INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2013/M31446
October 2013, Geneva (CH)

Source **Telecom ParisTech, Thomson Video Networks, TDF, INSA-IETR on behalf of the H2B2VS project**
Status **For consideration at the 106nd MPEG meeting**
Title **Input on WD2 of 13818-1 AMD6**
Author Jean Le Feuvre, Patrick Gendron, Anne-Laure Mevel, Jean-François Travers, Pascal Dupain, Mickael Raulet

1 Introduction

In 105th MPEG Meeting in Incheon, a stable syntax for carriage of media timeline and external resource location has been proposed in W13661. Following the meeting, a number of discussions on the topic have taken place in the H2B2VS[1] and other standard organizations such as DVB or HbbTV, and some companies have raised interest on the general topic and concerns on bandwidth overhead. This contribution reviews the bandwidth overhead introduced by TEMI and proposes an alternate transmission mode of timing.
This contribution also proposes fixes and improvements to the working draft.

2 Bandwidth Requirements Overview

2.1 PMT Signaling

The TEMI design is to declare a PES stream carrying all needed information in the PMT and have its timestamps rewritten by network devices when PCR gets adjusted, without requiring modification of the TEMI payload. A TEMI stream is assigned streamType 0x25 and can be signaled in the PMT without any further descriptor. Therefore the cost of TEMI stream declaration is (8+3+13+4+12) = 5 bytes, which can be neglected.
There were discussions within H2B2VS whereas this could be problematic, however not signaling the stream in the PMT would bring more complications than benefits:

- a reserved PID should be used for TEMI, thereby requiring PID inspection to detect the presence of TEMI
- timing info from different programs would have to be carried on the same PID, which is very problematic for transcoders which may induce different delays on different programs

[1] [1] H2B2VS is a Celtic+ project which aims at investigating the hybrid Broadcast Broadband distribution of TV programs and services, using HEVC as a compression technology. It is coordinated by Thomson Video Networks. For more information, visit http://www.celtic-initiative.org/Projects/Celtic-Plus-Projects/2012/H2B2VS/h2b2vs-default.asp

After careful investigations, we believe the proposed approach in the WD is the safest and simplest one for the existing infrastructure.

2.2 Timecode Transport

The current TEMI design is to send all timing and service information in a PES packet, with a description compact enough so that this PES packet fits in a single TS packet. As a reminder, using both a TEMI timeline descriptor and location descriptor in one single TS gives:
188 – 4 (TS hdr) – 9 (PES hdr inc. PTS) – 1 (TEMI first byte) - 11 (Timeline Desc) – 6 (basic fields of location descriptor) = 157 bytes for the location descriptor, for which 6 bytes are used by default, thus leaving 151 bytes to describe URL data.

Obviously, there is no requirement that each packet carries both a timeline and URL descriptor, however since the payload has to be carried in a TS packet, the minimal bitrate for sending media timestamps is the bitrate of one TS packet, 1.5 kb. If we consider p60 video for which each frame timestamp has to be exactly matched to external media timeline, this induces and overhead of: 60*1.5kbps = 90 kbps. If we now consider a typical service multiplex with 3 to 6 programs, we have an overall signaling of 270 kbps up to 540 kbps, which may fit quite a few digital services such as radios or embedded applications.

There have been questions regarding whether the TEMI information could be carried inband with the video streams. While obviously this makes little sense for location descriptors, it is worth investigating for timeline descriptor. As per 138181-1 Annex H, the only places where inband data can be injected in a PES stream are:

- adaptation field of the TS packet
- PES private_data of the PES header

The adaptation field is quite flexible and may be set large enough to contain TEMI timeline info. However, the adaptation field is usually processed and discarded at the demultiplexer level and might not be kept in typical transmuxing equipment, which is quite problematic. On the other hand, the PES header will pass a transmux equipment and is likely to be passed to a transcoder; some standard organizations have used PES_private_data to carry some specific information on some PES streams, such as supplemental audio services (DVB TS 101 154 Annex E), and passing PES_private_data through the transcoder might therefore be supported by most equipment.

Before investigating how to carry this information in PES_private_data, let's understand the benefit of this: signaling of PES private_data costs one byte, and the PES private_data field contains 16 bytes. In other words, we only need 17 extra bytes per frame rather than 188.

If we consider that the description part of TEMI is sent every second, this gives for p60 video:
8*(188 + 17*60) = 9.5 kbps
hence from 29 kbps to 58 kbps per multiplex

If we consider that the description part of TEMI is sent twice every second, this gives for p60 video:
8*(2*188 + 17*60) = 11 kbps
hence from 33.5 kbps to 67 kbps per multiplex

We believe that the gain is important enough to integrate this possibility in the current WD before going to CD.

One important thing is to make sure usage of TEMI in PES private_data does not break existing deployments, but from the best of our knowledge systems using PES private_data use escape mechanism in the first byte(s) to identify the private data type.
The feedback we had from encoders makers was that reformatting content in a different PID than the video one would be tricky and add complexity to the systems, whereas on the other hand extracting a 128 bit from the video PES header and copying it over after encoding would not be such a heavy task. This is very important since encoders may change the GOP structure on the fly and will likely need to modify the TEMI timeline.

