Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Signal Processing: *Image Communication* 18 (2003) 497–514

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

JPEG2000 over noisy communication channels thorough evaluation and cost analysis

G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, C. Chamzas*

Cultural and Educational Technology Institute, 58 Tsimiski Street, Xanthi 671 00, Greece

Received 20 September 2002; received in revised form 16 February 2003; accepted 26 February 2003

Abstract

In this paper, we examine the behavior of the JPEG2000 coding scheme over noisy or congested communication channels and highlight a cost policy aspect. Two error schemes are considered, involving bit errors (noisy channel) and packet-dropping (congested channel) effects. Two bit error methods are used, consisting of flipping or eliminating the bits, and various packet sizes are put to the test of packet dropping. Extensive performance results are presented accompanied by an overall cost analysis.

Keywords: JPEG2000; Error resilience; Noisy and congested communication networks; Cost analysis

1. Introduction

The emerging need for multimedia transmission over high-speed communication systems on which several restrictions and transmission policies apply, guide the research to new ways of data handling and algorithm applications, even before the data can be transmitted. One of the problems in engineering a packet switched network carrying both non-bursty delay-sensitive traffic (voice, video) and highly bursty delay-tolerant traffic (computer data, image data) is the congestion problem [1].

Since digital bitmap representations of images require large numbers of bits, data compression techniques are important for efficient transmission. Standard lossless compression methods, such as the lossless JPEG or the JPEG-LS and JBIG coders, provide with compression ratios of about 2:1 on the average. Unfortunately, such algorithms do not have the ability to allow packet dropping by the network. Hence, when a congested facility drops a packet containing compressed image data, the rest of the image is destroyed, unless the end-user is employing an end-to-end receive—acknowledge transmission—repeat mechanism. Such a protocol saves the transmitted information, but ultimately makes matters worse for the already congested network as it further increases traffic, not to mention the additional disadvantage of increasing transmission delay. Thus, to be effective as a congestion relieving mechanism, packet dropping must be allowed with the knowledge and blessing of the end-user. Presumably such a user would be given pricing advantages for the packets that he/she marks as droppable, since this

*Corresponding author. Tel.: +30-2541-078787x112 or 079571; fax: +30-2541-063656.

E-mail address: chamzas@ceti.gr (C. Chamzas).

0923-5965/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0923-5965(03)00038-9

information is delivered only when the network is idle. Similar problems appear when noisy communication channels carrying delay-sensitive image data change or drop bits.

JPEG2000, the new coding standard, comes to fulfill such requirements of progressive coding while providing with error control mechanisms. Several papers and publications consider the performance of this coder in noisy environments in order to compare the scheme with the existing ones [2–5]. In this work, we present the results of using the JPEG2000 coder in error resilient mode with Layer–Resolution–Component–Position (LRCP) priority, considering the overall effect of different error models. Preliminary results have been reported in [6]. The outcome of our tests is an overall communication channel cost policy analysis that can be used by providers to impose fees policies and users to evaluate provided services.

2. JPEG2000

The JPEG2000 standard provides with a set of features that are of importance to many high-end and emerging applications by taking advantage of new technologies. It addresses areas where current standards fail to produce the best quality or performance and provides capabilities to markets that currently do not use compression [2].

In Fig. 1 the general block diagram of a JPEG2000 encoder and decoder is depicted. The encoding scheme comprises of a forward wavelet transform, followed by quantization and bitplane entropy coding with rate allocation capabilities. The encoder is able to produce a fully embedded codestream, which is optimal in a rate-distortion aspect. The decoder reconstructs the image by inverting the steps of the encoder. Fig. 2 shows the basic JPEG2000 encoder blocks graphically.

Some of the most important features, relevant to this work, that this standard possesses are the following [7,8]:

- superior low bit-rate performance,
- lossless and lossy compression,
- progressive transmission by pixel accuracy and resolution,
- region-of-interest coding (ROI),
- random codestream access and processing,
- robustness to bit errors,
- open architecture,
- side channel spatial information (transparency),
- protective image security,
- continuous-tone and bi-level compression.

For an in-depth analysis of the JPEG200 coding scheme the reader could refer to an excellent textbook by Taubman and Marcellin [9] and the JPEG2000 Special Issue on Image Communication Journal published by Elsevier [10].

In this work, we consider the behavior of JPEG2000 over noisy or congested communication channels and measure the error resilient coding efficiency and performance. We like to make clear that we address here the general case of any theoretic communication network and do not restrict ourselves to real-life networks of any specific kind or implementation. As known, the most important and widespread networks are

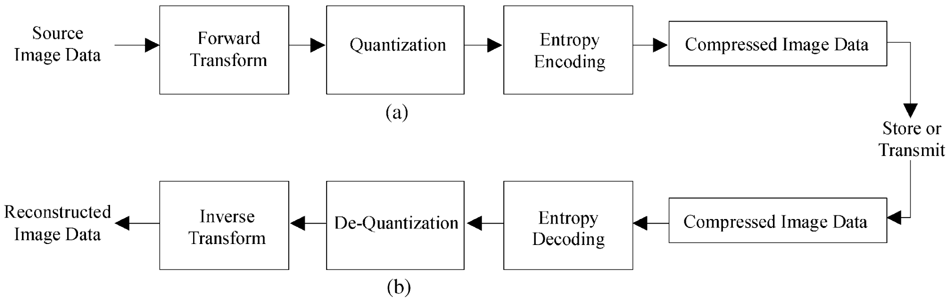

Fig. 1. JPEG2000 encoder (a) and decoder (b).

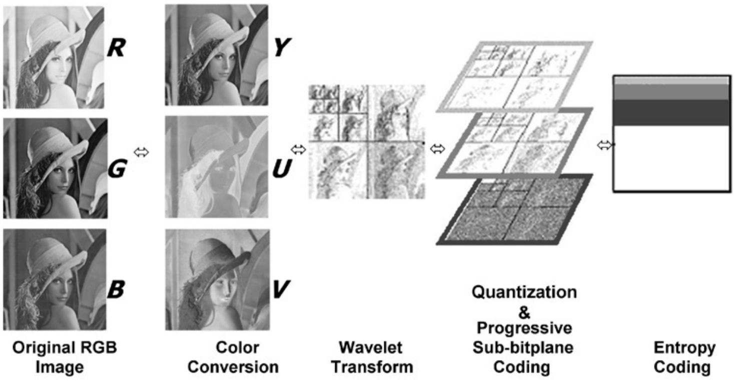

Fig. 2. Graphical representation of encoding/decoding in JPEG2000.

nowadays the packet-based wireless link layer implementations. Although we simulate conditions similar to those occurring in such networks, where every kind of information loss is actually transformed to packet loss, we also consider the case "dummy" networks where occurrence of bit errors actually takes place. The simulation results are fed to a cost analysis system that we define in this work and we, finally, produce "cost-quality" performance information, important to a communication service provider for the design and policy of a network.

3. JPEG2000 and error resilience

JPEG2000 uses a variable length coder (the MQ arithmetic coder) to compress the quantized wavelet coefficients. Variable length coding is known to be prone to channel or transmission errors. A bit error results in loss of synchronization at the entropy decoder and the reconstructed image can be severely damaged. To improve the performance of transmitting compressed images over error prone channels, error resilient bit stream syntax and tools are included in the standard.

The error resilience tools deal with channel errors using the following approaches:

- data partitioning and resynchronization,
- error detection and concealment, and
- Quality of Service (QoS) transmission based on priority [2,11,12].

Table 1
Tools for error resilience

Type of tool	Name
Entropy coding level	Code-blocks Termination of the arithmetic coder for each pass Reset of contexts after each coding pass Selective arithmetic coding Segmentation symbols
Packet level	Short packet format Packet with resynchronization marker

Error resilience is achieved at the entropy coding level and at the packet level. Table 1 summarizes the various ways this is achieved [2].

Entropy coding of the quantized coefficients is performed within code-blocks. Since encoding and decoding of the code-blocks are independent processes, bit errors in the bit stream of a code-block will be restricted within that code-block. To increase error resilience, termination of the arithmetic coder is allowed after every coding pass and the contexts may be reset after each coding pass. This allows the arithmetic decoder to continue the decoding process even if an error has occurred. The "lazy coding" mode is also useful for error resilience. This relates to the optional arithmetic coding bypass, in which bits are used as raw bits into the bit stream without arithmetic coding. This prevents the error propagation types to which variable length coding is susceptible.

