On Transmission of JPEG2000 Codestreams Over Packet Erasure Channels

Ali Bilgin, Zhenyu Wu and Michael W. Marcellin

Dept. of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721

ABSTRACT

An unequal loss protection framework for transmission of JPEG2000 codestreams over packet erasure channels is presented. A joint source-channel coding approach is adopted to form the JPEG2000 codestream and assign the appropriate amount of protection to different sections of the codestream. Experimental results indicate that the proposed scheme yields excellent performance across a wide range of packet loss rates.

Keywords: Unequal Loss Protection, JPEG2000, packet-erasure channels, Reed-Solomon codes

1. INTRODUCTION

When best-effort networks, such as those based on the Internet Protocol (IP), experience congestion, some of the transmitted packets are dropped. In some cases, the lost data can be recovered using a packet recovery mechanism, such as Automatic Repeat Request (ARQ). However, in real-time applications, the retransmitted packets may arrive after they are needed by the application. Furthermore, retransmission requests aggravate the congestion. There are also applications where the receiver does not have a feedback channel to communicate with the sender. Thus, it is very desirable that the application be resilient to such losses. This becomes especially important when the data transmitted over the network has been compressed prior to transmission. Since compression aims to remove the redundancy in the data, the recovery task becomes more challenging.

High performance image compression systems produce codestreams that exhibit little redundancy. Thus, such coders are highly sensitive to errors in the codestream and a single bit error can result in disposal of the entire codestream. When such codestreams are transmitted over packet erasure channels, parts of the codestream might be unavailable at the decoder due to packet losses. The decoder should then be able to reconstruct the image using the available packets.

Besides ARQ techniques, several different strategies have been proposed for combating packet erasures. One such strategy is to partition the bitstream into small segments that can be decompressed independently [1–3]. In [1], the author proposed a method that partitions wavelet coefficient trees into a fixed number of groups and compresses each group independently. This allows independent decompression of a group, and prevents errors from propagating beyond group boundaries. A similar method called Packetizable Zerotree Wavelet (PZW) was proposed in [2]. In this method, complete trees of wavelet coefficients were contained within a packet. Thus, each packet could be decompressed independently. The problem of minimizing packetization inefficiency due to bitstream alignment was addressed in [3]. The authors developed a theoretically optimal packetization scheme, and provide low-complexity suboptimal methods that achieve comparable performance.

Although, the above techniques provide adequate resilience against packet losses for low packet loss rates, their performance suffers when the network experiences high packet loss rates. In these cases, Forward Error Correction (FEC) techniques that introduce controlled redundancy into the bitstream become attractive. This redundancy is exploited at the decoder to correct some of the transmission errors. The FEC strategies applied in the packet loss case can be broadly classified into two: Equal Loss Protection (ELP) and Unequal Loss Protection (ULP) strategies. The ELP strategies assign an equal amount of FEC protection to the

Further author information: (Send correspondence to A.B.) A.B.: bilgin@ieee.org, Z.W. zywu@ece.arizona.edu, M.W.M.: marcellin@ece.arizona.edu

Image and Video Communications and Processing 2003, Bhaskaran Vasudev, T. Russell Hsing, Andrew G. Tescher, Touradj Ebrahimi, Editors, Proceedings of SPIE–IS&T Electronic Imaging, SPIE Vol. 5022 (2003)

entire codestream, while the ULP strategies assign varying amounts of protection to different sections of the bitstream, depending on their importance. The ELP schemes are designed for operation on a specific channel. If the channel conditions deteriorate, the amount of protection provided by the ELP scheme can become insufficient and the entire codestream can be lost. In contrast, the performance of the ULP schemes degrades gracefully with increasing packet loss. The Priority Encoding Transmission (PET) algorithm presented in [4] is an ULP scheme that assigns varying amounts of FEC to different message fragments based on priorities specified by the user. Similarly, in [5] , Mohr et. al. presented a ULP framework in which unequal amounts of FEC were applied to progressively compressed data.