3 Proposal

3.1 Fixes in existing WD

The current working draft defines splicing flags and timeline_id in the location descriptor. However, if we want to send the timing information at a higher frequency that the location (typically because the location descriptor is big enough to result in a TEMI AU spanning across several TS packets), we face several design issues:

- when receiving the TEMI TimelineDescriptor, we do not know which timeline it describes, since this info is given in the LocationDescriptor. It therefore has to be also included in the TimelineDescriptor.
- The force_reload flag should also be present in the TimelineDescriptor, as a service provider may want to trigger reloading of the HTTP resource without having to send back the URL

Add title to the Table

In table T-5 replace

Syntax	Nb bits	Mnemonic
has_timestamp has_ntp_timestamp has_timecode reserved	2 1 2 3	

With

Syntax	Nb bits	Mnemonic
has_timestamp has_ntp_timestamp has_timecode force_reload reserved timeline_id	2 1 2 1 2 8	

And add to semantics

`force_reload`: 1-bit flag indicating that add-on description shall be reloaded before attempting to map media times or locate media components. Reloading typically happens for manifest-based add-on such as MPEG-DASH or MPEG-MMT.

`timeline_id`: indicates the active timeline, as identified in a LocationDescriptor.

Additional fixes:
- the short timecode form is only 24 bytes, this should be fixed in the TimelineDescriptor.
- the name of the NTP timestamps is not aligned between table and semantics, fix to ntp_timestamp.

3.2 Inband carriage of timecodes

Replace Table T-1 by

Syntax	Nb bits	Mnemonic

TEMI_AU {
CRC_flag
timing_in_PES_private_data_flag
reserved
for (i=0; i<N; i++) {
temi_descriptor();
}
if (CRC_flag) {
CRC_32
}
}
	
	1
	1
	6
	
	32
		
		uimsbf

Add to semantics:

"timing_in_PES_private_data_flag : if set to 1, indicates that the associated timing information is carried in the PES private data of the audio or video streams of the associated program. The syntax for the PES_private_data carrying TEMI timeline is defined in table T-7."

In T.3.3 Timeline Descriptor, add after first sentence:

"This descriptor shall not be present in a TEMI Access Unit if the timing_in_PES_private_data_flag is set to 1."

Add new clause
"
T3.3 Carriage in PES_private_data

"When the TEMI Access Unit if the timing_in_PES_private_data_flag is set in TEMI access units, the timeline information is embedded in the PES_private_data field of the PES header in audio or video streams of the associated program. In this case, the 128 bits of private data have the following semantics

Syntax	Nb bits	Mnemonic
temi_pes_private_data {		
temi_sync_word	16	uimsbf
timeline_id	8	
force_reload	1	
sce_prevention	1	
timing_type	2	
reserved=0b111	3	
if (timing_type==0) {		
timescale	32	uimsbf
media_timestamp	64	uimsbf
}		
else if (timing_type==3) {		
reserved=0xFFFFFFFF	32	uimsbf
ntp_timestamp	64	uimsbf
}		
else if ((timing_type==1) || (timing_type==2)) {		
drop	1	
frames_per_tc_seconds	15	uimsbf
duration	16	uimsbf
if (timing_type==1) {		
short_time_code	24	uimsbf
reserved=0xFFFFFFFFFF	40	uimsbf
} else {		
long_time_code	64	uimsbf
}		
}		
}		

Semantics

`temi_sync_word` is a 2-byte sync word that shall be set to 0x5449 (hexa for "TI": Timecode Information).
`timing_type` indicates the nature of the timestamp included in the payload: media ticks, NTP clock or timecode as specified in RFC5484.

`sce_prevention` indicates that the timing value T (`media_timestamp, ntp_timestamp, short_time_code` or `long_time_code`) has been inverted in order to prevent emulation of the `packet_start_code_prefix (0x000001)`. The original timestamp can be recovered by inverting each bit in the given word. For example, if the short time code value for the frame is 0x000001 (first frame), the resulting coded field is `short_time_code=0xFFFFFE`

`timeline_id, force_reload, timescale, media_timestamp, ntp_timestamp, drop, frames_per_tc_seconds, duration, short_time_code` and `long_time_code` have the same semantics as in T-3.2
"

To help other bodies understand the impact and benefits of using TEMI in their systems, we suggest adding in the overview a summary on:

- Purpose of TEMI
- Timecode carriage possibilities: PES only , PES + private_data , private_data only if no service signaling is needed

- signaling expressivity: add-ons location compactness, splicing signaling, reloading of manifests, announcement of resources
- bitrate considerations

3.3 Carriage of PTP stamps

The IEEE 1588 Precision Time Protocol (PTP) defines a higher precision timing protocol than NTP, and codes its timestamps on 48+32 bits. These timestamps can fit in TEMI Timeline Descriptor or in temi_pes_private_data, and there are enough extension code points to use them. It could be useful to ask NBs about the usage of PTP for the purpose of TEMI, by adding an editor's note to the WD.

"EDITOR'S NOTE:
PTP (IEEE 1588) timestamps could be added to TEMI Timeline Descriptor or in temi_pes_private_data. WG11 welcomes feedback on this matter for the next meeting to clarify the need for PTP."

4 Conclusion

We suggest inclusion of the proposed tool for in-band carriage of timecode in the current text and issue PDAM6 of ISO/IEC 13818-1:2013 at the 106th MPEG meeting.