Table 2
PSNR (dB) of the image ‘café’ transmitted over a noisy channel for various error rates

bpp	0	10^{-6}	10^{-5}	10^{-4}
0.25	22.64	22.45	20.37	16.02
0.5	26.21	26.01	23.35	16.20
1.0	31.39	30.25	25.91	16.52
2.0	38.27	36.06	25.80	17.16

At the packet level, a packet with a resynchronization marker allows spatial partitioning and resynchronization. This is placed in front of every packet in a tile with a sequence number starting at zero and incremented with each packet.

Table 2 summarizes the average of 200 results of decoding image ‘café’, encoded with reversible filters and transmitted over a noisy channel with various error rates (10^{-6}, 10^{-5}, 10^{-4}) as described in [5] and verified during our experimentations.

For the purposes of our work, we have encoded the three standard test images ‘woman’, ‘café’ and ‘bike’, shown in Fig. 3, using the publicly available kakadu system version 3.0 [13].

For encoding we use the kakadu command:

```
kdu_compress  -i $1.pgm -o $1.jpc -rate -,.1,.25,.5,.75,1,1.5,2 -full
              Cuse_sop=yes Cuse_eph=yes Creversible=yes
              Cmodes="RESET|RESTART|ERTERM|SEGMARK"
```

where $1 represents the image filename, while error resilient decoding is guaranteed by using the command:

```
kdu_expand -i $1.jpc -o $1.pgm -resilient_sop
```

The produced encoded images are stored in codestreams with progressive-by-layer reconstruction up to lossless (using intermediate rates of 0.1, 0.25, 0.5, 0.75, 1, 1.5, 2 bpp), including error resilience markers. The extra markers for the error resilience impose an overhead of about 1%, as expected and referenced by many other works on this subject [3–5,14]. The encoded images have to pass through a virtual noisy or congested communication channel and to be decoded in order to evaluate the robustness of the encoder and to produce cost analysis data.

4. Simulation error modes and JPEG2000

The channel is simulated by a computer program, which hits the encoded codestream using two basic error schemes:

Scheme 1. *Bit errors*: the codestream is hit in bit level, using two probability parameters (noisy channel):

(i) an error probability p_e
(ii) a burst error probability p_b

The error probability p_e is the well-known error rate with typical values of 10^{-6}, 10^{-5}, 10^{-4}, while the burst probability imposes an additional error probability simulating bursty noise or outage periods, common to real-life telecommunication systems. Burst errors applied are of rates $p_b = 1 - q$ where q is the burst factor with values 1, 0.2, 0.1, 0.02, 0.01, 0.002, 0.001, 0.0002, 0.0001, 0.00001, 0.000001, 0.0000001, with the minimum factor corresponding to higher probability of occurrence of burst errors. Thus, $p_e = 10^{-6}$ and $q = 10^{-2}$ means that with probability $p_e = 10^{-6}$ we flip or drop a continuous sequence of bits of average length $(1-q)/q^2 = 9900$ bits. Typically, error probability introduces random errors, while burst probability imposes the appearance of continuous errors just after a random error occurred.

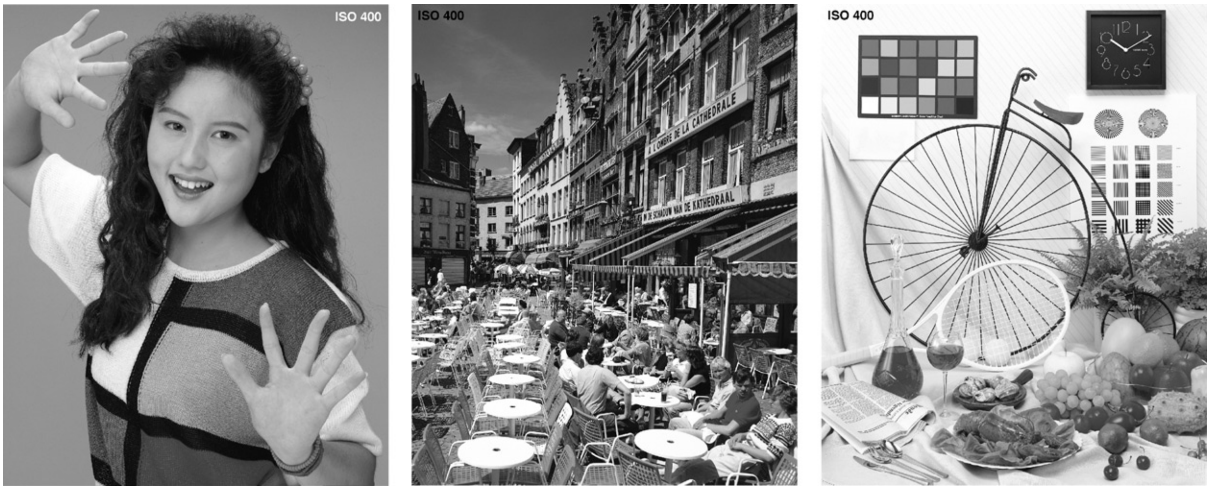

Fig. 3. Test images (from left to right: 'woman', 'café', 'bike').

In Scheme 1 we employ two modes with the following impact on a codestream, respectively:

(0) bit values are flipped (reversed);
(1) bits are dropped (do not reach the decoder—outage periods)

Scheme 2. *Packet dropping*: the codestream is hit on packet basis (congested network). Errors introduced in this scheme result in packet dropping. Typical packet sizes used in our experimentation are inside the range of 100–2000 bytes, while packet-dropping probability p_d, corresponds to typical values of 10^{-3}, 10^{-2}, 10^{-1}, 2×10^{-1}, 4×10^{-1}, 6×10^{-1}, 8×10^{-1}. At a first stage of our simulations, there is no burst probability ($q=1$). Thus, $p_\mathrm{d}=10^{-3}$ means that with a probability 10^{-3} a packet is dropped. Additionally, as the original images were encoded with eight (8) quality layers, this scheme takes into account that the droppable part of the codestream varies from 1 to 7 higher quality layers. The simulation, in this case, takes into account that the first packets of the JPEG2000 codestream always reach the decoder side error-free, just to make sure that, at least, global header information can always reach the decoder.

In order to simulate a more realistic communication network, similar to the wireless packet-based networks, we employed a second packet-dropping scenario, were the parameter of burstiness is present in the system: a burst error probability $p_\mathrm{b}=1-q$ is induced with q being, as in scheme 1, the burstiness factor assuming the values of 0.2, 0.1, 0.02, 0.01, 0.002, 0.001, 0.0002, 0.0001, 0.00001, 0.000001. As in the first scheme, the final packet-dropping rate is significantly altered by the introduction of the burst factor.

The overall simulation is as follows:

- The kakadu JPEG2000 encoder encodes an image with eight (8) quality layers with the higher layer representing lossless reconstruction, using reversible filters and error resilience capabilities (extra markers in the codestream).
- The produced codestream, in Scheme 1, is corrupted by a noisy channel simulation software with two error modes and with various random error and burst error rates. The hit symbols are flipped or dropped according to the imposed error mode (discussed in Section 4.1). In Scheme 2, the produced codestream is corrupted by a congested channel simulation program, where the hit packets are dropped using a packet dropping and a burstiness rate.
- The kakadu JPEG2000 decoder decodes the corrupted codestream taking into account that error resilience coding was applied, and reconstructs the image.

- An evaluator (program imgcmp, which is part of the public available “jasper” system [15]) is used to estimate the resulting image quality.

Results are shown and interpreted in the following sections. Image quality is represented by—the commonly used metric—peak signal-to-noise ratio (PSNR), which is calculated from the root mean squared error (RMSE) as follows:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{2^b - 1}{\mathrm{RMSE}}\right)^2 = 20\log_{10}\left(\frac{2^b - 1}{\mathrm{RMSE}}\right), \tag{1}$$

where b is the bit-depth (bpp) of the original image and RMSE is the root mean squared error of the reconstructed compared to the original image.

PSNR reduces to the calculation

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{255}{\mathrm{RMSE}}\right) \tag{2}$$

in the case of 8-bit gray-level images.

For the first error scheme, we run 100 tests on each test image, for the two error modes, the three error rates and the 12 burst probability factors. Additionally, we kept one to seven quality layers (of the eight) lossless. The simulation turned out to produce 151,200 results.

For the second error scheme, we run 100 tests on each test image and hit the stream by dropping packets of 20 different sizes from 100 to 2000 bytes (in 100 byte increments—20 sizes) with seven packet-dropping probabilities. Additionally, we kept from none to seven quality layers intact, while corrupting the remaining data. We got 336,000 results. In all cases the first packets were losslessly transmitted just to ensure uncorrupted global header information.