In this paper, we develop efficient techniques for transmission of JPEG2000 compressed codestreams over packet switched networks. In our earlier work [6] , we demonstrated that a carefully designed packetization strategy can provide significant performance improvements. In [6] , the packetization strategy was developed under the assumption that the packetization algorithm only had access to the JPEG2000 codestream. In other words, the source coding was already performed and a codestream was created. The problem then was to packetize this codestream for efficient transmission. The packetization algorithm was developed considering the properties of the JPEG2000 codestream, and FEC techniques were utilized to achieve further resilience. In contrast, we adopt a joint source-channel coding (JSCC) approach in this work. We assume that the packetization algorithm has access to the JPEG2000 encoder, and can control the formation of the JPEG2000 codestream. We develop a ULP strategy by allowing different amounts of protection for different sections of the codestream.

This paper is organized as follows: In the next section, we provide a brief overview of the error resilience properties of JPEG2000. In Section 3, the proposed packetization strategy is introduced. Experimental results and comparisons with other methods from the literature are presented in Section 4, and a brief summary of the paper is given in Section 5.

2. ERROR RESILIENCE OF JPEG2000 CODESTREAMS

JPEG2000 is the latest international standard for image compression [7,8] . JPEG2000 provides state-of-the-art compression performance and functionality. A simple block diagram of a JPEG2000 encoder is illustrated in Figure 1. As illustrated in the figure, the input image is first tiled into non-overlapping rectangular tiles. If the image has multiple components, a component transform can be applied to each tile. Each *tile-component* is then wavelet transformed and quantized. JPEG2000 utilizes several geometric structures to enable low-memory implementations, and to facilitate random spatial access. These geometric structures also play a crucial role in error resilience of the codestream, as we will discuss later. The smallest of such structuctures are called *codeblocks*. Codeblocks are formed by partitioning each subband into non-overlapping rectangular regions. Codeblocks of particular resolutions are grouped together to form *precincts*, as illustrated in Figure 2. The bitplane coder operates independently on each codeblock. Thus, the compressed data from each codeblock can be regarded as an embedded bitstream. The bitplane coder makes three passes over each bitplane of a codeblock. These passes are referred to as *coding passes*.

The compressed data from the codeblocks within each precint are arranged together to form *packets* [*]. Each packet contains a header that describes the contributions of the codeblocks within the precinct to the packet. One packet from each precinct of each resolution of each tile-component form a *layer*. A layer can be regarded as a quality increment for the entire tile. The packets that belong to a particular tile are grouped together to form a *tile-stream*, and the tile-streams are grouped together to form the JPEG2000 codestream. These structures are illustrated in Figure 3.

The bitplane coder in JPEG2000 utilizes a context-based adaptive arithmetic coder. Although this arithmetic coder is very efficient, it is also very sensitive to errors that might occur in the codestream. A single bit

[*]It is important to note the distinction between the packets that are used in JPEG2000 codestreams and the packets that are used within the context of packet-switched networks. To avoid misunderstanding, we will refer to the packets used in networking as *network packets* for the remainder of this paper.

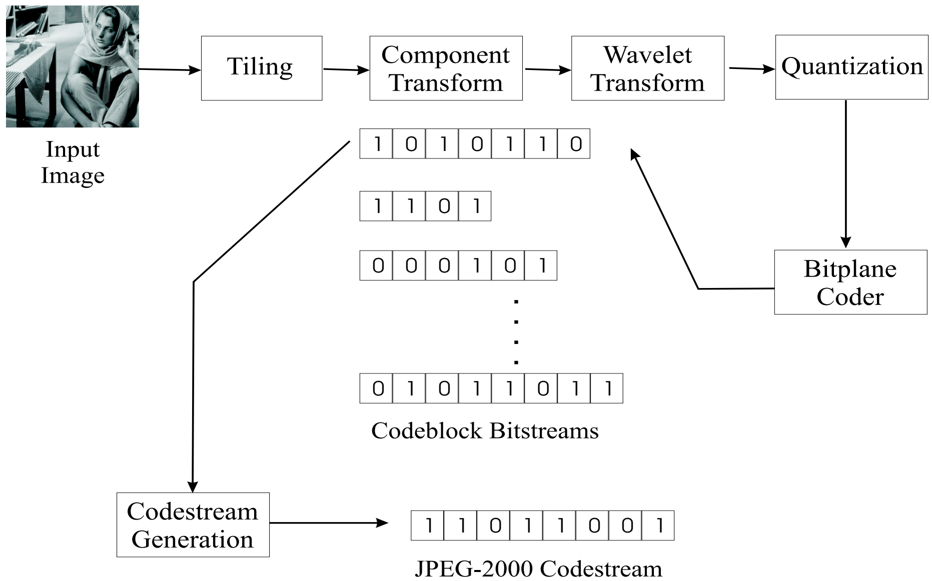

Figure 1. Block diagram of a JPEG2000 encoder.

