Unequal Error Protection of Multiple Programs Based on Length-variable Transport Stream Packets

Zhengyi Luo, Li Song, Shibao Zheng
Institute of Image Communication, Shanghai Jiao Tong University, 200240, Shanghai

Abstract

Unequal Error Protection (UEP), which provides important data with more protection, has been proven to be able to produce better quality in image communication. Previous UEP schemes are mostly proposed for single-image or single-program scenarios. Yet few are developed for multiple programs. Inspired by the MPEG-2 Transport Stream (TS), in this paper we transform the conventional TS packets to be length-variable, and propose a new UEP scheme, which is suitable for multiple-program scenarios. A theoretical model for this scheme is built in this paper, and experimental results also demonstrate the effectiveness of the scheme.[1]

1. Introduction

With the development of communication technologies, more and more video services are available nowadays. However, common channels, such as the Internet or wireless channels, are not always reliable. Video streams often suffer from channel errors (such as packet loss) inevitably during transmission, which will degrade image quality drastically. To deal with this problem, several solutions, such as forward-error-correction (FEC), automatic-repeat-request (ARQ), error concealment, etc, have been developed. Among them, FEC is a quite effective method for conquering transmission errors as no feedback channels or extra delay is involved. Based on all kinds of error correction codes, FEC appends decent amount of redundancy information during transmission. When data are corrupted or lost, the redundant information may be used to recover them. Especially, FEC can be closely combined with so-called Unequal Error Protection (UEP), which protect important data with more redundant bits and thus yields better performance.

The UEP are inspired by the idea of priority encoding transmission [1]. For one thing, several schemes about UEP for still images were proposed such as in [2]. In this scenario, firstly an embedded encoder such as the set partitioning in hierarchical trees (SPIHT) [3] is used to encode a still image. Then of the resulting bit streams the most important receive most protection. For another, many UEP schemes suitable for video transmission were also proposed. Some schemes are used to unequally protect frames of different types (I, P or B frames) [4]. Some are used to unequally protect different layers of scalable video streams [5]. And some are used to unequally protect different types of frames and different layers both [6]. However, it may be found that mostly they are developed for single-program scenario. Now there are more and more programs available for users, and it’s very common to transmit multiple programs with different priorities simultaneously. For example, multi-view surveillance becomes more and more popular now, and different view usually exhibits different importance. So UEP schemes suitable for multiple programs are badly needed in such scenarios.

In this paper, based on FEC, we introduce an UEP scheme suitable for multiple programs. The scheme results from the MPEG-2 Transport Stream (TS) [7], which is a widely used method for transmission of multiple programs. But in order to accommodate UEP, we transform the standard fixed-length TS packets to length-variable ones, which are then injected into an interleaver with its length information and FEC codes. As our scheme doesn’t need special assistance from the channel, it can be easily applied with different channels.

The rest of the paper is organized as follows. In section 2, the UEP scheme for multiple programs is introduced in detail. In section 3, the optimal allocation of the channel rate is formulated for the UEP scheme.

[1] This work was supported in part under China Grants NSFC No.60702044.

