

CCITT SGXV
Working Party XV/1
Experts Group for ATM Video Coding

Document AVC-387
November 1992

Title: Discussion of Priority Transport Option for MPEG-2

Purpose: Discussion

Authors: Kuriacose Joseph
Joel Zdepski
Regis Saint Girons

On behalf of: David Sarnoff Research Center, Princeton NJ.
Thomson Consumer Electronics, Los Angeles CA.

1. Introduction:

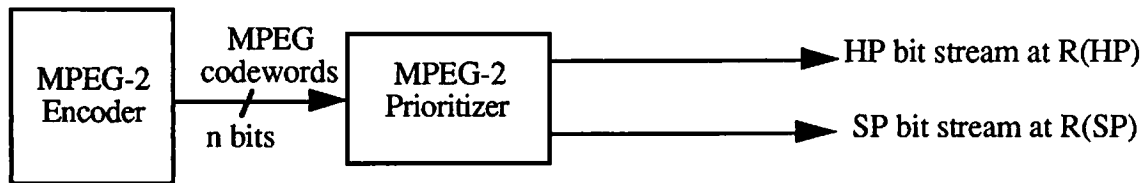
This discussion addresses the issue of transmitting MPEG-2 in a system with two available channels, one with significantly better channel loss characteristic than the other. Such two-tier transmission of MPEG video is considered useful for scenarios in which a minimum decoded quality must be maintained during periods of high error rate and/or network congestion. In general, it has been demonstrated that two-tier MPEG systems can tolerate a relatively high level of channel impairment while still providing a useful image during channel/network outage.

An example of two-tier MPEG transport is the AD-HDTV terrestrial broadcasting system [1], in which there are separate high priority (HP) and standard priority (SP) channels with different bit error rates. The video bit rates in these two channels are approximately in the ratio 1:4. Another scenario relates to ATM network transmission with CLP priority on a specified fraction of the video cells.

It is recognized that scalability syntax under discussion for MPEG-2 may address the requirements for robust two-tier transport in some cases. However, a separate mechanism for splitting the MPEG stream into HP/SP streams is desirable to address situations where: (a) a non-hierarchical coding algorithm is used for maximum compression efficiency; or (b) where the scalable syntax provides a lower tier with bit-rate greater than that supported by the available HP channel.

Earlier papers have presented the transport system within which priority processing was implemented in AD-HDTV [2] and also a suggested syntax for supporting priority processing directly in MPEG-2 [3]. This document presents a discussion of how priority processing can be implemented in a practical MPEG-2 system and describes how the bitstreams can be synchronized (or resynchronized when there are losses) at the decoder using the priority syntax.

2. Prioritization of MPEG-2



Transmission of Prioritized MPEG-2



Reception and decoding of Prioritized MPEG-2

Fig. 1. Architecture for implementation of prioritization

Fig. 1 is a simplified system diagram of an MPEG-2 system with priority processing. Two fixed bit-rate channels are available for transmission, called the HP and SP channels. The HP channel has a significantly lower loss rate than the SP channel. The interface between the MPEG-2 encoder and the MPEG-2 prioritizer is at an MPEG codeword level. At the decoder, the VLD process merges the two bitstreams to provide the decoder with the MPEG codewords for making pictures. Loss in channel data is detected at the transport layer and resolved by the VLD by transmission of "error tokens" to the decoder.

There are two aspects to designing a priority system.

1. Priority syntax

Implementation of priority processing requires definition of a priority syntax, including:

- i. Classification of MPEG data, based on "importance" for decoding, into priority classes. Slice headers and higher layers form the highest priority class. Intermediate Macroblock level information is transmitted with intermediate priority classes assignments. At the DCT coefficient level, codewords are assigned a priority class in accordance their index in the zig-zag scan.
- ii. A "Priority Breakpoint" (PBP) field needs to be defined as a part of the Slice Header. The recommendation is to have 8 bits in the standard. A recommended syntax is shown below.

```
slice() {
    slice_start_code    32      bslbf
    quantizer_scale    5       uimsbf
    PBP                 8       uimsbf
    etc.....
}
```

These slice headers are transmitted in their entirety on the HP channel. On the SP channel the data for each slice is preceded only by the corresponding slice start code, to serve as a marker for

synchronization. The above syntax constrains the priority processor to transmit all data at the slice header level and above always on the HP channel. Note that the overhead involved in this 2-tier approach is 40 bits per slice.