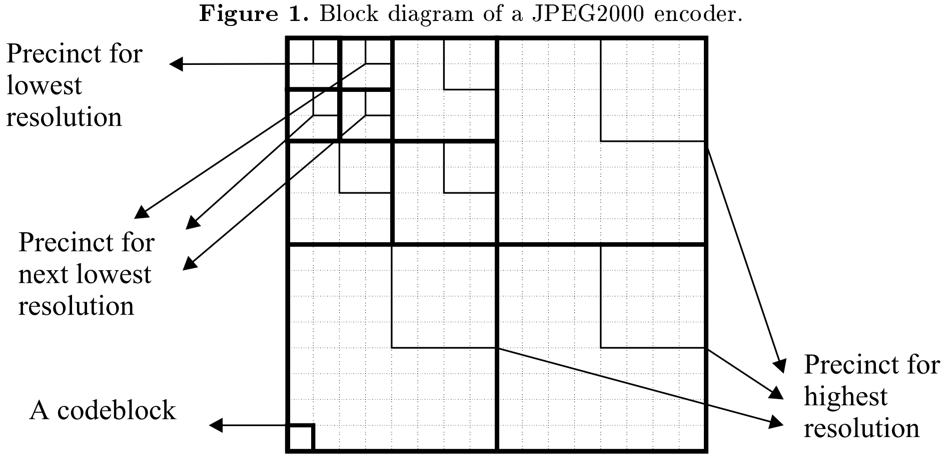

Figure 2. Partitioning of wavelet subbands for a tile-component.

error can result in loss of synchronization between the encoder and the decoder. To avoid such catastrophic scenarios, several error resilience mechanisms are provided by the standard. As illustrated in Figure 3, the JPEG2000 codestream consists of several partitions, and the data is organized in a hierarchical fashion. This organization allows isolation of errors and prevents them from propagating into other sections of the codestream. For example, if an error occurs in a codeblock codestream, this error is isolated within the codeblock and does not propagate to other codeblocks. Similarly, loss of a packet does not preclude decompression of packets that belong to other precints.

It should also be noted that JPEG2000 codestreams contain several header segments that are required for correct decompression. For example, the codestream starts with the main header. The main header contains critical information such as image and codeblock sizes, and is essential for correct decompression. Similarly, the tile-stream header can contain several parameters that are required for correct decompression of a tile-stream. Thus, it is important to ensure that the headers can be reconstructed at the decoder to avoid

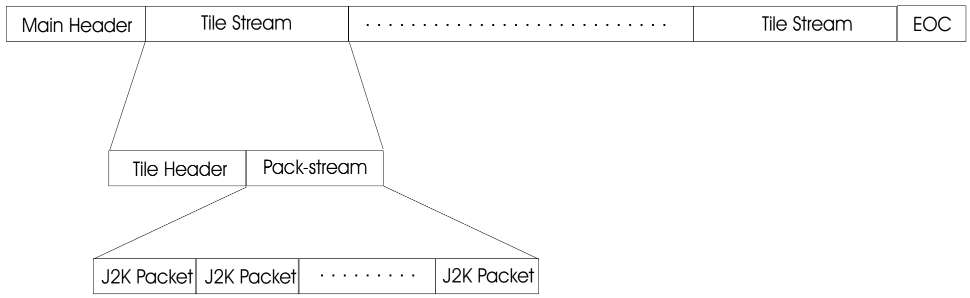

Figure 3. A simple JPEG2000 codestream.

catastrophic failures. JPEG2000 provides a mechanism where the packet headers can be removed from each packet, and stored in tile-part headers or the main header. Since the packet headers contain information about the contribution of each codeblock to the packet, this mechanism can be very valuable for error resilience.

In addition to the hierarchical organization of the codestream, JPEG2000 provides several tools that improve error resilience. For example, generation of the codestream using the ERTERM option, creates predictably terminated codewords. If the decoder can not identify this predictable termination at the end of the codeword, it can conclude that an error has occurred. JPEG2000 also uses byte-stuffing to avoid certain values inside codeblock codestreams. Unexpected detection of one of these values would also suggest that an error has occurred. The RESTART option can also provide significant advantages in error resilience [9] . When this option is employed, the arithmetic coder is restarted at the beginning of each coding pass.

