# REDUCED-DELAY SELECTIVE ARQ FOR LOW BIT-RATE IMAGE AND MULTIMEDIA DATA TRANSMISSION

*Tuyet-Trang Lam[1], Lina J. Karam[1], Rida A. Bazzi[2], Glen P. Abousleman[3]*

[1]Electrical Engineering Dept., Arizona State University, Tempe, AZ 85287-5706.
[2] Computer Science & Engineering Dept., Arizona State University, Tempe, AZ 85287-5406.
[3] Compression, Comm. & Intelligence Lab, General Dynamics C4 Systems, Scottsdale, AZ 85257.
*lamsnow@asu.edu, karam@asu.edu, bazzi@asu.edu, glen.abousleman@gdds.com*

## ABSTRACT

This paper presents a reduced-delay selective-ARQ scheme based on a similarity check function that measures the degree of corruption in transmitted multimedia data packets. Accordingly, if the packet is found to be badly corrupted based on a specified similarity criteria, it can be considered as a lost packet and retransmitted. The degree of corruption contributed by a particular packet is measured by the proposed similarity check function at the receiver without explicit knowledge of the original source data. Simulation results and comparisons with a conventional ARQ scheme are provided to illustrate the performance of the proposed method.

## 1. INTRODUCTION

In typical data transmission systems, the original source file is compressed, packetized, and transmitted through a noisy channel. Because bit corruption can cause a catastrophic loss of information, the packets are protected with channel codes, and error detection is performed on each packet at the receiver. If a feedback channel is available at the decoder, retransmission requests can be provided by automatic-repeat request (ARQ) and hybrid-ARQ protocols (H-ARQ) [1] at the cost of a reduction in throughput. However, existing error detection schemes do not provide information about the effects of a corrupted packet on the quality of the reconstructed data. Bit errors on significant bits may propagate throughout the reconstructed data, while bit errors on less significant bits may not present visible distortions.

A scheme has been proposed in [2] for selective error recovery of DCT block-based image compression. A "side match test" consisting of computing the mean squared error between pixels in adjacent blocks has been proposed to detect block errors that causes serious image degradation. This test combined with sending the block length shows an increase of channel throughput compared to a conventional ARQ scheme. However, the method of [2] is not suitable for wavelet-based and other non-block based coders.

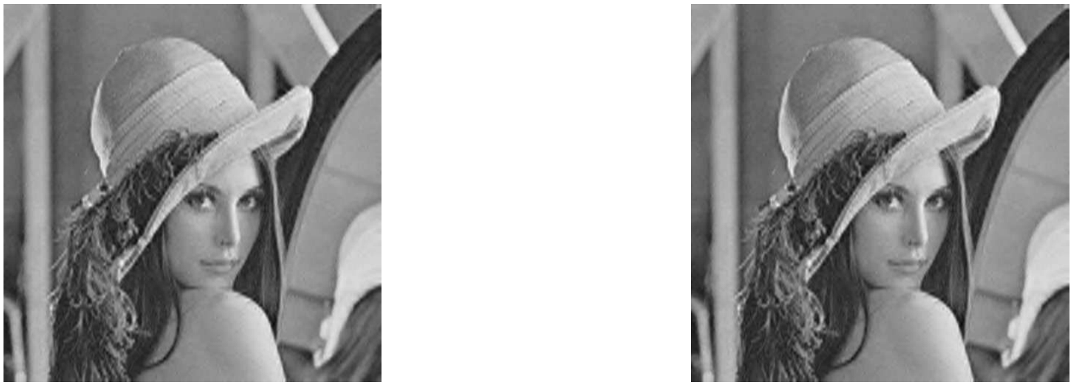
In this paper, a more general framework based on similarity check functions is presented. In our proposed selective-ARQ (S-ARQ) scheme, rather than directly retransmitting all erroneous packets, the degree of corruption is measured using a similarity check function that does not require explicit knowledge of the original data at the receiver. Accordingly, if the packet is found to be badly corrupted based on a specified similarity criteria, it can be considered as a lost packet and recovered using ARQ or hybrid-ARQ. If the applied similarity check function indicates that the degree of corruption of the considered packet is acceptable, the packet is still usable and no retransmission is needed. In this way, the total amount of information needed to be retransmitted can be decreased and, consequently the delay due to retransmission is decreased and more bits can be allocated to the source coding for enhanced quality.