4.1. Results for error Scheme 1: occurrence of bit errors

We start the presentation of results from error Scheme 1, which represents the scenario of hitting the codestream with bit errors (noisy channel, no retransmission).

Overall average results showed that mode 1 (bit dropping) has a greater impact on the decoded codestream, considering the ‘distortion-error probability’ performance regardless of the error probability. An explanation to this is that in a bit dropping environment, additionally to loosing plain data bits (with the effect of loosing synchronization until a following header or marker is encountered), we may loose bits corresponding to packet or tile-part headers, thus destroying, probably, large parts of the codestream, as the following data are discarded as meaningless until a marker that is legal and meaningful in the process is encountered.

Fig. 4 shows the average data corruption percentage for the three error probabilities and for the two error modes. Values are averaged over all burst probabilities. The effect of burstiness is evident in these measurements, since, for example when error rate is 10^{-6} the average corruption becomes about 10%, which corresponds to an effective error rate of 10^{-1}, much higher than the original 10^{-6}. As expected, data corruption is independent of the error mode used. Hereafter, all references to error rates include an averaging over all burst error rates, thus by error probability we actually mean the probability of starting a sequence (block) of errors.

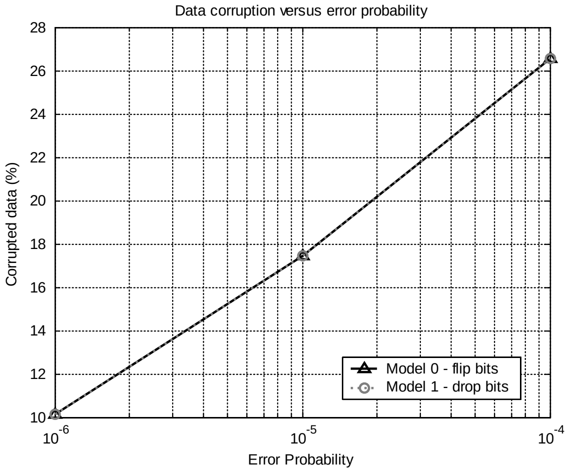

Fig. 4. Average data corruption (%) versus error probability for the three error modes.

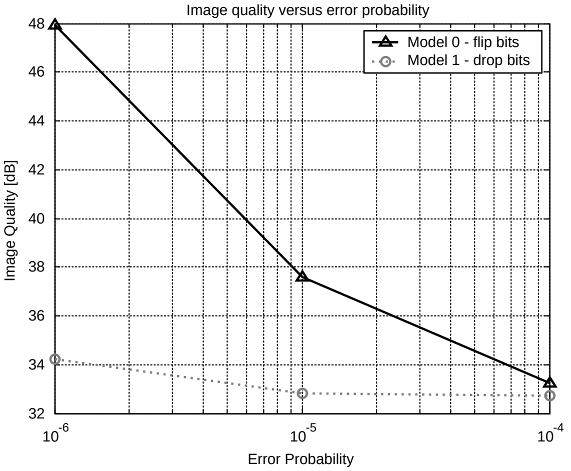

Fig. 5. Average quality versus error probability for the two error modes.

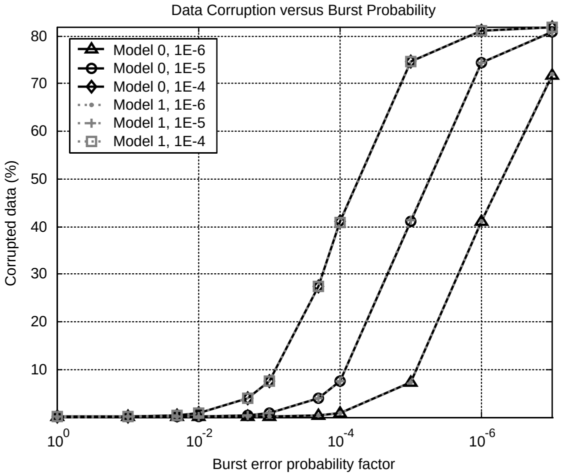

Fig. 7. Data corruption (%) versus burst error factor.

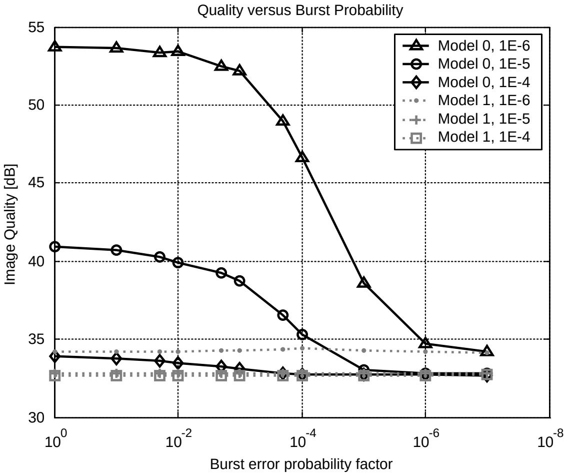

Fig. 6. Quality versus burst error factor.

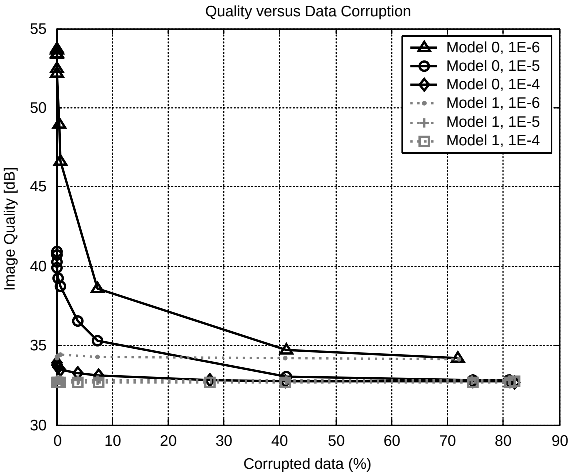

Fig. 8. Quality versus data corruption (%).

Fig. 5 shows the average image quality (PSNR in dB) versus error probability for the error modes described above, and, again, averaged over all burst error rates.

One can observe that with error mode 1 (bit dropping) quality is lower than with error mode 0 (bit flipping), but the difference decreases as error rate increases. In Figs. 6–8 the impact of burst errors in image quality and data corruption is shown. Fig. 6 depicts the image quality versus the burst factor *q*. It is noticeable that error mode 1 results in significantly lower quality measurements, especially in the case of error rate 10^{-6} where the difference to the error mode 0 measured quality takes the value of about 20 dB. In higher error rates these differences tend to take lower values and become almost insignificant in the case of error rate 10^{-4}.

Fig. 7 depicts the percentage of corrupted data versus the burst error factor *q*. It is clear that the choice of the error mode does not affect the results. Both modes give exactly the same measurements from the data corruption point of view. Data corruption is measured by means of the percentage of symbols actually hit by the error and burst error generator.

Fig. 8 shows the decoded image quality versus the percentage of data corruption, and it is, actually, a combination of the graphs in Figs. 6 and 7. One can observe that quality drops rapidly; for example, on the curve corresponding to error rate 10^{-6} quality drops to about 38 dB for data corruption of up to about 10% and remains almost constant (about 3 dB or less total decrease) for data corruption up to almost 75%. As expected from the previous results, the worst case is when working with mode 1, where curves reveal an almost flat tendency representing the fact that even with low corruption the quality is expected to be poor. On the other hand, when applying error mode 0, the outcome is proportional to the error and burst error rate present in the network.

Another important aspect of our experimentations is illustrated in the following figures. In Figs. 9 and 10 we show the average data corruption and quality per number of lossless quality layers. If the system can guarantee the lossless transmission of the first 1–7 layers, then the effect of the application of noise in the remaining layers is displayed in terms of percentage of data corruption in Fig. 9, and average PSNR in Fig. 10. It is evident that curves in Fig. 9 can be approximated by straight lines of different slopes for every error rate. For example, for the case of error rate 10^{-5} (vertically middle curve), the approximating line can be expressed as

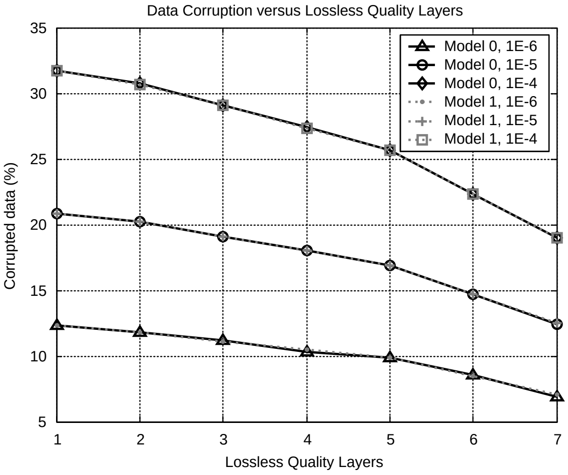

Fig. 9. Data corruption versus number of lossless quality layers.