2. Prioritization Procedure

Priority algorithms need to be specified to split a single MPEG codeword input stream into two streams, such that each of the bitstreams generated meets the bit rate requirements of its channel, (i.e. overflow or underflow conditions should be avoided on the channels; occurrence of such conditions leads to inconsistency in computing VBV_delay and/or data loss). In general, the goal of the priority algorithms should be to steer data that is considered more important for subjective image quality to the HP channel. A simplistic priority processing algorithm steers equal fractions of all frames, consistent with the bit rates $R(HP)$ and $R(SP)$, to the two channels, but this results in poor image quality under losses. A more sophisticated priority processor algorithm tends to favor more data for the I-frames on the HP, motion vectors and some low frequency DCT coefficients for the P-frames on the HP and almost all data for the B-frames on the SP channel since it is not used for prediction, with the details of the data breakup depending on the values of $R(HP)$ and $R(SP)$.

Note that the entire priority processing is carried out independent of the rate control algorithm in the encoder. The resource to be exploited is the relative HP and SP encoder buffer fullness. With a simple priority algorithm the HP and SP buffer fullness (relative to $R(HP)$ and $R(SP)$) will be the same. With more sophisticated algorithms the HP and SP buffer fullness vary with respect to each other depending on the frame-type being processed.

A two step process is suggested for implementing priority processing:

i. At the beginning of each frame, based on: (i) HP and SP buffer occupancies, (ii) an estimate for the number of bits in the frame, and (iii) frame type, the desired fraction of HP bits (HPF) for that frame is computed.

ii. In the second step of the procedure, individual slices of the frame are processed and MPEG codewords in each slice are directed to the HP or SP channel in such a manner as to meet the HPF requirements for the frame. This second part of the prioritization procedure (also called Priority Assignment in [1]) is where the priority breakpoint is computed and MPEG codewords transmitted on the channels. An algorithm is required to obtain the appropriate priority breakpoint. One approach is illustrated in Fig. 2. A process of accumulation of bits for each priority class in the slice is carried out. The PBP is computed as that class for which the partial sum of these bits, in the order of decreasing importance of the priority classes, meets HPF. Once a priority breakpoint is computed for a slice, the MPEG codewords are directed to one of the two available bitstreams depending on the priority class of the particular codeword. Data with a priority class greater than or equal to the PBP goes on the HP channel and all other data goes SP.

Note that the computation and processing speed required in the priority processor relates to the MPEG codeword rate and not the pixel rate. Note also that the decoding process is completely independent of the algorithm selection approach at the encoder and depends only on consistency of the syntax definition.