This paper is organized as follows. Section 2 describes the new concept of the similarity check function. This function can be used to intelligently retransmit data for packets that most need to be corrected rather than retransmitting data for all corrupted packets. Section 3 illustrates the proposed selective-ARQ scheme using several examples of similarity check function designs applied with the wavelet-based TCQ image coder presented in [3]. A conclusion is given in Section 4.

## 2. SIMILARITY CHECK FUNCTION

The proposed concept of a similarity check function is motivated by the use of hash functions to provide fault tolerance and security in a distributed storage environment [4]. A hash is computed to uniquely represent a file (or a piece of a file) using only a few bits. It can be thought of as a digital fingerprint of a larger document [5]. Because the hash is much smaller than the original file, it is more efficient to fully protect the hash than the corresponding file.

0-7803-8874-7/05/$20.00 ©2005 IEEE 



(a) No noise, PSNR = 29.88 dB　　(b) BSC with BER= 0.001, PSNR= 27.34 dB

**Fig. 1**. $512 \times 512$ Lena image coded at 0.246 bpp.

The protected hash is sent along with the file. When the file is received, a new hash is computed from that file. If the received file is identical to the original file, the two hashes will match. If the hashes do not match, the received file is discarded. Similarly, our similarity check function is designed such that, when applied to a piece of data (or packet), it results in an output (i.e., an index) that can be represented with an insignificant number of bits relative to the original piece of data, without the need to have the original piece of data. However, while a hash can only indicate whether there is an exact match between the received and transmitted pieces of data, and thus, cannot provide information about the amount of corruption introduced in a file, the proposed similarity check function is designed to measure the degree of corruption introduced. For multimedia applications, a file does not need to be exactly identical to the original data to be usable, which motivates the proposed similarity check function general framework. Note that a CRC code or a hash function can be considered as a special similarity check function with the similarity criteria set to be the identity function.

Let the original data be transmitted (or stored) as N packets (or pieces), $p_1, \cdots, p_N$, each of size $L$. Let $p'_1 \cdots, p'_N$ be the corresponding received packets. Let $P$ denote the set containing all packets (data pieces) of size L. The similarity check function, $S$, is defined as a mapping from the set P to the set $I = \{I_1, \cdots, I_r\}$ as follows:

$$S(p \in P) = I_l,\ l \in \{1, \cdots, r\}, \tag{1}$$

where $I_l$ in (1) is referred to as the similarity check index. Note that only $c = \log_2(r)$ bits are needed to represent elements in the set I, and $c$ is selected to be very small compared to the packet size. Two packets, $p_k$ and $p'_k$, are considered similar if and only if $S(p_k) = S(p'_k)$.

The similarity check function can be designed based on MSE or perceptual criteria. For example, Fig. 1 shows the $512 \times 512$ Lena image compressed with the wavelet-based TCQ image coder of [3] at 0.25 bits per pixel (bpp) and transmitted through an ideal channel with no noise (Fig. 1a), and through the binary symmetric channel (BSC) with bit error rate (BER), $P_b = 0.001$ (Fig. 1b). For this image, 71 bits have been corrupted, resulting in 65 out of 256 $(25.4\%)$ corrupted packets. Despite the fact that the PSNR for the image in Fig. 1b is 2 dB lower than the image in Fig. 1a, both images are very similar. Thus, for multimedia applications, it would be wasteful to discard the image in Fig. 1b because it is not identical to the image in Fig. 1a.

The similarity check function can be used to estimate the amount of protection needed to receive acceptable data. In the proposed selective-ARQ scheme, the similarity check function is used to evaluate the number of packets to be protected. Then, rather than correcting all corrupted packets, only packets that result in a high degree of degradation will be corrected using corresponding retransmitted data. In this S-ARQ method, the decoded data will not be exactly the same as the transmitted data, but annoying noise artifacts can be corrected. Note that, if not enough bits are assigned to fully recover all severely corrupted or lost packets, the similarity check function can also be designed to provide usable information to correct a portion of the lost packets.

## 3. CODING EXAMPLE USING A SIMILARITY CHECK FUNCTION AND SELECTIVE ARQ

In the following coding example, the wavelet-based TCQ coder of [3] is used to code $512 \times 512$ images, which are then transmitted through the BSC with a bit error rate, $P_b$ = 0.001. The coder in [3] has been modified to perform a 4-level dyadic wavelet decomposition to simplify the packetization.

