Robust critical data recovery for MPEG-4 AAC encoded bitstreams

Ruijing Hu, Xucen Huang, Michel Kieffer, Olivier Derrien, Pierre Duhamel

ROBUST CRITICAL DATA RECOVERY FOR MPEG-4 AAC ENCODED BITSTREAMS

Ruijing Hu[1]*, Xucen Huang*[1]*, Michel Kieffer*[1]*, Olivier Derrien*[2]*, and Pierre Duhamel*[1]

[1] L2S, CNRS–SUPELEC–Univ Paris-Sud, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France
[2] LMA, CNRS, Marseille, France

ABSTRACT

This paper presents a bandwidth-efficient method for improved decoding of critical data generated by the MPEG-4 AAC audio coder when encoded bitstreams are transmitted over noisy channels. The critical data of each encoded frame is estimated using the redundancy due to the correlation between successive frame headers and using an optional CRC as an error-correcting code. Simulation results for an AWGN channel show a substantial link budget improvement (more than 5 dB) compared to a classical hard-decoding method. Once critical data are efficiently estimated, the work of previously proposed joint source-channel decoding techniques to recover the remaining parts of the MPEG4 AAC frames is significantly facilitated.

***Index Terms*—** Audio coding, Cyclic codes, Decoding, MAP estimation, Mobile communication, Robustness

1. INTRODUCTION

High-quality audio codecs, such as the MPEG-4 AAC [1], provide high compression efficiency. Nevertheless, since they were designed for data transmission over reliable channels, such codecs are very sensitive to transmission errors. A single bit in error in a packet of compressed data may lead to the loss of the whole packet. Such errors are unavoidable when streaming audio over wireless networks in last generation mobile communication systems, characterized by unreliable transmission channels which introduce noise at bit level and frame losses.

Packetisation of data and protection of the packets with error-detection codes (CRC or checksum) [2, 3] is a first solution to alleviate this problem. Erroneous packets are identified and retransmitted, if possible. Nevertheless, in scenarii with strong delay constraints, *e.g.*, in visiophony, retransmission is difficult and may even become impossible when broadcasting data, *e.g.*, in digital audio broadcasting. In addition to packetization, strong error-correcting codes, *e.g.*, turbo-codes, LDPC, at *Physical* (PHY) layer may be combined with packet-erasure codes at intermediate protocol layers [4, 5]. The redundancy introduced by these codes reduces the bandwidth allocated for the data and may be over-sized in good channel conditions. In bad conditions, some corrupted packets may still not be recovered and are assumed lost. Error-concealment techniques [6, 7] can then be used by the source decoders at *Application* (APL) layer. They exploit the redundancy (temporal and/or spatial) found in the multimedia data for estimating the missing information.

Joint source/channel decoding (JSCD) techniques have been proposed to recover corrupted packets [8]. JSCD takes advantage of various sources of redundancy present in the coding and transmission chain. Residual redundancy may come from to the syntax of variable-length source codes [9, 10, 11, 12, 13], from the semantic of the source coders [14, 15, 16, 17], from the packetization of compressed data [18], and from the protocol stack itself [19], see [8] for more details. Altogether, the various redundancies can attain an unexpected amount, and have the potential of avoiding many packet retransmissions.

In [20], the JSCD of *scalefactors* in MPEG-AAC coded bitstream provides a significant improvement in perceptual signal quality compared to classical decoding method. Nevertheless, this result assumes that critical data (mainly packet headers) in every packet are correctly received. This assumption requires that critical data are better protected against transmission errors.

This paper introduces an alternative approach to recover efficiently critical data in each MPEG4-AAC frame, which are assumed to be only protected by the optional CRC of MPEG4-AAC. A JSCD technique is again used to exploit the correlation between the headers of successive packets, in conjunction with soft information provided by the channel decoders at PHY layer and forwarded to the APL layer using a permeable protocol stack [21, 19] involving joint protocol channel decoding techniques to recover the various headers of the protocol stack. The proposed decoding technique is an adaptation of that proposed in [19], designed for the recovery of header at various levels of the protocol stack.