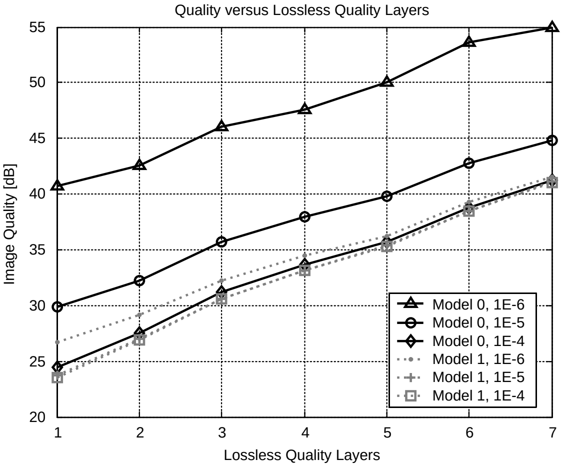

Fig. 10. Average quality versus number of lossless quality layers.

$$y = -1.42x + 22.86 \tag{3}$$

corresponding to a slope of about -1.42 [% corruption/quality layer].

Furthermore, Fig. 10 illustrates the fact that quality is directly associated both with error rates and error modes. In fact, all curves in this figure can be approximated by straight lines of about 2.5 [dB/quality layer] slope and expressed by

$$z = 2.5x + b, \tag{4}$$

where the parameter b can take a value depending on the curve (for example, for the 10^{-5} mode 0 curve, $b = 27.34$).

What we have in this case is a linear response for the quality when the controlling variable is the number of lossless quality layers reaching the decoder. By knowing the decoding quality for the lower quality layer for a given error mode and error rate, we are able to predict the quality of the decoded image whenever additional quality layers reach losslessly the decoder.

4.2. Results for error Scheme 2: congested network with packet dropping

In this section, we present the results from simulations using error Scheme 2, which represents the scenario of dropping packets in a congested network. Thus, what we call now error probability is the probability with which the network will drop packets. Fig. 11 shows the average data corruption percentage for the seven error probabilities (averaged over all packet sizes in the experiment). The actual corrupted data percentage is slightly smaller than the one indicated by the error probability, since the global information packets in the beginning of the codestream are not allowed to be corrupted, and because we average over situations where different number of quality layers are allowed to be considered droppable.

Following the rapid increase in data corruption, average quality decreases equally rapid with increasing error rate, as shown in Fig. 12, where average quality is depicted versus error probability. Again the situation is as expected, since the packet dropping with increasing powers of 10, truncates more and more of the original codestream and the decoder rapidly becomes unable to reconstruct the received data (even when full error resilience coding is forced during the encoding).

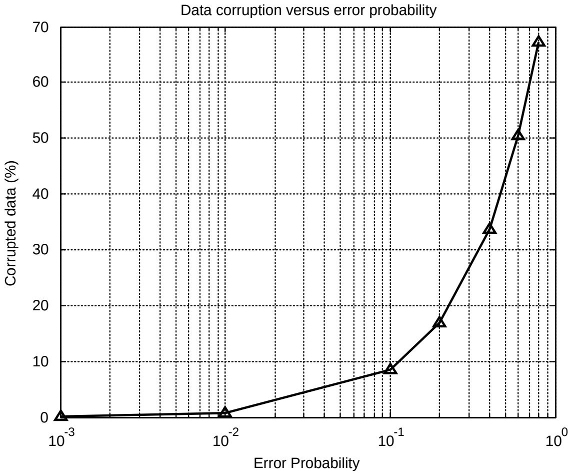

Fig. 11. Average data corruption (%) versus error probability.

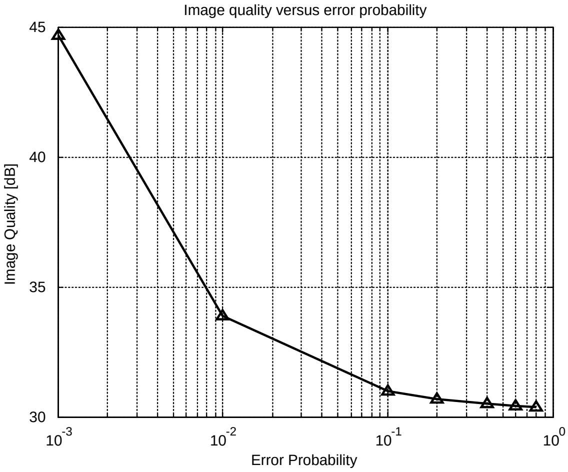

Fig. 12. Average quality versus error rate.

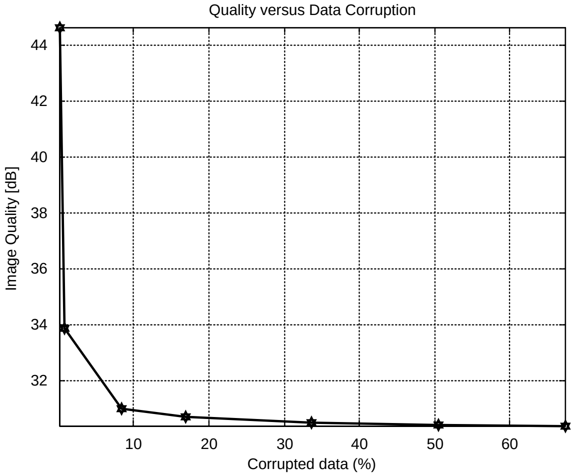

Fig. 13. Quality versus data corruption (%).

Following the paradigm of error Scheme 1, we have to combine the graphs of Figs. 11 and 12 in order to produce a Quality–Corruption plot. This is shown in Fig. 13, where average image quality is shown versus the percentage of data corruption for the seven error rates. It is quite clear that the curve shows an exponentially decreasing tendency, a behavior similar to that of error Scheme 1 (Fig. 8).

Another important aspect is illustrated in the following figures. In Figs. 14 and 15 we show the average data corruption and quality versus the

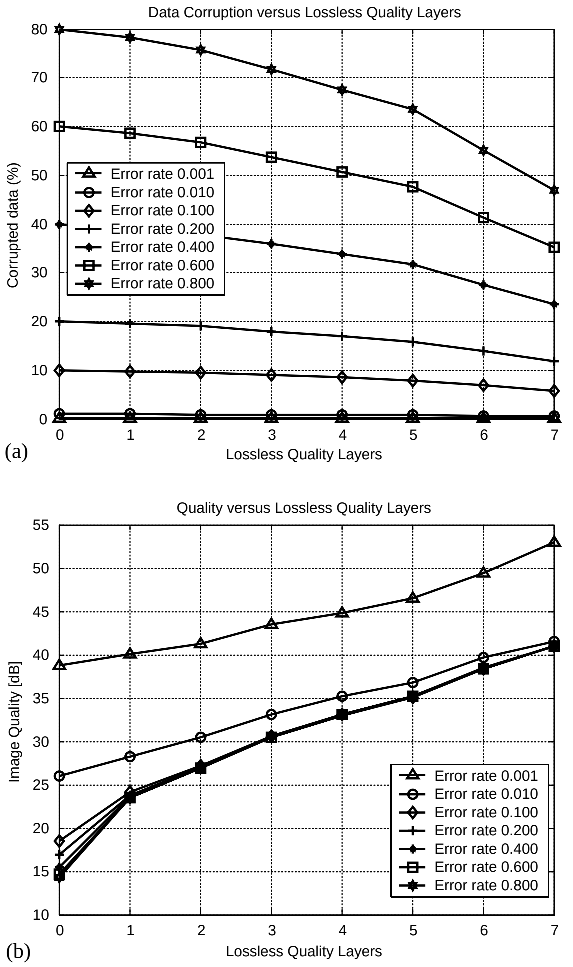

Fig. 14. Data corruption versus size of packet and number of lossless quality layers.