Let $P = \{p_i\}$, $i = 1, \cdots, N$, be the set of all packets that represent the data. As shown in Fig. 2, each packet

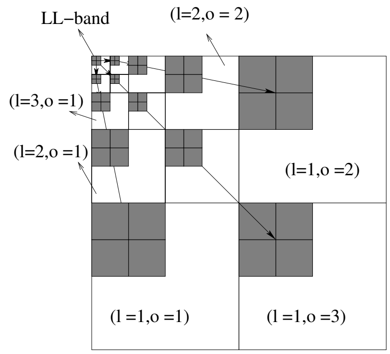



**Fig. 2**. Packetization of wavelet coefficients for a 13 subband decomposition. (l,o) denotes the level and orientation of a child subband.

$p_i$ consists of 4 quantized wavelet coefficients from the LL subband ($b_{i,4,0}$), and its corresponding children subbands $b_{i,l,o}$, where $l = 1, \cdots, 4$, and $o = 1, 2, 3$, denote, respectively, the level and orientation of a child subband. To limit the propagation of bit errors across packets, the TCQ quantizer is re-initialized after coding the LL subband of each packet. The above packetization results in $N = 256$ packets of length $L = 250$ bits for a $512 \times 512$ image.

A similarity check function is designed to measure the degree of corruption in perceptually significant subbands including the LL subband and $k$ of its corresponding children subbands $b_{i,l,o}$, where $l = (4 - (k \text{ div } 3) + 1)$ to 4, and $o = 1, 2, 3$, for all levels $l$ except for the lowest $l$ for which $o = 1$ to $((k-1) \mod 3)+1)$. The parameter $k$ can be varied depending on the noisy conditions of the channel and the available bit budget. The similarity check function $S_k(.)$ evaluated for a packet $p_i \in P$ can be defined as a vector of indices $S_k(p_i) = \mathbf{I}_{pi} = \{I_{p_i,0}, I_{p_i,1} \cdots, I_{p_i,k}\}$, where $I_{p_i,0}$ represents the variability of the 4 LL-coefficients coded in packet $p_i$ forming the block $b_{i,4,0}$ and $I_{p_i,j}$, $1 \leq j \leq k$, represents the variability of the wavelet coefficients in the corresponding children subband $b_{i,l,o}$. Each $I_{p_i,j}$ is computed as follows:

$$I_{p_i,j} = \begin{cases} 1, & \text{if } \max_{x \in b_{i,l,o}}(|x - \mu_{i,l,o}|) > T \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $T$ represents a threshold, $\mu_{i,l,o}$ is the mean of the wavelet coefficients in the child subband $b_{i,l,o}$ corresponding to packet $p_i$, and $j = 3(4 - l) + o$ . In our implementation, $T = 1$ since, in the coder [3], all coefficients are normalized to have a variance of 1.

The proposed S-ARQ-based joint source-channel coding scheme proceeds as follows:

1. At the transmitter side, for each packet, $p_i$, compute the similarity check function $S_k(p_i)$ = $\{I_{p_i,0}, \cdots, I_{p_i,k}\}$, $I_{p_i,j} \in \{0, 1\}$, as in (2). This will result in a $k$-bit output $\mathbf{I}_{p_i}$.
2. Send the resulting $k$-bit similarity check indices, $\mathbf{I}_{p_i}$, to the receiver along with the data packets. Note that $k$ is selected to be very small compared to the packet size. So, these constitute an insignificant portion of the bit budget and can be protected fully without significant added overhead.
3. At the receiver, for each received packet, $p'_i$, compute the similarity check indices $S_k(p'_i) = \mathbf{I}_{p'_i}$ as in (2), and compare with the corresponding transmitted similarity check index, $S_k(p_i) = \mathbf{I}_{\mathbf{p_i}}$. If $S_k(p'_i) \neq \mathbf{I}_{p_i}$, then $p'_i$ is requested for retransmission using ARQ or H-ARQ; otherwise, $p'_i$ is kept and used to reconstruct the image.

In addition, the minimum and maximum values of the entire LL-subband are sent along with the data using few bits and can be used to detect coefficients lying outside of the acceptable range.