978-0-7695-3507-4/08 $25.00 © 2008 IEEE
DOI 10.1109/CSIE.2009.605

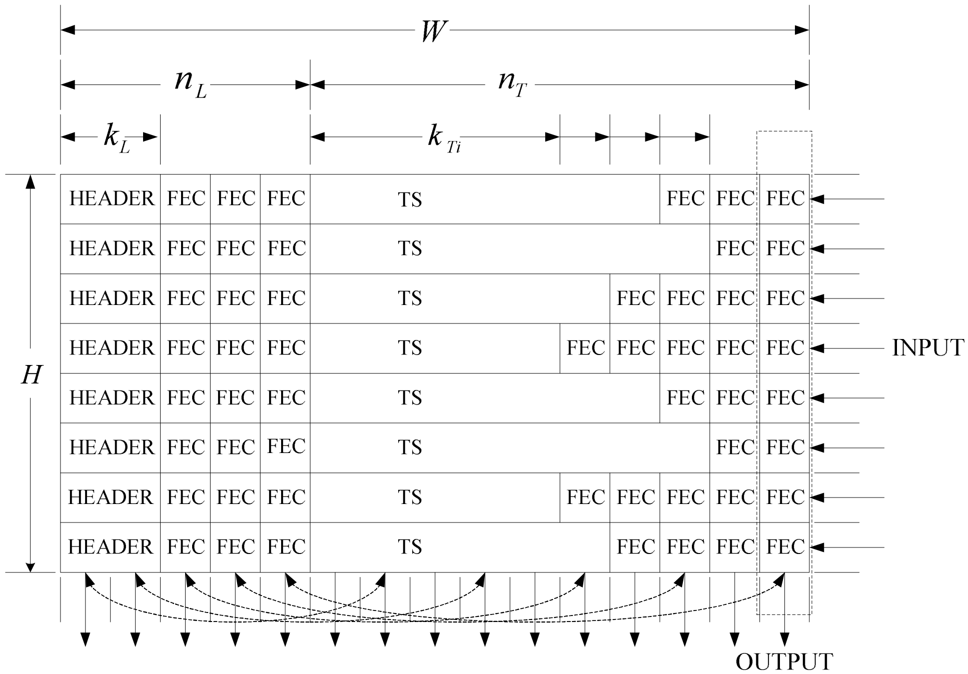

Figure 1. Illustration of the proposed UEP scheme

Experimental results validating the effectiveness of this scheme are shown in Section 4. And Section 5 draws the conclusion.

2. Proposed UEP scheme

In order to transport multiple programs in a single bitstream, several methods have been defined by previous standards. But among those methods, MPEG-2 Transport Stream (TS) undoubtedly enjoys much popularity at present. So now we just develop our scheme based on this. But it should be noted that our scheme is not confined to TS. If possible, streams defined by other standards can easily be joined with our scheme.

MPEG-2 Transport Stream is composed of successive TS packets of equal length. Usually fixed amount of parity bits are appended with every TS packet against channels errors, which is just the Equal Error Protection (EEP) case. But in order to provide different protection to every multiplexed program, here we make the length of TS packets variable. (In this paper, we continue to call the length-variable packet "TS packet".) That is TS heads keep the same, but length of the TS payloads no longer get constrained as before and can vary in a wider range. Then different programs may correspond to TS packets with different length. And when the total length of the TS packet and FEC codes is fixed, different number of FEC codes may be appended for different TS packets. Consequently the purpose of UEP for different programs will be reached. The overall UEP scheme is shown in Fig. 1, where data are inputted to the interleaver by rows and outputted by columns. As TS packets now have been transformed to be length-variable, length information has to be sent for decoding. In previous UEP schemes, decoding-related header information, such as the length of TS packets used in this paper, may assume to be conveyed in another reliable channel. To make our proposed scheme more practical, we transmit the headers, i.e. the length information, in the same channel as data packets. So as shown in Fig. 1, there are corresponding FEC codes for both the headers and the length-variable TS packets in the interleaver. Here we just call the output of the interleaver "units". And when the interleaver is full, a group of outputted units can be transmitted.

As header information is essential to the decoding of length-variable TS packets, before transmission we rearrange the order of the outputted units to make the units related to headers dispersed evenly as shown in Fig. 1. Then in case of burst noise, the header information won't be corrupted easily and robustness of our scheme is improved.

Since header packets and TS packets of different programs need protection with different priority, careful channel rate allocation is necessary to ensure the optimal tradeoff between channel utilization and error robustness. This is presented in the ensuing section.

3. Channel rate allocation for multiple programs

According to MPEG-2 Part I [7], there exist some TS packets containing the Program Specific Information (PSI), which is used to indicate the number of programs and components of every program in the bitstream. Though such TS packets are vitally important for decoding, usually they are very short in terms of our scheme. So in the proposed scheme, they may be allocated with more FEC codes and accordingly received correctly with higher probability. Besides, PSI doesn't change frequently. And TS packets containing PSI are always repeatedly transmitted according to the standard. So when deriving the theoretical model, we always suppose PSI has been received correctly.

Reed-Solomon (RS) codes [8], which are used as the FEC tools in the analytical model, are widely used error correction codes. RS codes are usually described as (n,k), which means there are k source symbols and $n-k$ protection symbols in the total n symbols. When the errors can be positioned, RS codes can recover up to $n-k$ error symbols. But when the location of errors is unknown, only up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ error symbols can be recovered. As in most cases, data are often expressed in bytes. When doing the experiments, we treat every byte as a symbol and use the RS codes in the Galois field of size $2^8 = 256$.

Parameters about the interleaver have been labeled in Fig. 1. Let k_L bytes be reserved for the header information and let TS packets for program i be k_{Ti} bytes long. So the RS codes used for the header information and TS packets for program i are (n_L, k_L) and (n_T, k_{Ti}) respectively. As width of the interleaver W has been preset, the sum of n_L and n_T is constant and always equal to W. Let height of the interleaver be H. Therefore every outputted unit is H bytes long.