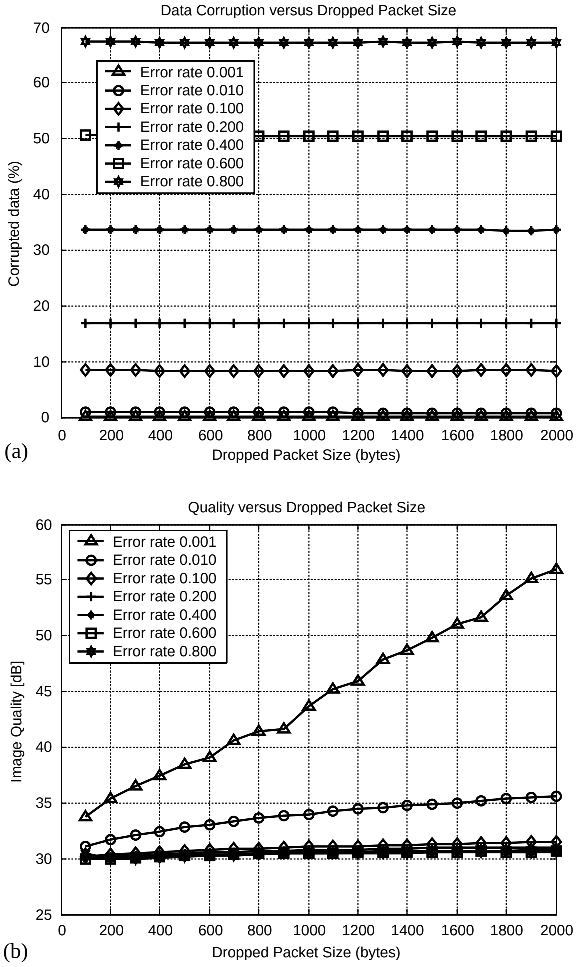

Fig. 15. Quality versus size of packet and number of lossless quality layers.

packet size used and lossless quality layers selected for all error rates. Fig. 14 shows the percentage of data corruption and average decoded image quality versus the number of losslessly transmitted quality layers, while Fig. 15 shows the data corruption and average decoded image quality versus the size of the packets.

An important issue raised in Fig. 15b is the fact that the final decoded image quality is proportional to the dropped packet size, with increasing packet size resulting in increasing image quality and, additionally in a linear manner. Another result is that the slope of these curves is increasing with decreasing packet-dropping rate. This increasing in slope with decreasing error rate is illustrated in Fig. 16.

Introduction of burstiness into the packet-dropping scenario alters significantly the overall error rate as in the case of bit errors. The overall packet-dropping evaluation turns out to be similar to the case when burstiness is not present but error rate is highly increased. Note here that the scheme we have tested assumes a burstiness probability which controls the number of continues packets that are dropped by the network. This can be considered to be the worst case scenario were packet dropping during a congestion period affects multiple users simultaneously.

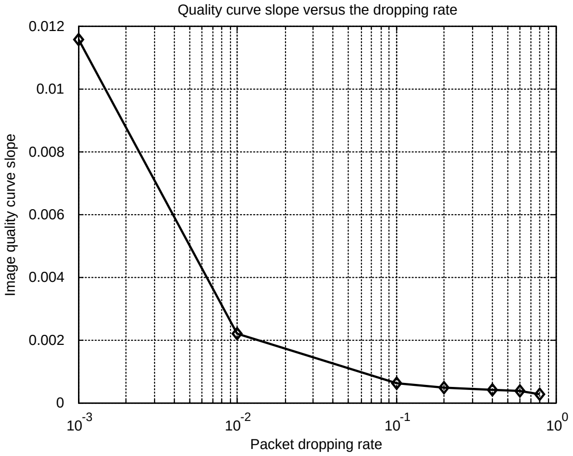

Fig. 16. Quality curves' slopes versus packet-dropping rate.

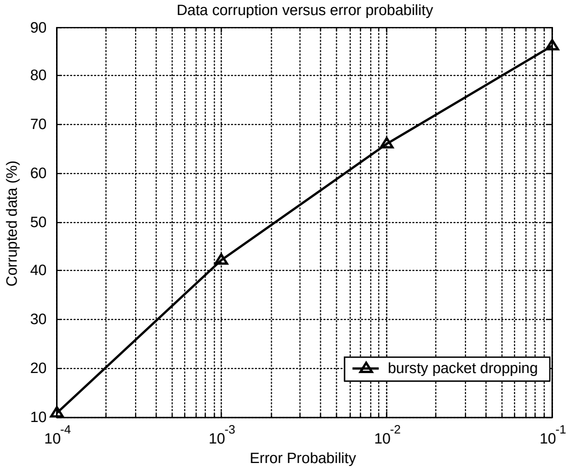

Fig. 17. Average data corruption (%) versus error probability with burstiness present.

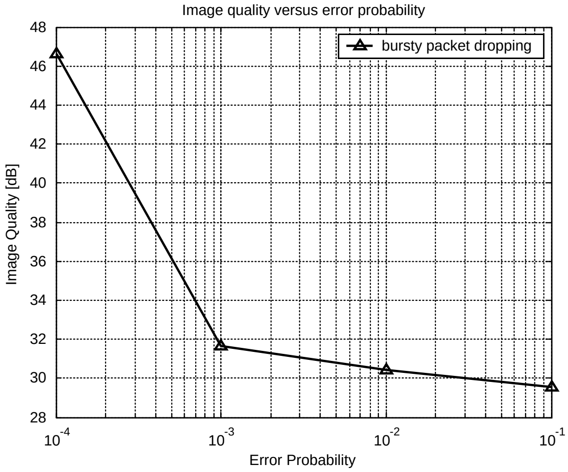

Fig. 18. Average quality versus error rate with burstiness present.

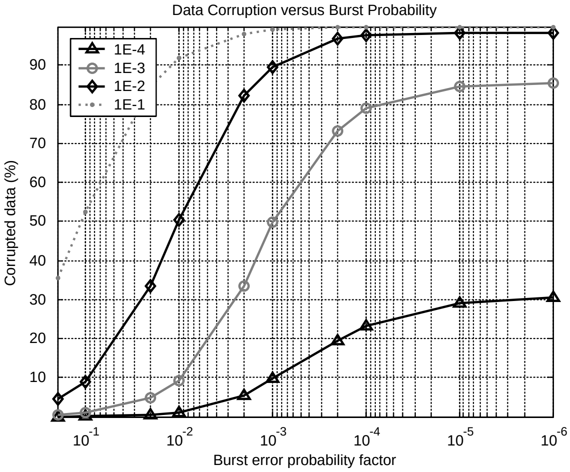

Fig. 19. Average data corruption (%) versus burst error probability.

Fig. 17 shows the average data corruption percentage for the four error probabilities (averaged over all packet sizes, error-free selected quality layers and burstiness factors in the experiment). The actual corrupted data percentage is larger than the one indicated by the error probability due to burstiness.

Following the rapid increase in data corruption, average quality decreases equally rapid with increasing error rate, as shown in Fig. 18, where average quality is depicted versus error probability (averaged over all the other parameters). Again the situation is as expected, since packet dropping with increasing powers of 10 and the introduction of burstiness in its worst case, truncates more and more of the original codestream and the decoder becomes unable to reconstruct the received data (even when full error resilience coding is forced during the encoding).

The effect of burstiness is shown in the following figure (Fig. 19), where the average data corruption percentage and the average quality are depicted as functions of the burst probability factor.

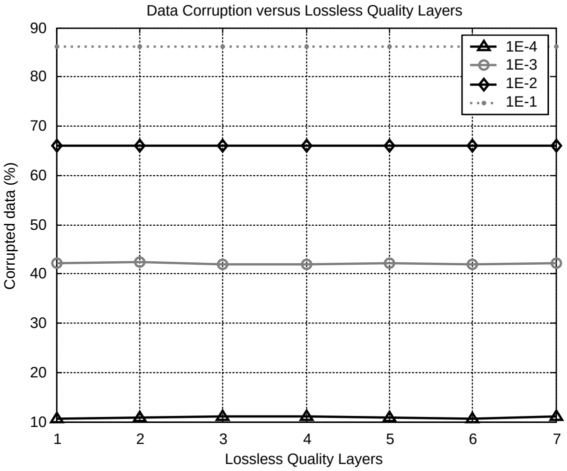

Fig. 20. Data corruption versus error-free quality layers.