Table 1 presents the performance results averaged over 50 simulations using the proposed S-ARQ scheme with the similarity check function $S_k$ for different values of $k = 0, 3, 6$. For comparison, Table 1 also shows results using a conventional ARQ scheme with a 16-bit CRC ($\text{CRC}_{16}$-ARQ) that is applied to fully protect the entire transmitted data, and also with an 8-bit CRC ($\text{CRC}_8$-ARQ-LL) that is applied to fully protect only the most significant LL subband. In Table 1, the number of corrected packets (column 3) is the average number of packets that were detected to be unusable and are retransmitted. The number of parity bits (column 2) is the number of bits that are appended to each packet by the corresponding similarity check function. Fig. 3 shows that a very good image quality can be obtained by only retransmitting a fraction of the corrupted packets. $S_0$ was able to detect the most annoying artifacts due to errors occurring in the LL-subband, whereas using $\text{CRC}_8$-ARQ-LL doubles the number of packets requested for retransmission without improving significantly the quality of the corrected image (Fig. 3c). In comparison, Fig. 3d shows that, for the same number of retransmitted packets as in Fig. 3c, the proposed S-ARQ scheme results in an improved performance in terms of both PSNR and visual quality by protecting perceptually relevant data in higher subbands. Figs. 3d and 3e show that the high frequency noise artifacts decrease as more perceptually significant subbands are protected by the proposed similarity check function, while the number of retransmission is still significantly lower than the conventional ARQ scheme (Fig. 3f).

## 4. CONCLUSION

In this paper, we have presented a selective-ARQ scheme based on the concept of a similarity check function that measures the amount of corruption introduced in an image

(a) Not protected, PSNR = 26.93
check bits per packet : 0 (0%)
Retransmitted packet: 0 (0%)

(b)$S_0$-ARQ, PSNR = 29.58
check bits per packet : 1 (0.4%)
Retransmitted packets: 4 (1.6%)

(c)CRC$_8$-ARQ-LL, PSNR= 29.61
check bits per packet: 8 (3.2%)
Retransmitted packets: 10 (3.9%)

(d) $S_3$-ARQ, PSNR = 29.63
check bits per packet: 4 (1.6%)
Retransmitted packets: 10 (3.9%)

(e) $S_6$-ARQ, PSNR = 29.69
check bits per packet: 7 (2.8%)
Retransmitted packets: 28 (10.9%)

(f) CRC$_{16}$-ARQ, PSNR = 29.82
check bits per packet: 16 (6.4%)
Retransmitted packets: 65 (25.4%)

**Fig. 3**. PSNR (in dB) of $512 \times 512$ Lena source coded at 0.242 bpp over the BSC channel with a bit error rate, $P_b = 0.001$.

**Table 1**. $512 \times 512$ Lena image coded at 0.242 bpp using the wavelet-based TCQ coder in [3], ber=0.001.

| similarity function | parity bits | packets corrected | PSNR in dB |
|---|---|---|---|
| None | 0 | 0 | 28.12 |
| $S_0$ | 1 | 2.0 | 29.31 |
| CRC$_8$ | 8 | 6.97 | 29.41 |
| $S_3$ | 4 | 6.08 | 29.47 |
| $S_6$ | 7 | 18.32 | 29.51 |
| CRC$_{16}$ | 16 | 58.1 | 29.82 |

without explicit knowledge of the original data at the receiver. In the proposed low-delay S-ARQ scheme, not all corrupted packets need to be corrected and retransmitted; only those corrupted packets that results in perceptually annoying artifacts are detected by the similarity check function and retransmitted. Compared to the conventional ARQ scheme, the proposed method results in a reduced delay and improved performance due to a lower retransmission rate.

## 5. REFERENCES


[1] S. Wicker, *Error control systems for digital communication and storage*, Prentice Hall PTR, 1995.

[2] N. Matoba, Y. Kondo, and T. Tanaka, "Still image transmission using ARQ over Rayleigh fading channels," *IEE Electronics Letters*, vol. 32, no. 9, pp. 803–804, Apr. 1996.

[3] T.-T. Lam, G.P. Abousleman, and L.J. Karam, "Image coding with robust channel-optimized trellis-coded quantization," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 940–951, June 2000.

[4] D. MacKay, *Information theory, Inference and learning algorithms*, chapter 12, Cambridge University Press, 2003.

[5] R. Rivest, "The MD5 message digest algorithm," Internet RFC 1321, Apr. 1992.