Now when deriving the model, we make use of a packet loss channel, supposed that packets drop stochastically. That a length-variable TS packet cannot be correctly decoded may be caused by two reasons. First, its length information may be lost. Second, though the length information is available, the TS packet itself is not received correctly. Let the packet loss rate of the channel be p_l. Then the probability P_i that decoding of the length-variable TS packets for program i goes wrong is

$$P_i = \sum_{m=n_L-k_L+1}^{n_L} \binom{n_L}{m} p_l^{\,m} (1-p_l)^{n_L-m} + \sum_{m=0}^{n_L-k_L} \binom{n_L}{m} p_l^{\,m} (1-p_l)^{n_L-m} \bullet \sum_{n=n_T-k_{Ti}+1}^{n_T} \binom{n_T}{n} p_l^{\,n} (1-p_l)^{n_T-n} \quad (1)$$

Suppose there are M programs, i.e. there are M kinds of different length-variable TS packets here. Let the bit rate for program i be B_i. And let the available bandwidth be $B_{\max}$. To make the bit streams for all M programs transmitted properly, we have to ensure that the total bit rates are not more than $B_{\max}$, i.e.

$$\frac{W}{n_T}(1+\eta)\sum_{i=1}^{M} B_i \frac{n_T}{k_{Ti}} \le B_{\max} \quad (2)$$

where W is width of the interleaver and η stands for the preset ratio that the TS-related overhead (such as PSI and TS heads) may account for.

In order to unequally protect programs, we need to select a group of weighting coefficients $\{w_i\}_{i=1}^{M}$ in advance, with large values assigned to important programs. Then to minimize the probabilities of erroneous decoding for programs according to different priorities, we may build the following constrained optimization problem:

$$\begin{aligned} &\min J = \sum_{i=1}^{M} w_i P_i \\ &s.t. \frac{W}{n_T}(1+\eta)\sum_{i=1}^{M} B_i \frac{n_T}{k_{Ti}} \le B_{\max} \\ &n_L \ge k_L \\ &n_L \le W \\ &k_{Ti} \ge k_{T\min} \ (i=1,2,\cdots,M) \\ &k_{Ti} \le n_T \ (i=1,2,\cdots,M) \\ &n_T = W - n_L \end{aligned} \quad (3)$$

Here $k_{T\min}$ is the preset minimal length for the length-variable TS packets. And the solution of (3) is the vector $\left(n_L, \{k_{Ti}\}_{i=1}^{M}\right)$ that minimizes J, the weighted sum of the probabilities of erroneous decoding for all programs.

Problem (3) essentially belongs to integral programming, which cannot be solved by trivial methods. And we resort to heuristic methods, such as the genetic algorithm [9], in this paper.

4. Experimental results

Now the proposed scheme is evaluated for unequal protection of 3 programs, each of which contains a single H.264/AVC [10] video stream. Here width of the interleaver is $W = 255$, and height is set to $H = 200$. $k_L = 1$, i.e. length of every TS packet can be represented in one byte.

For comparison purpose, 3 programs contain the same H.264/AVC video stream. Two standard 4CIF sequences "soccer" and "crew" containing 300 frames are tested. Group of Pictures (GOPs) of IPPP… structure consisting of 32 frames are considered for each. And both sequences are encoded at 25 fps and 1000 kbps.

Assume we want to put most protection to program 1, median protection to program 2, and least protection to program 3. Let the available bandwidth be 3600 kbps. When allocating the channel rate, we consider the common packet loss rate of 10%. For illustration purpose, we choose significantly different weighting coefficients, setting $w_1 = 10^2$, $w_2 = 1$, $w_3 = 10^{-3}$ empirically. Then the genetic algorithm may be applied with the parameters $p_l = 0.1$, $B_i = 1000 (i = 1,2,3)$, $B_{max} = 3600$, $\eta = 0.1$. After iterating for 2000 times with a population size of 400, we get a near-optimal solution $(n_L = 6, k_{T1} = 198, k_{T2} = 208, k_{T3} = 243)$. In the case of EEP, length of TS packets doesn't change. So no bytes need to be reserved for the header information in the interleaver. Then based on the same bit rate constraint, the optimal length of TS packets for every program is $k_{Ti} = 215 (i = 1,2,3)$. Then in the following experiments, TS packetizing and interleaving are both configured according to the above results.