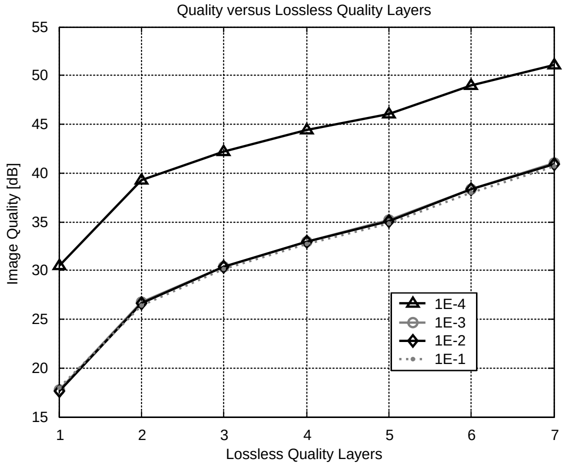

Fig. 21. Quality versus error-free quality layers.

The next two figures highlight the effect of the variable error-free quality layers. In Fig. 20 we show the average data corruption and in Fig. 21 the average quality versus the error-free transmitted quality layers.

5. Cost analysis

In Section 3, we defined the simulation process. In Section 4, we presented the impact of the error schemes and modes. In this section, we will present the cost analysis aspect of the system, for the same set of resulting data.

For the cost analysis of such a system, one has to device some sort of policy to evaluate the various parts of a transmitted codestream. For the purposes of our work, we have adopted the following network policy:

- The codestream is now divided into droppable and non-droppable parts, in the sense that the network is able to guarantee lossless transmission of the lower quality layers 1–7 (which are considered to be non-droppable or essential). The remaining quality layers (7–1 accordingly) are subject to channel errors and are considered to be droppable and thus, they are not guaranteed to reach their target losslessly.
- The non-droppable and droppable bytes are assigned a cost ratio. Cost ratios tested are from 1:1 to 10:1 in 1:10 increments (90 ratios). Non-droppable data bytes are assigned a cost of 1 cost unit per byte and thus, cost ratios are expressed in terms of non-droppable:droppable cost units per byte.

According to these assumptions cost can be calculated as

$$C = N_{\mathrm{nd}} + c_{\mathrm{r}} N_{\mathrm{d}}, \tag{5}$$

where C is the total cost, N_{nd} are the non-droppable data bytes, c_{r} is the cost ratio and N_{d} are the droppable data bytes.

5.1. Cost analysis when flipping or dropping bits

In our experimentations, due to the sizes of the test images, cost calculations resulted in a cost range of $[0.5 \times 10^7, 10^7]$ cost units, so costs had to be normalized by dividing with the maximum value to the range $(0, 1]$. In most cases the presented cost is expressed in terms of other quantities such as quality or lossless data, and, thus, is a relative quantity.

Fig. 22 depicts the cost per "healthy" data versus the burst probability factor for all error rates and for cost ratio of non-droppable:droppable data set to 3:1. By cost per "healthy" data we mean the cost relative to data not corrupted by

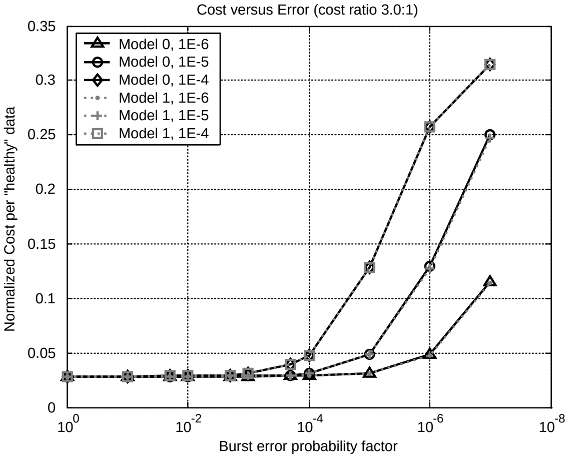

Fig. 22. Cost per "healthy" data versus burst error factor.

channel errors, i.e. how much do non-corrupted data received by the decoder cost.

Fig. 23, which we present as the overall evaluation diagram of the channel, depicts the average cost per dB of finally decoded quality versus the decoded image quality (PSNR in dB) for the seven quality layers, and for cost ratio set to (a) 2:1, (b) 3:1 and (c) 4:1.

The data points are obtained by increasing the number of non-droppable quality layers from 1 to 7 (0.1–2 bpp). From this diagram, one is able to evaluate when the cost of every dB of quality is worth the achieved quality and by how many lossless quality layers can this be accomplished. For example, in the case of cost ratio 3:1, mode 0 and error rate 10^{-5}, the cheapest dB of quality is at about 36 dB when using three lossless quality layers, the corresponding cost per dB is about 0.08. From another point of view, we must transmit the image in an error-free mode up to 0.5 bpp and allow errors to occur to the rest.

System designers could plot diagrams similar to the ones shown in Fig. 23 for their cases of cost ratio system policy. In our experimentations, only cost ratios ranging from 2:1 to 4:1 has shown to give interesting results concerning policy choices: it is the only situation where the cheapest quality dBs are not located at the edges of the curves. When using cost ratios of less than 2:1, the cheapest dBs are achieved when all but one (7) quality layers are guaranteed to be losslessly transmitted. When using cost ratios of more than 4:1, the cheapest dBs are achieved when only one (1)—the lower—quality layer is guaranteed to be losslessly transmitted.

In addition to Fig. 23, Fig. 24 represents a plot of the number of lossless quality layers adequate to achieve minimum cost versus the cost ratio. In this figure, it is made clear how many quality layers should be losslessly transmitted in order to achieve the cheapest quality dBs for every cost ratio in the range of 1:1–10:1. Continuing the example given for Fig. 23, we see that for cost ratio 3:1 (value 3 in x-axis) and for mode 0 and error rate 10^{-5}, the number of quality layers is 3—which actually can be deduced from Fig. 23, where the third marker on the corresponding curve represents the three quality layers transmitted error-free. Figs. 23 and 24 can provide with the best choice in fees policy (cost ratio) for a given situation of error mode, error rate and quality level.

The conclusions that can be extracted from the results and the cost analysis diagrams are:

- When a noisy channel experiences high error rates (such as 10^{-4}) then image quality is expected to be low on the average (33 dB, in the range of 24–41 dB). In the case of error rate 10^{-5} where quality ranges from 28 to 44 dB (36 dB on the average). When error rate is 10^{-6} the quality ranges between 40 and 55 dB (47 dB on the average).
- When mode 0 errors are present (bits are flipped) then the cost per dB of quality is lower than in error mode 1. From Fig. 24 we can deduce that one would typically need less error-free quality layers to achieve cheap dBs. It is also notable that the final decoded image quality would be much better, approaching the "visually lossless" levels. Cheap quality dBs can be achieved by using typically one error-free quality layer less than when applying mode 1 errors, which indicates that a smaller amount of data should be guaranteed to be error-free transmitted, and with high quality final outcome (about 45 dB for error rate 10^{-6}).
- When mode 1 errors are present (bits are dropped), i.e. channel with outage periods, then the cost per dB of quality is high no matter