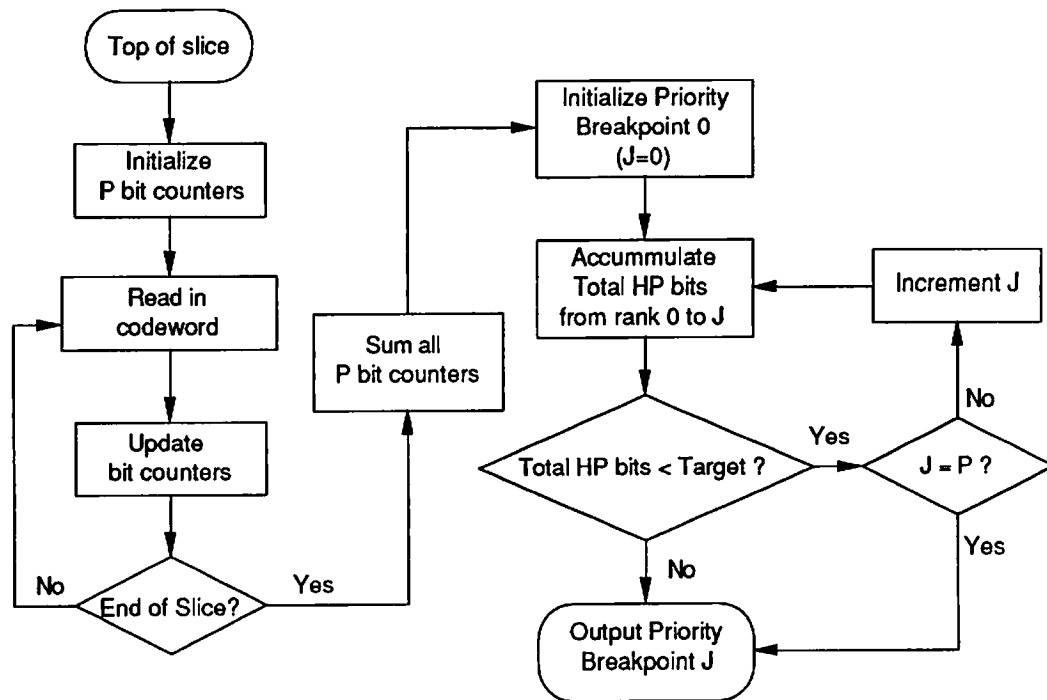


Figure 2: Flow Chart of Priority Breakpoint Computations

3. Decoding of prioritized bitstream:

At the decoder, the merging of prioritized bitstreams is achieved during the process of variable length decoding (VLD). The VLD examines the decoding state and compares its state to the PBP that it extracts from the slice headers in the HP bitstream. If the class of the next codeword to be decoded is greater or equal to the PBP, the decoder uses the HP bitstream, else it uses the SP bitstream. The VLD signals data loss to the video decoder. If the loss is in the SP bitstream, bitstream and video synchronization is still maintained in the HP stream, and decoding continues with the HP data only. An SP error token is transmitted to the decoder whenever the VLD does not have the SP data corresponding to the HP data being forwarded. The SP bitstream is resynchronized by the VLD at the next available slice using the slice start code transmitted on the SP channel for aligning the two bitstreams. The simplest error concealment that the decoder can do during SP losses is to insert an EOB at the end of the HP data it receives, and to reconstruct low resolutions DCT blocks. If no DCT codewords were received the macroblock type could be converted to a type that contains no DCT coefficients. If the error is in the HP bitstream, the next available start code on the HP channel is sought by the VLD and HP decoding resumes from that point. Once HP resynchronization is established the SP channel is aligned to the HP channel. HP losses are indicated to the decoder using appropriate HP error tokens. The error concealment will be more sophisticated than for SP losses only.

Within the context of a higher level transport protocol (such as the AD-HDTV adaptation layer [4], or video AAL in ATM systems), it is also possible to recover synchronization of the SP bitstreams at a macroblock rather than slice boundaries. This leads to effective recovery of more data than with the simple MPEG-2 transmission approach, and hence leads to better decoded pictures under losses.

4. Interfaces

The critical interfaces are those between the encoder and prioritizer and that between the VLD and decoder. For simplicity of implementation of the various subsystems, the easiest interface to deal

with is at an MPEG codeword level, i.e., the interface represents the MPEG codewords. An example interface is a parallel bus interface of n bits, in which k bits represent the "MPEG-type" of the data being passed and the other $(n-k)$ bits are a representation of its value. There are several possible configurations of this interface based on what is easy to implement in VLSI. At the other extreme, a very simplistic design could actually be based on a modified compressed MPEG bitstream level interface so that the encoding/decoding and priority processing/VLD modules are completely separated. This implementation however results in the need for additional hardware: a partial VLD process will have to be done at the priority processor on the encoder side, and a VLC capability is required at the VLD on the receive side.

On the encoder side, the MPEG-type and value fields received at the priority processor are first used to generate the partial bit accumulations for computation of PBP, and then to actually format the compressed data for transmission on the HP or SP channel.

As illustrated in Fig. 1, the decoder side needs an additional bit in the VLD to decoder interface to handle the occurrence of channel losses. This bit would normally remain inactive for actual MPEG codewords and goes active to indicate that an error token is being transmitted to the decoder. The error token codeword makes use of the n bits in the interface somewhat differently than an MPEG codeword. One bit is required to indicate which channel the data loss occurred in. If appropriate a PBP value could be transmitted to the decoder to aid in error concealment. The exact nature of the error-token bits at the interface is also a function of the philosophy used in the design. In one approach one may choose to indicate loss of data for every single decoding entity (i.e. block or macroblock) that the decoder expects. In another approach the VLD just indicates the location of recovery of synchronization in the bitstream and leaves the logic for determining the part of the picture for which data is lost to the decoder.

5. Synchronization of prioritized bitstreams with losses

It is clear that the VBV_delay values are unreliable for synchronization on channels with data erasures. In order to maintain decoder synchronization in this case it is important to have some method of indicating the absolute decode time for each frame. In AD-HDTV just-in-time timing packets were transmitted that served the purpose. In MPEG-2 a method of distributing clock and indicating absolute decoding time based on that clock is probably a good solution. The decode time can be handled at the higher level, e.g., using picture level user data [5][6], while the distribution of the clock is a link/network level function.

6. References:

- [1] Advanced Digital Television: Prototype Hardware Description, FCC Working Party 1 Final Certification Document, Feb. 1992.
- [2] Saint Girons et. al., Transport and Error Concealment for MPEG-2S, MPEG91/311, Kurihama, Nov. 1991.
- [3] K. Joseph, et.al., Proposal for MPEG-2 Syntax to Support Prioritization, MPEG92/ 353, June 1992.
- [4] K. Joseph, et. al., MPEG++: A Robust Compression and Transport System fo Digital HDTV, Image Communication, August 1992.
- [5] J. Zdepski, et. al., An MPEG Video user Data and its Syntax, MPEG92/352, June 1992.
- [6] Picture Header Modification for Source Clock Recovery, AVC-336, September 1992.