Comparison of the EEP and the proposed UEP schemes using the H.264/AVC bit streams of "soccer" and "crew" is shown in Fig. 2 and Fig. 3. In experiments, every unit outputted from the interleaver is dropped stochastically according to a preset probability. For the "soccer" example, we may find that all programs are corrupted when packet loss rate reaches 11% or so in the case of EEP. In the case of UEP, program 3, with the least protection, is corrupted as long as packet loss rate reaches 2%. But program 1, the most important one, can be decoded fluently until the packet loss rate is above 15%. The "crew" example gives similar results.

From Fig. 2 and Fig. 3, it may also be observed that the max endurable packet loss rate is slightly less than we expect. This is because a video frame is first packetized into a Packetized Elementary Stream (PES) packet [7], which is then divided and packetized into several TS packets. So during decoding, heads of PES packets are vitally important. If the TS packet containing the PES head is lost, the whole frame will be corrupted even if other TS packets of this frame can be received correctly.

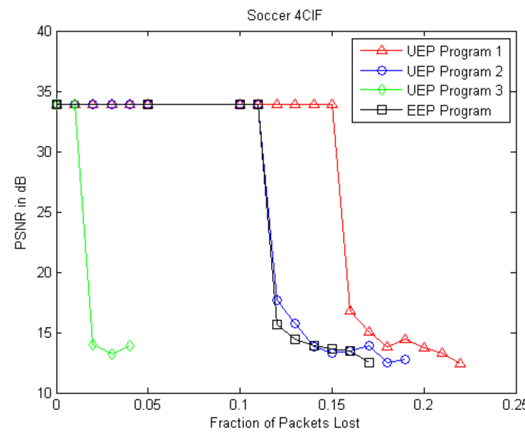

Figure 2. Effect of packet loss on PSNR for UEP and EEP of the "soccer" sequence

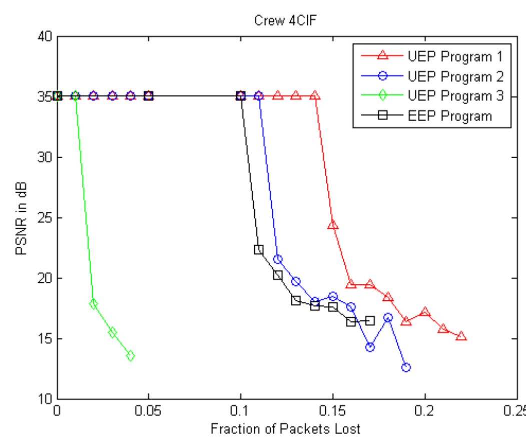

Figure 3. Effect of packet loss on PSNR for UEP and EEP of the "crew" sequence

5. Conclusion

In this paper, we propose an UEP scheme suitable for multiple programs, which can put different level of protection to different programs based on predefined priorities. The scheme transforms MPEG-2 TS packets to be length-variable and makes the UEP for multiple programs possible. The scheme is tested in a packet loss scenario. And its effect is clearly shown by the experiment results. Next, we will adapt this scheme to both packet loss and bit error channels, and make it more applicable for video services in the future.

6. References

[1] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *Information Theory, IEEE Transactions on*, vol.42, no.6, pp.1737-1744, Nov 1996.

[2] 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," *Selected Areas in Communications, IEEE Journal on*, vol.18, no.6, pp.819-828, Jun 2000.

[3] A. Said, and W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.6, no.3, pp.243-250, Jun 1996.

[4] H. Wu, M. Claypool, and R. Kinicki, "A model for MPEG with forward error correction and TCP-friendly bandwidth," *Proc. of the ACM NOSSDAV,* Monterey, CA, USA, pp. 122-130, Jun 2003.

[5] M. van der Schaar, and H. Radha, "Unequal packet loss resilience for fine-granular-scalability video," *Multimedia, IEEE Transactions on*, vol.3, no.4, pp.381-394, Dec 2001.

[6] T. Fang, and L. Chau, "GOP-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks," *Image Processing, IEEE Transactions on*, vol.15, no.6, pp.1323-1330, June 2006.

[7] ISO/IEC JTC 1/SC 29/WG 11, *Generic coding of moving pictures and associate audio information, Part 1: Systems, 13818-1,* Nov 1994.

[8] S. Lin, J. Daniel, and J. Costello, *Error Control Coding (second edition),* Prentice-Hall, NJ, Jun 2004.

[9] X.P. Wang, and L.M. Cao, *Genetic algorithm-theory, application and software implementation,* Xi'an Jiaotong University Press, China, Jan 2002.

[10] ISO/IEC JTC 1/SC 29/WG 11, *Advanced video coding for generic audiovisual services, 14496-10,* Mar 2005.