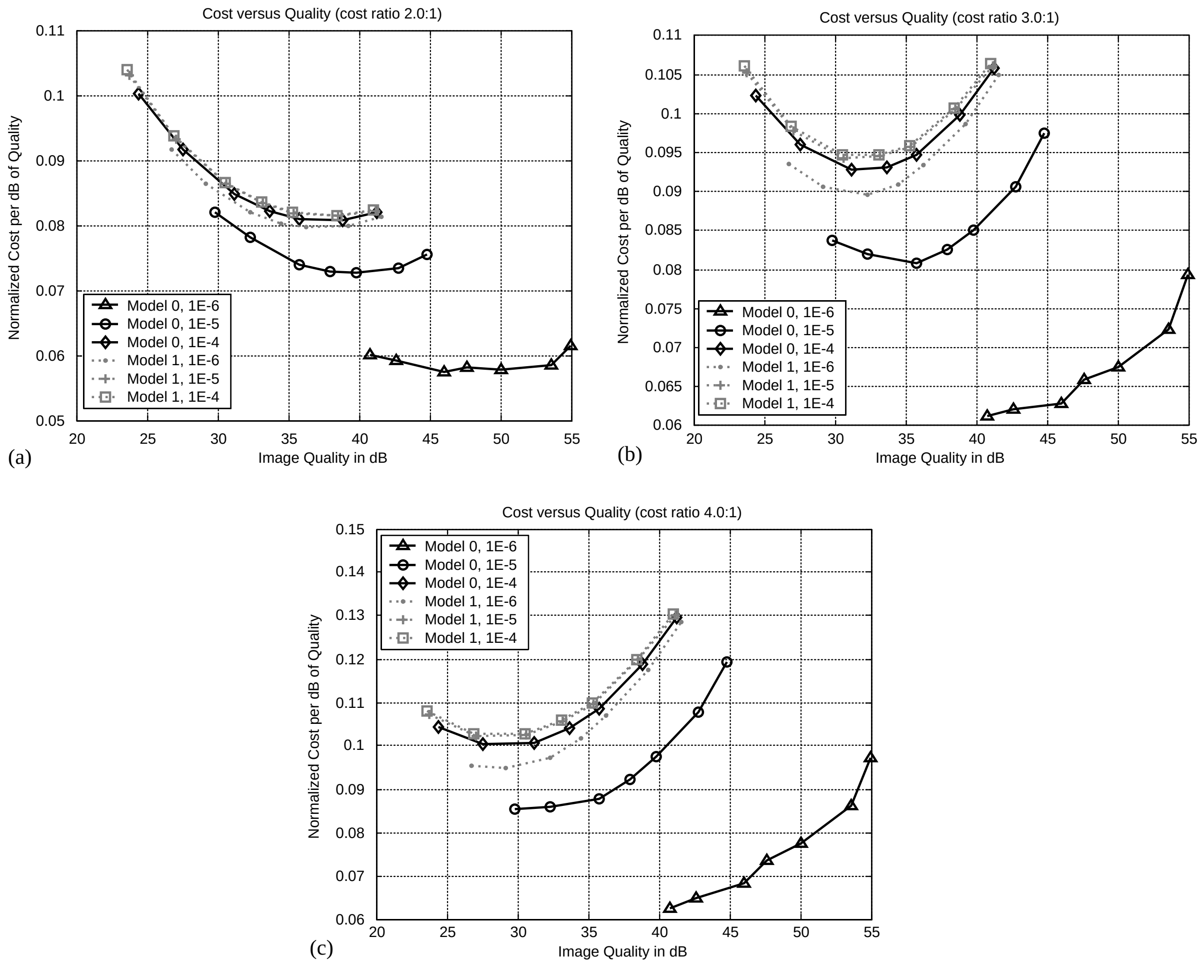

Fig. 23. Average cost per dB of image quality versus image quality (PSNR in dB) for cost ratios: (a) 2:1, (b) 3:1 and (c) 4:1.

what the error rate is, as shown in Fig. 23. It is also notable that the final decoded image quality would be low even when all but one (7) quality layers are transmitted error-free. For a cost ratio of 2:1, cheap quality dBs can be achieved by using five to seven error-free quality layers which indicates that a large amount of data should be guaranteed to be transmitted in an error correction mode, with low quality final outcome.

- Cost ratios inside 2:1–4:1 are “critical” in the sense that curves (as in Fig. 23) are concave with almost horizontal first derivative. Lower bounds of costs per quality dBs occur in typically the middle of the curves. In all other situations, the curves increase or decrease monotonically and thus, correspond to non-realistic and “unfair” costs.

5.2. Cost analysis when dropping packets

Following the steps of experimenting with JPEG2000 codestreams hit by bit errors, we now present the results of applying the second error scheme, the packet dropping. In this situation the end-user can mark some of the packets as droppable, setting a better price for them. After dropping packets of various sizes (100–2000 bytes with 100 bytes step), and varying the size of

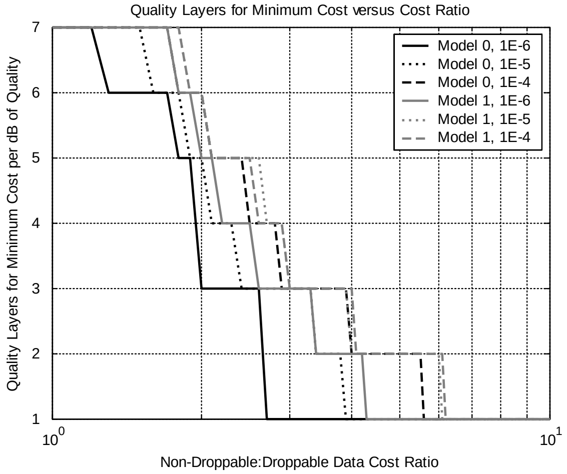

Fig. 24. Quality layers for minimum cost per dB of quality versus cost ratio.

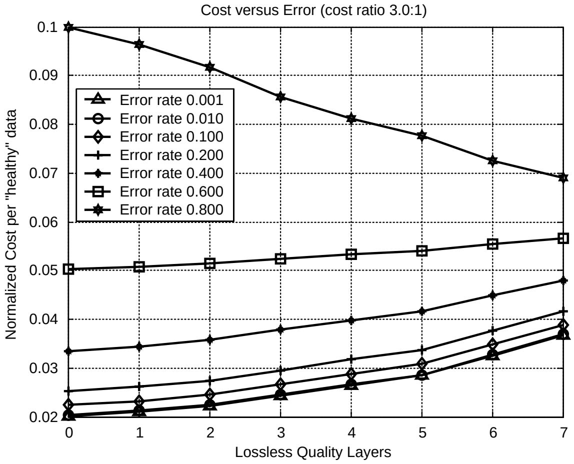

Fig. 25. Cost per “healthy” data versus number of lossless quality layers.

non-droppable data, we reached to the following results and conclusions.

In Fig. 25 we show the cost per “healthy” data, i.e. how much does a correct byte cost on the average, versus the size of non-droppable data (in terms of lossless quality layers). In this diagram the cost is expressed as the data actually reaching the decoder for the various packet sizes. The cost is increasing with the packet size for low error rates, while it decreases elsewhere.

In Fig. 26, we show the cost per quality dB versus the decoded image quality, again for the cases of 2:1, 3:1 and 4:1 cost ratios. Each curve represents a different error rate in the system.

Results are quite similar to those for error Scheme 1, with high error rates demanding more quality layers to be transmitted as non-droppable data, and resulting in typically low overall image quality. This fact is also illustrated in Fig. 27 where the number of lossless quality layers to achieve minimum cost per quality dB is drawn versus the cost ratio.

Similar results are also obtained for the case of packet dropping with burstiness.

A network administrator can take these results and thoughts into account when considering about designing a communication channel and about imposing policies and fees according to end-user requirements and network provider restrictions, costs and profit. On the other hand, the end-user is able to evaluate the provided services and choose the most suitable provider for his special requirements and applications.

If we define as fair policy the one that charges the same cost amount per dB of quality delivered, then we tend to choose cost ratio curves (in Figs. 23 and 26) which are balanced between their minimum value, or symmetric about a vertical axis passing through the minimum value. This way we can summarize the overall cost analysis for a “fair” channel in Table 3, where the preferred cost ratio is shown for every error rate used.

6. Conclusions

In this work, we simulated a communication network assigned with the task to transmit progressive-by-quality JPEG2000 codestreams through a noisy or congested channel. Extensive tests run on the standard test images using various error schemes, modes, rates, burst rates, packet sizes and a variable number of error-free transmitted quality layers gave interesting results concerning not only the error resilience capabilities and restrictions of the JPEG2000 coding scheme, but also highlighted a cost policy analysis aspect of