Section 2 gives an overview of MPEG-4 AAC decoding and identifies the critical data in the MPEG4-AAC bitstreams. Section 3 introduces the JSCD of MPEG4-AAC critical data. Finally, Section 4 presents simulation results before drawing some conclusions.

This work has been partly funded by the NEWCOM++ NoE.

2. CRITICAL DATA OF THE MPEG4-AAC FRAMES

In this study, unreliable transmission channels which generate bit-level noise are considered. Data are segmented in transport-level frames, according to the network protocol [3]. It is assumed that, at the receiver side, reliability information at the output of the PHY layer reaches the APL layer and may be used by the source decoder [19, 22]. This assumption is reasonable because all processing is performed inside the receiver terminal.

The MPEG-4 AAC standard specifies a bitstream format for encoded audio data. The bitstream is segmented in frames, of fixed length in the case of fixed bit-rate encoding, or variable length in the case of variable bit-rate (VBR) encoding. These are source level frames, and one frame does not necessarily correspond to a unique transport level frame.

The simplest AAC bitstream is considered: A monophonic audio signal encoded with the Low Complexity profile. A frame is made up of a fixed header, a variable header, a data block, and a marker indicating the end of the frame. The data block comprises a *global gain* (Gg) field, an *individual channel stream* (ICS) field, a *section* field, a *scalefactor* field, a *spectral data* field, and an optional field, called *data stream element* (DSE). The most critical parts of the frame are the headers, the ICS, and the section fields [23]. When these data are corrupted, decoding is almost impossible. Thus, errors on critical data are usually considered equivalent to a frame loss. The MPEG4-AAC standard provides an optional CRC covering the 192 first bits of the frame, allowing to detect a corruption of most critical data.

In what follows, this CRC is used as an error-correcting code, in conjunction with the residual redundancy left by the MPEG4-AAC coder in the headers it generates and the redundancy due to the correlation between successive headers.

3. MAP ESTIMATOR FOR THE CRITICAL DATA

Consider the n-th incoming MPEG-4 AAC packet. Since the 192 first bits are protected by the CRC $\mathbf{c}$ of ℓ_c bits of MPEG-4 AAC, only some parts of the long frames are protected. Fields protected by the CRC may have various properties, which will be used in a different way as far as the corresponding redundancy is concerned.

Four types of fields protected by CRC may be identified.

- The *constant* fields, represented by the vector $\mathbf{k}_n$, are assumed to be *known*.
- The *predictable* fields are embedded in the vector $\mathbf{p}_n$. In contrast with the known fields, the predictable fields are estimated by exploiting the *correlation* between successive packets, and, if available, information provided by the lower layers of the protocol stack. The provided information is represented by R_n, which will be defined formally in what follows. The predictable fields are assumed to be entirely determined if the preceding packets and if the header of the lower layers have been correctly decoded.
- The important *unknown* fields are collected in the vector $\mathbf{u}_n$. These parameters are either completely unknown or limited to a configuration set $\Omega_u(\mathbf{k}_n, \mathbf{p}_n, R_n)$ whose content is determined by the values of $\mathbf{k}_n$, $\mathbf{p}_n$, and R_n.
- Finally, the vector $\mathbf{o}_n$ contains the *other* fields covered by the CRC. This last part contains less critical data, which may be robustly decoded using specific techniques [20].

The first three fields $\mathbf{k}$, $\mathbf{p}$, and $\mathbf{u}$ are the *critical data* of the AAC frame, and R_n gathers the critical data of the $n-1$-th frame

$$R_n \quad = \quad \{\mathbf{k}_{n-1}, \mathbf{p}_{n-1}, \mathbf{u}_{n-1}\}.$$

To lighten notations, when there is no ambiguity, the packet number is omitted in what follows.

