A Forward Error Recovery Technique For MPEG-II Video Transport

R. Radhakrishna Pillai
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613

B. Prabhakaran
School of Computing
National University of Singapore
Singapore 119260

Qiu Qiang
School of Computing
National University of Singapore
Singapore 119260

pillai@krdl.org.sg

prabha@comp.nus .edu.sg

qiuqiang@comp.nus.edu.sg

1 INTRODUCTION

Digital video is increasingly being used as an important medium of information. MPEG-II is widely used for compression of digital video in order to reduce the network bandwidth and storage requirements [4]. Digital video, when used in networked multimedia applications, suffers from data losses/errors. This is a serious problem in the case of wireless networks [7]. There are several ways to recover from these losses or errors. Recovery mechanisms based on re-transmission of the data may not be suitable in many cases because of the real-time nature of the applications and the absence of reverse channel for feedback. Real-time communication of digital video, as in the case of video conferencing, benefit from forward error correction/recovery techniques. Forward error correction (FEC) codes and frequent synchronizing codewords have been proposed in [5, 6]. These techniques require low complexity hardware, however they add redundancy thereby lowering the coding efficiency. FEC can potentially add more errors to a bit stream once the correcting capability of the code has been exceeded. An error-resilient entropy coding (EREC) technique has been suggested in [2, 3, 1]. EREC achieves bit stream resynchronization. [2, 3, 1] also propose a hierarchical pyramid predictor in place of the standard differentially coded DPCM. These proposals consider channels subject to burst errors and packet errors and aim at providing an environment where performance degrades gracefully with increasing channel noise. Several simple error resilient approaches are suggested in [7]. These approaches include error concealment, temporal localization by sending extra I-frames, and spatial localization by providing early resynchronization of the elements in the bit stream that are coded differentially between MPEG-II macro-blocks.

ACM Multimedia '99 (Part 2) 10/99 Orlando, FL, USA

In this work, we consider the MPEG-II system streams: elementary stream (ES), packetised elementary stream (PES) and transport stream (TS) structures. While MPEG-II PES has a variable length (in terms of bits), TS has fixed length. These streams, as explained in Section 2, are mapped in such a way that PES headers always occur at the start of the TS packets. This is done to help the video decoder to easily synchronize on the PES header. Our approach is to use the space that *might* be available in a TS packet (due to its fixed length) to store header information, to facilitate forward error recovery of MPEG-II video. Thus, the proposed technique does not consume additional network bandwidth.

2 MPEG-II SYSTEMS

Figure 1 illustrates the different streams composing the MPEG-II systems layer [4, 8]. The output of the MPEG-II video/audio encoder is the *elementary stream*. This elementary stream is usually organized into *access units*. An access unit is a frame or picture in the case of a video stream or an audio frame, in the case of an audio elementary stream. This elementary stream is now mapped onto a *packetized elementary stream* (PES), that consists of PES packets. As shown in Figure 1, each PES packet has a PES header followed by a variable size payload. This payload can be an exact access unit (which is a frame or a picture in the context of video) of the elementary stream. The PES packet is then converted into *transport stream* (TS) packets, also consisting of a header and a payload. TS packets, as can be seen from Figure 1, are of fixed length. MPEG-II defines this length to be 188 bytes. This length was chosen with ATM and ATM Adaptation Layer (AAL-1) as possible transport protocol layers in mind. A TS packet maps exactly into the payload of four ATM cells. An ATM cell has 48 bytes of payload, but one byte of the payload is used for the AAL-1 protocol (i.e., $4 \times 47 = 188$).

3 FORWARD ERROR RECOVERY

Mapping of a PES packet onto the TS packet is done in such a way that if the data of a PES packet is not completely filling the TS packet, then MPEG-II stuffs dummy bytes (hex FF) into it. The start of the next PES packet is then put into the next TS packet, as shown in Figure 1. Our approach is to use the space that is stuffed with dummy bytes, for replicating important information that will help in a better frame decoding. Replicated information include PES header and TS headers, depending on the space available for replication. Space available in TS header is usually proportional to (PES_Packet_Length mod 184) (since 4 bytes are used by the TS header as part of the 188-bytes sized TS packet). It is obvious immediately that space may not always be available in the TS packet for replicating header information. However, in reality the possibility for replication is quite high, as we found in our implementation experience (Section 4).

Replication at the Encoder: Replication at the encoder has to be done explicitly. Space available for replication in the last TS packet is proportional to (PES_Packet_Length mod $TS\_PayLoad\_Size$). Depending on the space available, the encoder can replicate the PES and/or TS headers. If space is available, encoder replicates header information and represents the length of the replicated data as the last byte in the last TS packet of the PES stream.