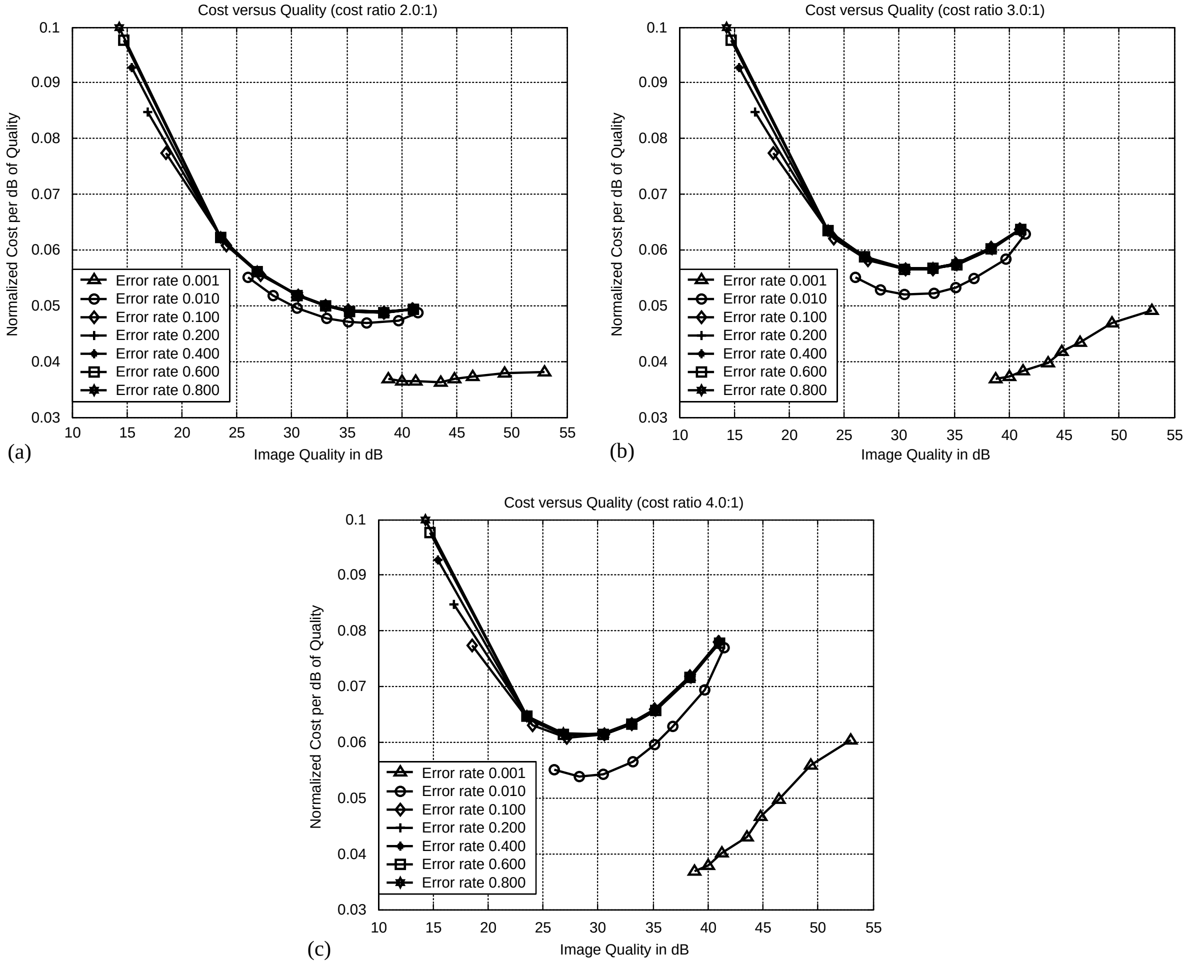

Fig. 26. Average cost per dB of image quality versus image quality (PSNR in dB) for cost ratios: (a) 2:1, (b) 3:1 and (c) 4:1.

such a communication system. After analyzing this behavior, we defined a "fair" cost network policy and provided with a table of fair cost ratios according to the error rate.

Summarizing, we are able to say that:

(1) *in a noisy channel (bit-flipping environment),*
- in our error model, corruption of data and quality of decoded image averaged over the burstiness of error occurrence are linearly dependant on the logarithm of the probability of error;
- decoded image quality versus data corruption exhibits a parabolic behavior matching a usual rate-distortion curve;
- data corruption and decoded image quality versus the number of guaranteed transmitted error-free quality layers, both exhibit straight line behavior with constant slopes (in the case of quality) or slopes dependent upon the error rate (in the case of corruption);
- cost of "healthy" data is proportional to error rate, and increases with increasing burst error rates;
- cost per quality dB exhibits upper and lower bounds in strict relation with the error rate. Proper selection of cost ratios can result in optimum or "fair" channel operation (from a cost aspect);

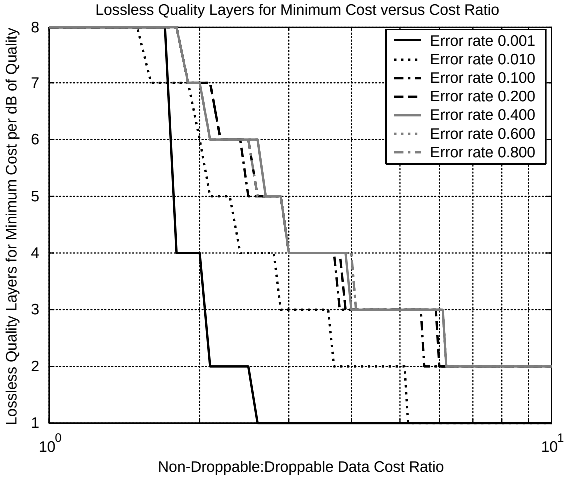

Fig. 27. Quality layers for minimum cost per dB of quality versus cost ratio.

Table 3
"Fair" cost ratios for a congested network with different error rates

Error rate	Packet dropping cost ratio
0.001	1.9:1
0.01	2.4:1
0.1	4.0:1
0.2	4.5:1
0.4	5.0:1
0.6	5.4:1
0.8	5.5:1

- error-free quality layers-cost ratio curves, when complementing cost-quality curves, can provide with additional insight aiding in the selection of the appropriate guaranteed error-free bandwidths according to network utilization needs and policies;

(2) *in a channel with outage periods (bit-dropping environment),*

- quality of decoded image is lower than the one observed in the bit-flipping mode and shows less dependence on burst error rates;
- decoded image quality versus data corruption does not exhibit a parabolic behavior but it rather expresses a linear relation, with quality being almost constant for any value of data corruption percentage;
- data corruption and decoded image quality versus the number of guaranteed transmitted lossless quality layers, both exhibit straight line behavior with constant slopes (in the case of quality) or slopes dependent upon the error rate (in the case of corruption), matching the behavior of bit-flipping error mode. Again quality levels are much lower than in bit-flipping error mode;
- cost of "healthy" data, cost per quality dB and lossless quality layers-cost ratio curves, exhibit the same characteristics as illustrated in the case of bit-flipping error modes. The differences are the lower quality with a higher cost per quality dB;

(3) *in a congested channel (packet-dropping environment),*

- decoded image quality versus data corruption exhibits the same parabolic behavior as in the case of bit-flipping error mode;
- decoded image quality versus the size of packets transmitted by the network confirms that in high congestion small packets are preferable;
- cost per quality dB versus the error-free transmitted quality layers shows similar characteristics to the ones in the bit-hitting modes. The fact is also highlighted in the error-free transmitted quality layers for minimum cost versus the cost ratio graph.

References

[1] C. Chamzas, D.L. Duttweiler, Encoding facsimile images for packet-switched networks, IEEE J. Selected Area Commun. 7 (5) (June 1989), 857–864.

[2] C. Christopoulos, A. Skodras, T. Ebrahimi, The JPEG 2000 still image coding system: an overview, IEEE Trans. Consumer Electron. 46 (4) (November 2000) 1103–1127.

[3] D. Santa-Cruz, T. Ebrahimi, A study of JPEG 2000 still image coding versus other standards, Proceeding of the X European Signal Processing Conference (EUSIPCO), Tampere, Vol. 2, September 2000, pp. 673–676.

[4] D. Santa-Cruz,T. Ebrahimi, An analytical study of JPEG 2000 functionalities, Proceedings of the IEEE ICIP 2000, Vancouver, BC, Canada, September 2000.

[5] D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Karsson, C.A. Christopoulos, JPEG 2000 still image coding versus other standards, Proceedings of the SPIE 45th Annual Meeting, Applications of Digital Image Processing XXIII, Vol. 4115, San Diego, CA, USA, Jul. 30–Aug. 4, 2000, pp. 446–454.

[6] G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, C. Chamzas, JPEG2000 over noisy communication channels—the cost analysis aspect, ICIP 2002, Rochester, NY, September 22–25, 2002, USA.

[7] ISO/IEC JTC1/SC29/WG1 N390R, New work item: JPEG 2000 image coding system, March 1997.

[8] ISO/IEC JTC1/SC29/WG1 N505, Call for contributions for JPEG 2000 (JTC 1.29.14 15444): Image Coding System, March 1997.

[9] D.S. Taubman, M.W. Marcellin, JPEG Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, Dordrecht, 2002.

[10] JPEG2000 Special Issue, Signal Processing: Image Communication, Vol. 17, No. 1, Elsevier, Amsterdam, January 2002.

[11] I. Moccagatta, S. Sudagar, J. Liang, H. Chen, Error resilient coding in JPEG2000 and MPEG-4, IEEE J. Selected Areas Commun. (JSAC) 18 (6) (June 2000) 899–914.

[12] J. Liang, R. Talluri, Tools for robust image and video coding in JPEG2000 and MPEG-4 standards, Proceedings of the SPIE Visual Communications and Image Processing Conference (VCIP99), San Jose, CA, USA, 23–29 January 1999.

[13] http://maestro.ee.unsw.edu.au/~taubman/kakadu.

[14] M. Boliek, C. Christopoulos, E. Majani (Eds.), JPEG2000 Part I Final Draft International Standard (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855, August 2000.

[15] http://www.ece.ubc.ca/~madams/jasper.