Assume that some soft information $\mathbf{y} = [\mathbf{y}_k, \mathbf{y}_p, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c]$ on $\mathbf{k}$, $\mathbf{p}$, $\mathbf{u}$, $\mathbf{o}$, and $\mathbf{c}$ is received from the lower protocol layer. Since $\mathbf{k}$ and $\mathbf{p}$ are known or may be exactly predicted from the already received data, only $\mathbf{u}$ remains to be estimated. Taking into account the observations $\mathbf{y}$, the knowledge of $\mathbf{k}$, $\mathbf{p}$, and R, as well as the CRC properties, one may obtain the maximum *a posteriori* estimate

$$\widehat{\mathbf{u}}_{MAP} = \arg \max_{\mathbf{u} \in \Omega_u} P(\mathbf{u}|\mathbf{k}, \mathbf{p}, R, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c)$$

for $\mathbf{u}$. After some derivations detailed in [22], one obtains

$$\widehat{\mathbf{u}}_{MAP} = \arg \max_{\mathbf{u} \in \Omega_u} P(\mathbf{y}_u|\mathbf{u}) \Psi(\bullet). \tag{1}$$

with

$$\Psi(\bullet) \quad = \quad \sum_{\mathbf{o}} P(\mathbf{o}) P(\mathbf{y}_o|\mathbf{o}) P(\mathbf{y}_c|\mathcal{F}([\mathbf{k}, \mathbf{p}, \mathbf{u}, \mathbf{o}])), \tag{2}$$

and where $\mathcal{F}$ is the CRC encoding function.

The evaluation of (2) requires summing 2^{ℓ_o} terms for each $\mathbf{u} \in \Omega_u$. This results in many cases to an unmanageable computational complexity. A trellis decoding inspired from [24] has been proposed in [22]. This techniques allows to reduce the evaluation of (2) to $O(2^{\ell_c})$ operations. Splitting the CRC into several parts assumed to be independent allows to further reduce the complexity to $O(2^{\ell_c/M})$ operations, where M is the number of parts in which the CRC is divided. Further simplification introduces some suboptimality in the estimation algorithm.

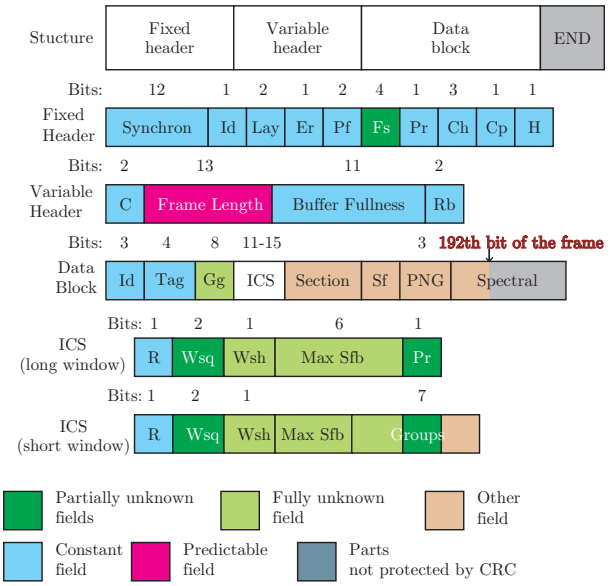

Fig. 1. Fields of an MPEG-4 AAC monophonic audio frame

4. APPLICATION

The JSCD of critical data in a monophonic audio signal is performed using the algorithm described in Section 3.

Figure 1 recalls the structure of an MPEG4-AAC encoded frame. Examples of the four types of fields are provided in what follows. `Synchronisation` is a 12-bit known field, since it contains only ones. `Frame length` is a predictable field, since the length of the frame may be obtained from lower protocol layers. `Fs`, the sampling frequency is a field of 4 bits, which may take any value between 0000 and 1011, it is thus a unknown field with a limited set of values. `Gg`, the global gain is an 8-bit field which may take any value, it is thus fully unknown. Finally, `Section` and `Sf` are examples of other fields, which may be more efficiently decoded with other JSCD techniques than those presented here.