For a comprehensive review of JPEG2000, the interested reader is referred to [7,8,10] .

3. PACKETIZATION OF JPEG2000 CODESTREAMS

The goal of packetization is to enable the decoder to reconstruct a high quality image, even when some of the network packets are lost during transmission. When a ULP strategy is employed for packetization, different parts of the codestream receive different amount of protection depending on their importance. Here, we consider the data that yields a greater distortion reduction for the final reconstructed image as more important. Since JPEG2000 provides mechanisms for ordering more important and less important data within the codestream (through the use of layers), it fits perfectly into a ULP framework.

In this work, we use JPEG2000 codestreams that contain multiple quality layers. A simple block diagram of the proposed scheme is presented in Figure 4 . As illustrated in the figure, each quality layer of the codestream is assigned the appropriate level of protection, and the channel-coded data are interleaved across all packets. As mentioned earlier, the header data is crucial for correct decompression of the codestream. Thus, the header is strongly protected to ensure that it will be error free at the decoder. The amount of protection applied to each level is stored together with the header data. The channel-coded data is interleaved in a continous fashion to minimize packetization inefficiency. Using the layer length information stored in the packed packet headers, together with the protection level information, the decoder is able to extract different layers. It should be noted that the formation of the layers and the selection of the appropriate protection levels for each layer are performed using a JSCC strategy, as described in the following section.

3.1. Joint Source-Channel Coding

In JSCC, rate allocation is used to distribute the available total rate between the source and the channel coders, in order to optimize some desired measure for a given channel condition. In this work, we use a two-level, hybrid-optimization rate allocation scheme that is based on [11] . The details of this optimization scheme are presented in [12] . Unlike [12] where rate compatible punctured convolutional codes (RCPC) and turbo codes (RCPT) were used over a Binary Symmetric Channel (BSC), we use Reed-Solomon (RS) codes which are a

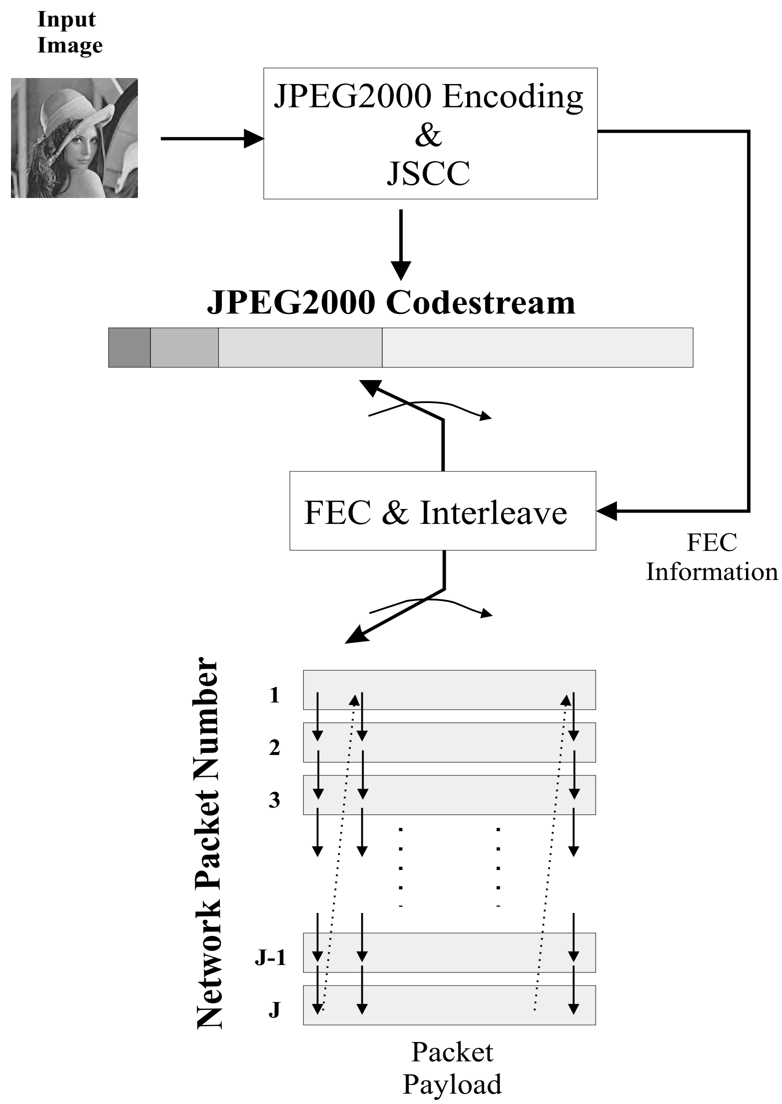

Figure 4. Illustration of the packetization method.