Recovery at the Decoder: In case of any corruption or packet loss, decoder should identify whether any header replication was carried out and if so, recover the header information from replicated data. Before describing the recovery process, we first describe the structure of the PES and TS headers, shown in Figure 2. PES header has a PES_packet_length field that specifies the length of the payload (i.e., the size of the elementary access unit, a frame). If (PES_Packet_Length mod $TS\_Payload\_Size \neq 0$), it implies that header replication might have been done by the encoder. Then the decoder needs to identify the last TS packet to recover the replicated information. Identification of the last TS packet is slightly tricky since the MPEG-II transport stream is designed to possibly multiplex different elementary streams. PID (Packet Identifier) is a field in the TS header (in Figure 2(a)) that describes the PES to which the TS payload belongs. TS header also has a *continuity_counter* field that specifies the sequence number of the TS packets for a particular PID. Using these fields and the PES_Packet_Length field in the PES header, the decoder can identify the last TS packet whose last byte gives the length of the replicated information. Decoder can then use this information to reconstruct the frame data.

4 IMPLEMENTATION EXPERIENCE

This forward error recovery approach has been implemented and tested. The augmented coder and decoder currently runs on Windows platform. The coder takes in stored, digitized video and replicates headers in TS packets, if possible. The decoder reconstructs the frame data, with the help of replicated data, if necessary, and displays the frames through Microsoft Directshow. The program for the augmented coder and decoder, along with a demonstration video clip is available for experimentation at
http://www.comp.nus.edu.sg/~qiuqiang/proj/code.html.
Testing of the implemented code was done by randomly *dropping* TS packets at the source, after encoding the frames with replication. On an average, encoder was able to find replication space of 110 bytes (this number obviously depends on the digitized video used for testing). Decoder was able to reconstruct frames whenever TS packets with replicated data were received. Quality of the reconstructed frames depend on the number of bytes used for replication. We are currently in the process of quantitatively measuring the performance in terms of the TS packet loss, reconstructed frames, and quality of the reconstructed frames.

References

[1] R.E.M. Swann and N.G. Kingsbury, "Bandwidth Efficient Transmission of MPEG-2 Video Over Noisy Mobile Links", Signal Processing, Vol. 12, No. 2, April 1998, pp. 105-115.

[2] R.E.M. Swann and N.G. Kingsbury, "Transcoding of MPEG-2 for Enhanced Error Resilience to Transmission Errors", Proceedings of the International Conference on Image Processing, pp. 813-816, September 1996.

[3] D.W. Redmill and N.G. Kingsbury, "The EREC: an Error Resilient Technique for Coding Variable Length Blocks of Data", IEEE Transactions on Image Processing, 4(5), pp. 565-574, April 1996.

[4] ISO/IEC JTC1/WG11 N0702 Rev Generic Coding of Moving Pictures and Associated Audio.

[5] O.A. Aho and J. Juhola, "Error Resilience Techniques for MPEG-2 Compressed Video Signals", Proceedings of the International Broadcasting Convention, IEE, September 1994.

[6] E. Ayanoglu, P. Pancha, A. Reibman, and S. Talwar, "Forward Error Control for MPEG-2 Video Transport in a Wireless ATM LAN", Proceedings of International Conference on Image Processing, pp. 833-836, September 1996.

[7] J. Zhang, M.R. Frater, J.F. Arnold, and T.M. Percival, "MPEG2 Video Services for Wireless ATM Networks", IEEE Journal on Selected Areas in Communication, Vol. 15, No. 1, January 1997.

[8] M. Orzessek and P. Sommer, *ATM & MPEG-2: Integrating Digital Video Into Broadband Networks*, Prentice Hall PTR, New Jersey, 1998.

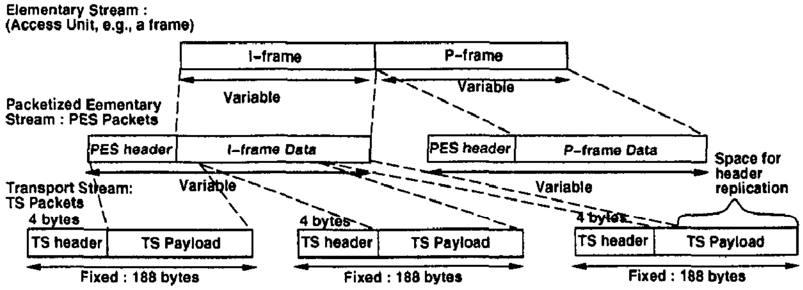

Figure 1: **MPEG-2 Systems Layer**

8 bits	1 bit	1 bit	1 bit	13bits	2 bits	2 bits	4 bits
sync_byte	trans_error_ indicator	payload_unit_ start_indicator	trans_ priority	PID	trans_scra mbling_ctrl	adaptation_ field_ctrl	continuity_ counter

4 bytes

(a) Transport Stream Packet Header

packet_start_ code_prefix	stream_ id	PES_packet_ length	Indicators	Flags

Variable Size

(b) PES Packet Header

Figure 2: **TS and PES Header Structures**