The 5 first seconds of "Tom's dinner" by Suzanne Vega (first 5 seconds, sample rate 48 kHz) have been encoded with an MPEG-4 AAC in a Low Complexity profile at 64 kbits/s. The 192 first bits of each frame are encoded with the CRC specified by the MPEG-4 AAC standard. To simplify simulation, each packet is BPSK modulated and transmitted over an AWGN channel, modeling the actual transmission channel and the various layers of a permeable protocol stack.

Several decoders have been employed to estimate the critical data of each frame. The non-informed hard decoder is a standard hard-input decoder. The informed hard decoder takes into account the knowledge of the known and predictable fields to only estimate the unknown fields via a hard decision on the received bits. In other words, it makes use of the same information as used by our robust decoder. The robust decoder corresponds to an implementation of the estimator (1) using a trellis, as proposed in [22]. The performance of the three decoders is evaluated in terms of frame error rate (FER), evaluated only on the known, predictable, and unknown fields.

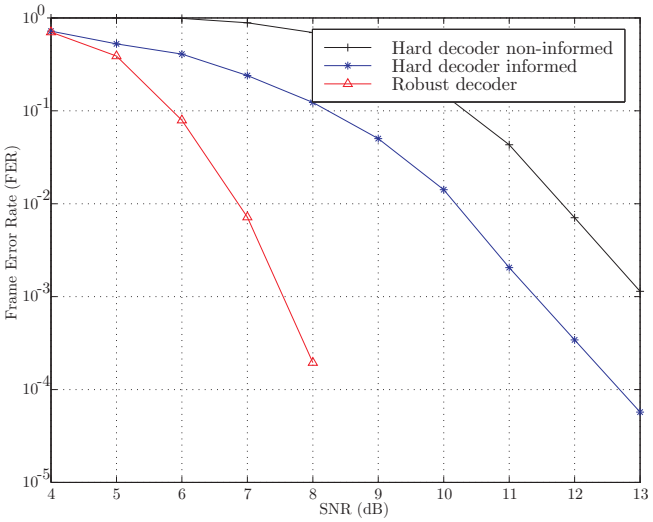

Fig. 2. Frame Error Rate as a function of the SNR for three decoders

Figure 2 shows the frame error rate for different values of SNR. For a FER of 10^{-3}, compared with the hard decoders, from 4 to 5 dB in SNR are gained with the proposed robust decoder, depending on whether the known and predictable fields are exploited by the hard decoder.

In [20], a JSCD of the `Scalefactor` field has been proposed, assuming that the headers were correctly decoded. The JSCD starts to provide satisfying results at 11 dB. One notices that at such values of the SNR, the robust decoder allows to get a vanishing FER. Assuming that headers may be correctly decoded, as done in [20] is thus reasonable.

5. CONCLUSIONS

In this paper, a robust MPEG4-AAC critical data estimator has been presented. This JSCD technique takes advantage of the presence of an optional CRC and of the correlation between successive packet headers. This redundancy allows large parts of the critical data to be known in advance, and to reduce the search set for the unknown parts.

Compared to a classical hard-decoding scheme, the frame error rate (measured on the critical data) is significantly reduced. A gain from 4 to 5 dB is obtained in terms of SNR. If additional JSCD techniques are employed to decode other parts of the MPEG4-AAC frames, the critical data estimator allows these data to be error-free in the SNR region for which, *e.g.*, robust scalefactor decoders work fine, see [20].

The robust decoder presented in this work and that in [20] have now to be combined in order to study the performance

of JSCD techniques on whole MPEG4-AAC frames.