class of maximum distance separable codes [13] , and perform the optimization for a packet erasure channel. We adopt this strategy, since the RS codes are very effective against erasures. An (N, k) RS code can recover $N - k$ erasures, where N denotes the channel block length, and k denotes the number of source symbols. In this work, we set $N = 255$, and adjust k to achieve the desired level of protection for a specific erasure rate.

The goal of joint source-channel optimization is to find an optimal rate allocation scheme V, in order to minimize the expected distortion subject to a designated total rate. For JPEG2000 codestreams without channel noise, this is equivalent to minimizing the sum of distortions contributed from the codeblocks, while keeping the sum of their bitstream lengths within the designated total length:

$$\min \sum_b D_b \qquad \text{s.t.} \qquad \sum_b L_b \le L_T \tag{1}$$

where D_b and L_b are the distortion and bitstream length of codeblock b, respectively, and L_T is the designated file size [7, 8, 14] .

With channel noise, for each codeblock b, its expected distortion with some rate allocation scheme $\vec{V_b}$ can be written as:

$$E[D_b(\vec{V_b})] = D_{b,0} - E[\Delta D_b(\vec{V_b})] \tag{2}$$

where $D_{b,0}$ is the initial (zero rate) distortion, $E[\Delta D_b(\vec{V_b})]$ is the expected distortion reduction when $\vec{V_b}$ is employed, resulting in a length $L_b(\vec{V_b})$ bitstream for codeblock b.

With Eq. (2), in the noisy channel case, Eq. (1) can be rewritten as:

$$\min \sum_b (D_{b,0} - E[\Delta D_b(\vec{V_b})]) \quad \text{s.t.} \quad \sum_b L_b(\vec{V_b}) \leq L_T$$

or equivalently,

$$\min \sum_b -E[\Delta D_b(\vec{V_b})] \qquad \text{s.t.} \qquad \sum_b L_b(\vec{V_b}) \leq L_T \tag{3}$$

Adapting the results from [15] , we can solve Eq. (3) using Lagrangian multiplier method:

$$\min\{\sum_b -E[\Delta D_b(\vec{V_b})] + \lambda \sum_b L_b(\vec{V_b})\} \tag{4}$$

by minimizing each term (corresponding to each b) independently. By sweeping λ over the range of zero to infinity, sets of $\{V\}$ and $\{\sum_b L_b\}$ can be created. If a $(\sum_b L_b)$ happens to equal L_T, then a desired rate allocation scheme V has been found.

Minimizing each term in Eq. (4) corresponds to an optimization task at the coding pass level within a single codeblock. For codeblock b, denote N_c as the maximum number of coding passes that may be included. For each coding pass i, there is an associated distortion reduction d_i with length l_i bytes, where $i \in [1, N_c]$. From Eq. (4), for a given λ, it is desired to find a rate allocation scheme $\vec{V_b}$ which minimizes:

$$-E[\Delta D_b(\vec{V_b})] + \lambda L_b(\vec{V_b}) \tag{5}$$

Denote r_i as the channel code rate assigned for coding pass i $(0 \leq r_i \leq 1)$ and $P(r_i, \frac{l_i}{r_i})$ as the probability that there are one or more uncorrected errors in coding pass i if channel coding rate r_i is employed, and let $N_c^{'}$ $(N_c^{'} \leq N_c)$ be the number of coding passes included by $\vec{V_b}$. When decoding a JPEG2000 codestream with error resilience mode switches RESTART and ERTERM[8] , a JPEG2000 decoder can decode all correct coding passes within a given codeblock prior to the coding pass containing the first error. Decoding of other codeblocks is unaffected by such an error. With this condition, the expected distortion reduction for codeblock b including $N_c^{'}$ coding passes using channel code rates $r_1, r_2, \ldots, r_{N_c^{'}}$ is

