p_{ave}} \tag{A.3}$$

Now p_{ave} can easily be related to the coder operating rate and the image dimension if the packet size is fixed, as in the case of ATM cells. The operating rate in bits/pixel multiplied by the frame resolution in pixels gives the total number of bits generated in a frame. p_{ave} is then determined by dividing the total number of bits generated by cell size which in our work is 384 bits.

Appendix B

Evaluation of the Decoded Packet Loss Probability

Here we provide a manner of calculating the decoded packet loss probability P_{dec} for the correlated channel model. For a (N, K) code with a erasure correcting ability of e packets, losses only occur when more than e of N packets are lost. So the decoded loss probability is given by

$$P_{dec} = \sum_{k=e+1}^{N} \frac{kp_k}{N}, \tag{B.1}$$

where p_k is the probability of having k losses in N packets. For the case where losses occur independently (or the interleaving is ideal)

$$P_{dec} = \frac{Np - \sum_{k=0}^{e} \begin{pmatrix} N \\ k \end{pmatrix} p^k(1-p)^k}{N} = \frac{\sum_{k=e+1}^{N} \begin{pmatrix} N \\ k \end{pmatrix} p^k(1-p)^k}{N}, \tag{B.2}$$

where p is the probability of packet loss.

However, when losses are correlated, then the calculation of p_k is not so straightforward. Our technique of computing p_k is as follows. Suppose, without loss of generality, the initial state of the Markov chain is the *loss* state. Also, for purposes of illustration, let the codelength be 3. Our objective then is to find $p_k, 0 \leq k \leq 3$. Then all the subsequently possible transitions can be outlined in the form of a tree. Every node of the tree would either be a *loss* state or a *transmit* state. Since the code length is 3, there would a total of $2^3 = 8$ nodes in the final stage. The probability of traversing a node would depend on the transition probability of the Markov chain. For example, the probability of moving from a *loss* state to another *loss* state would correspond to the one-step transition probability ρ_{LL} in Fig. 7. Now p_0 is the probability of hitting 3 *transmit* stages in traversing from the top to the bottom nodes. Then by multiplying the corresponding path probabilities p_0 can be calculated. Similarly, other $p_k's$ can be calculated. With a knowledge of the $p_k's$, P_{dec} can be calculated through (B.1).

Now the complexity of this computation increases with codelength N. To drastically reduce the complexity, more intelligent computation can be performed by observing that that at any stage of the tree there are only two possible outcomes, namely *loss* and *transmit*. Therefore many of the paths of the tree can be merged together, but only after adding the path weights which in our case is the transition probabilities. Furthermore, observe that due to practical constraints in terms of delay and complexity, the codelength N would be also restricted to relatively moderate lengths therefore reducing the complexity of this technique. The implementation of the algorithm works quite fast for most cases. In [23], further details and verification of this computation are provided. Other methods of computing (B.1) are possible such as the use of simulations.

REFERENCES

[1] N. Shachum and P. McKenney, "Packet Recovery in High-Speed Networks using Coding and Buffer Management," *Proc. IEEE INFOCOM*, San Fran., CA, pp. 124–131, June 1990.

[2] A.M. McAuley, "Reliable Broadband Communication using a Burst Erasure Correcting Code," *Proc. ACM SIGCOMM* , Philadelphia, PA, pp. 297–306, Sept. 1990.

[3] R. Kishimoto and K. Irie, "HDTV Transmission in an ATM-based Network," *Image Communication*, vol. 3, pp. 111–122, June 1991.

[4] L. Zhang, "Statistics of Cell Loss and its Application for Forward Error Recovery in ATM Networks," *Proc. IEEE GLOBECOM*, Orlando, FL, pp. 694–698, Dec. 1992.

[5] E.W. Biersack, "Performance Evaluation of Forward Error Correction in ATM Networks," *Proc. ACM SIGCOMM*, Baltimore, MD, pp. 248–257, Aug. 1992.

[6] H. Ohta and T. Kitami, "A Cell Loss Recovery Method using FEC in ATM Networks," *IEEE J. Select Areas Commun.*, vol. 9, no. 9, pp. 1471–1483, Dec. 1991.

[7] C.Partridge, *Gigabit Networking*, Addison Wesley Publishing Company, 1993.

[8] A. S. Acampora, *An Introduction to Broadband Networks*, Plenum Press, 1994.

[9] V. Parthasarathy, J. W. Modestino and K. S. Vastola, "Reliable Transmission of High-Quality Video over ATM Networks," *submitted to IEEE Trans. Image Processing.*

[10] V.Parthasarathy, J.W.Modestino and K.S.Vastola, "Design of a Transport Protocol for Variable-rate Video Transmission over ATM Networks," *International Workshop on Packet Video*, Portland, 1994.

[11] Y.H.Kim and J.W.Modestino, "Adaptive Entropy Coded Subband Coding of Images," *IEEE Trans. Image Processing*, vol. IP-1, pp. 31-48, Jan 1992.

[12] Y.H.Kim, "Adaptive Entropy coded Predictive Vector Quantization of Images," Ph.D Dissertation, Dept. of Electrical Eng., Rensselaer Polytechnic Institute, Troy, NY-12180, 1990.

[13] Y.H.Kim and J.W.Modestino, "Adaptive Entropy-Coded Subband Coding of Image Sequences," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 975-987, June 1993.

[14] G. Karlsson and M. Vetterli, "Packet Video and its Integration into the Network Architecture," *IEEE J. Select Areas Commun.*, vol. SAC, no. 7, pp. 739-751, June 1989.

[15] M.Ghanbari and C.J.Hughes, "Packing Coded Video Signals into ATM Cells," *IEEE/ACM Trans. Networking*, vol. 1, no. 5, pp. 505-509, Oct. 1993.

[16] R.F.Rey, editor, *Engineering and Operations in the Bell System*, AT&T Bell Labs, Murray Hill, NJ, 2nd edition, 1983.

[17] A.Baiocchi, N.B.Melazzi, A.Roveri and R.Winkler, "Loss Performance Analysis of an ATM Multiplexer Loaded with High-Speed On-Off Sources," *IEEE J. Select Areas Commun.*, vol. 9, no. 3, pp. 388-393, April 1991.

[18] A.M.Michelson and A.H.Levesque, *Error Control Techniques for Digital Communication*, John Wiley, NY, 1985.

[19] R.E.Blahut, *Theory and Practice of Error Control Codes*, Addison Wesley Publishing Company, 1983.

[20] S.Lin and D.J.Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, N.J, 1983.

[21] R.Nagarajan and J.F.Kurose, "On Defining, Computing and Guaranteeing Quality-of-Service in High-Speed Networks," *Proc. IEEE INFOCOM*, Florence, Italy, pp. 2016–2025, April 1992.

[22] R.J.McEliece and W.E.Stark, "Channels with Block Interference," *IEEE Trans. on Inform. Theory*, Vol. IT-30, pp. 44-53, Jan 1984.

[23] V.Parthasarathy, "Transport Protocols for Multi-Resolution Video Transmission over ATM Networks," Ph.D Dissertation, Dept. of Electrical Eng., Rensselaer Polytechnic Institute, Troy, NY-12180, 1995.