6. REFERENCES

[1] ISO/IEC, "MPEG-4 information technology - very low bitrate audio-visual coding - part3: Audio)," International Organization for Standardization, Tech. Rep. 14496-3, 1998.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1984.

[3] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd ed. Boston: Addison Wesley, 2005.

[4] T. Richardson and U. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[5] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press, 2003.

[6] W.-Y. KUNG, C.-S. KIM, and C.-C. J. KUO, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE transactions on circuits and systems for video technology*, vol. 16, pp. 789–802, 2006.

[7] M. C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Processing: Image Communication*, vol. 14, pp. 473–492, 1999.

[8] P. Duhamel and M. Kieffer, *Joint Source-channel Decoding: A Cross-layer Perspective With Applications in Video Broadcasting*. Academic Press, 2009.

[9] V. Buttigieg and P. Farrell, "On variable-length error-correcting codes," in *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, 27 June-1 July 1994, p. 507.

[10] J. Hagenauer, "Source-controlled channel decoding," *IEEE trans. on Communications*, vol. 43, no. 9, pp. 2449–2457, 1995.

[11] S. Kaiser and M. Bystrom, "Soft decoding of variable-length codes," in *Proc. IEEE ICC*, vol. 3, New Orleans, 2000, pp. 1203–1207.

[12] L. Perros-Meilhac and C. Lamy, "Huffman tree based metric derivation for a low-complexity soft VLC decoding," in *Proc. IEEE ICC*, vol. 2, 2002, pp. 783–787.

[13] R. Thobaben and J. Kliewer, "On iterative source-channel decoding for variable-length encoded markov sources using a bit-level trellis," in *Proc. IV IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'03)*, Rome, 2003.

[14] T. Tillo, M. Grangetto, and G. Olmo, "A flexible error resilient scheme for jpeg 2000," in *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, 29 Sept.-1 Oct. 2004, pp. 295–298.

[15] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet, "Robust vlc sequence decoding exploiting additional video stream properties with reduced complexity," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, June 2004, pp. 375–378, taipei, Taiwan.

[16] C. Bergeron and C. Lamy-Bergot, "Soft-input decoding of variable-length codes applied to the H.264 standard," in *Proc. IEEE 6th Workshop on Multimedia Signal Processing*, 29 Sept.-1 Oct. 2004, pp. 87–90.

[17] G. Sabeva, S. Ben-Jamaa, M. . Kieffer, and P. Duhamel, "Robust decoding of h.264 encoded video transmitted over wireless channels," in *Proceedings of MMSP*, Victoria, Canada, 2006.

[18] C. Lee, M. Kieffer, and P. Duhamel, "Soft decoding of VLC encoded data for robust transmission of packetized video," in *Proceedings of ICASSP*, 2005, pp. 737–740.

[19] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel, "Robust header recovery based enhanced permeable protocol layer mechanism," in *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications, 2008. SPAWC*, 6-9 July 2008, pp. 91–95.

[20] O. Derrien, K. Kieffer, and P. Duhamel, "Joint source/channel decoding of scalefactors in mpeg-aac encoded bitstreams," in *Proc. European Signal Processing Conferences (EUSIPCO)*, 2008.

[21] H. Jenkac, T. Stockhammer, and W. Xu, "Permeable-layer receiver for reliable multicast transmission in wireless systems," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 3, 13-17 March 2005, pp. 1805–1811.

[22] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel, "Robust mac-lite and soft header recovery for packetized multimedia transmission," *IEEE Trans. on Communications*, 2010, to appear.

[23] J. Korhonen and Y. Wang, "Schemes for error resilient streaming of perceptually coded audio," in *Proc. International Conference on Multimedia and Expo (ICME '03)*, vol. 3, 6-9 July 2003, pp. 165–168.

[24] J. K. Wolf, "Efficient maximum-likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. 24, no. 1, pp. 76–80, 1978.