$$\begin{aligned} E[\Delta D_b(\vec{V_b})] \quad = \quad & \sum_{j=1}^{N_c^{'}-1} (\sum_{k=1}^{j} d_k)(\prod_{k=1}^{j}[1 - P(r_k, \frac{l_k}{r_k})])P(r_{j+1}, \frac{l_{j+1}}{r_{j+1}}) \\ & +(\sum_{k=1}^{N_c^{'}} d_k)(\prod_{k=1}^{N_c^{'}}[1 - P(r_k, \frac{l_k}{r_k})]). \end{aligned} \tag{6}$$

Eq. (6) can be evaluated for any rate allocation scheme $\vec{V_b} = (r_1, r_2, \ldots, r_{N_c^{'}})$. Notice that $\vec{V_b}$ must jointly optimize $N_c^{'}$ and the channel code rates $r_1, r_2, \ldots, r_{N_c^{'}}$.

In [11] , an exhaustive search based scheme was proposed to find the optimal $\vec{V_b}$ for Eq. (6) over a reduced search space with $\binom{N_c+R-1}{R}$ candidates, where R is the number of available channel codes. But it becomes impractical to search when R is large. Thus, in [12] a dynamic-programming based optimization which aims to reduce the overall computational complexity is proposed. In this work, we utilize the algorithm described in [12] .

4. EXPERIMENTAL RESULTS

In this section, we present results of the experiments we have performed using the proposed method. We consider 53-byte ATM packets with payloads of 48 bytes. Experiments were repeated for 10000 realizations of each packet loss case. The 512 × 512 Lena image was used as a single-tile image for the experiments. We have used Kakadu [16] as our JPEG2000 codec. As mentioned earlier, packed packet headers were used to store the packet headers within the main header (referred to as PPM). The JPEG2000 codestreams were generated to contain three layers, and the contents of each layer, as well as the channel code used for each layer, were selected as described in the previous section. All presented results were obtained for 683 network packets (1.0005 bits/pixel). In these experiments, the main header was protected strongly with a carefully chosen RS code rate such that the probability of incorrect decompression of the main header is under 10^{-6}.

We first present the performance of the proposed scheme for different packet loss rates in Table 1. In the table, the proposed scheme is referred to as ULP-J2K. The table also contains results obtained using other methods in the literature. The Optimal Packetization (OP) algorithm [3] uses SPIHT and minimizes the packetization inefficiency due to bitstream alignment. However, OP does not utilize any FEC. PJ2K [6] is our earlier work. In PJ2K, a single-layer JPEG2000 codestream was used, and the contents of main header (w/PPM) and the bitstreams of the codeblocks in the LL subband were protected using RS codes. The remainder of the codeblock codestreams were packetized under alignment constraints imposed to minimize network packet dependencies. The results in the table indicate that the ULP-J2K scheme provides significant advantages over existing methods, especially for high packet loss rates. Another advantage of the ULP-J2K scheme over the PJ2K scheme is that the percentage of low quality images that are generated by ULP-J2K is significantly lower than that of PJ2K. This is illustrated in Figure 5. In the figure, the x-axis denotes PSNR, and the y-axis denoted the cumulative percentage of simulations that resulted in PSNR values lower than the corresponding x value. A mean packet loss rate of 10% was used to generate this figure. It can be seen that while PJ2K can sometimes generate images that are low quality, ULP-J2K almost always produces high quality images. It is important to note that 16 × 16 codeblocks were used to obtain the results for the PJ2K algorithm presented in this figure. In contrast, the ULP-J2K scheme used 64 × 64 codeblocks. It was discovered in [6] that the PJ2K method performed better with this codeblock size, since errors occuring in small codeblocks could be better isolated. However, this problem does not occur for the ULP-J2K scheme.

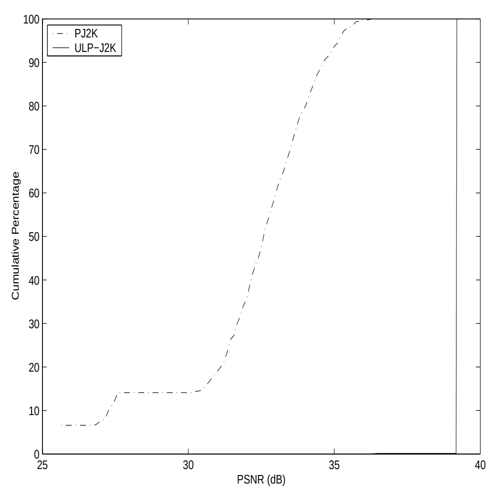

Figure 5. Comparison of PJ2K and ULP-J2K Methods for a Mean Packet Loss Rate of 10%.

It should be noted that although the ULP-J2K method is designed for a particular network packet loss rate, its operation under lower packet loss rates is possible. The performance of the ULP-J2K for this scenario would be very close to its performance under the design packet loss rate.

	Packet Loss Rate					
Packetization Method	1%	2%	5%	10%	20%	30%
ULP-J2K	39.84	39.76	39.49	39.25	38.52	37.72
PJ2K [6]	38.12	37.20	35.24	32.10	29.99	28.70
OP [3]	36.28	34.13	30.62	27.64	24.44	-

Table 1. Average PSNR results for different packetization strategies (683 network packets, 1.0005 bits/pixel).

Figure 6 presents four representative images generated using the presented method for network packet loss rates of 2%, 10%, 20%, and 30%. Notice that the image quality degrades gracefully and is acceptable even at the high loss rate of 30%.

5. SUMMARY

In this paper, a ULP framework for transmission of JPEG2000 codestreams over packet erasure channels is presented. The flexibility allowed by the JPEG2000 standard is utilized to generate a codestream that fits well into the ULP framework. The formation of the quality layers in JPEG2000 and the selection of the appropriate protection levels for each layer are performed using a JSCC strategy. Comparisons with other methods in the literature reveal that the proposed technique provides significant advantages across a wide range of packet loss rates.

REFERENCES

1. C. D. Creusere, "A new method for robust image compression based on embedded zerotree wavelet algorithm," *IEEE Transactions on Image Processing* **6**, pp. 1436–1442, Oct. 1997.
2. J. K. Rogers and P. C. Cosman, "Wavelet zerotree image compression with packetization," *IEEE Signal Processing Letters* **5**, pp. 105–107, May 1998.
3. X. Wu, S. Cheng, and Z. Xiong, "On packetization of embedded multimedia bitstreams," *IEEE Transactions on Multimedia* **3**, pp. 132–140, Mar. 2001.
4. A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory* **42**, pp. 1737–1744, Nov. 1996.
5. A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Communications* **18**, pp. 819–828, June 2000.
6. A. Bilgin, Z. Wu, and M. Marcellin, "On packetization of JPEG2000 codestreams," in *Visual Communications and Image Processing*, **4671-2**, pp. 614–625, Jan. 2002.
7. "JPEG 2000 Part I Final Draft International Standard," *ISO/IEC JTC 1/SC 29/ WG1, Doc. No. N1855* , Aug. 2000.
8. D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*, Kluwer Academic Publishers, Massachusetts, 2001.
9. A. Bilgin, Z. Wu, and M. Marcellin, "Decompression of corrupt JPEG2000 codestreams," in *Data Compression Conference*, Mar. 2003.
10. M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, "An overview of JPEG-2000," in *Data Compression Conference*, pp. 523–541, Mar. 2000.
11. Z. Wu, A. Bilgin, and M. Marcellin, "Unequal error protection for transmission of JPEG2000 codestreams over noisy channels," in *International Conference on Image Processing (ICIP2002)*, **1**, pp. 213–216, Sept. 2002.
12. Z. Wu, A. Bilgin, and M. Marcellin, "An efficient joint source-channel rate allocation scheme for JPEG2000 codestreams," in *Data Compression Conference*, Mar. 2003.

13. S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, New Jersey, 1983.
14. D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Proc.* **9**, pp. 1158–1170, July 2000.
15. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acous. and Sig. Proc.* **36**, pp. 1445–1453, Sept. 1988.
16. *http://www.kakadusoftware.com.*

(a) 2% Packet Loss (PSNR = 39.76 dB).

(b) 10% Packet Loss (PSNR = 39.22 dB).

(c) 20% Packet Loss (PSNR = 38.56 dB).

(d) 30% Packet Loss (PSNR = 37.82 dB).

Figure 6. Image quality at 1.0005 bits/pixel for different network packet loss rates.