FINAL DRAFT

AMENDMENT

ISO/IEC 23008-1:2014 FDAM 2

ISO/IEC JTC 1/SC 29

Secretariat: JISC

Voting begins on: **2015-06-24**

Voting terminates on: **2015-08-24**

Information technology — High efficiency coding and media delivery in heterogeneous environments —

Part 1: MPEG media transport (MMT)

AMENDMENT 2: Header Compression and Cross Layer Interface

Technologies de l'information — Codage à haute efficacité et livraison des medias dans des environnements hétérogènes —

Partie 1: Transport des médias MPEG

AMENDEMENT 2: .

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

Reference number
ISO/IEC 23008-1:2014/FDAM 2:2015(E)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 2 to ISO/IEC 23008-1:2014 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

— *Part 1: MPEG media transport (MMT)*

— *Part 2: High efficiency video coding (HEVC)*

— *Part 3: 3D Audio*

— *Part 10: FEC Codes*

— *Part 11 : Composition Information (CI)*

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 1: MPEG media transport (MMT)), AMENDMENT 2: Header Compression and Cross Layer Interface

The following instructions apply to the (re-organized) first edition of 23008-1

Add the following definitions to clause 3.1, suitable numbered

3.1.37

network abstraction for media
parameter that is used for an interface between media application layer and underlying network layer

Replace the following definition in 3.1.12,

3.1.12

FEC source packet
MMTP packet along with source FEC payload identifier

to

3.1.12

FEC source packet
MMTP packet protected by an FEC encoding

Add the following abbreviated terms to clause 3.2, suitable numbered

CLI cross layer interface

NAM network abstraction for media

Replace the following Table 2 in 8.2.3,

Table 2 - FEC_types

Value	Description
0	MMTP packet without AL-FEC protection

1	MMTP packet with AL-FEC protection (FEC source packet)
2	MMTP packet for repair symbol(s) (FEC repair packet)
3	Reserved for future use

to

Table AMD 2. 1 - FEC_types

Value	Description
0	MMTP packet without Source FEC Payload ID
1	MMTP packet with Source FEC Payload ID
2	MMTP packet for repair symbol(s) for FEC Payload Mode 0 (FEC repair packet)
3	MMTP packet for repair symbol(s) for FEC Payload Mode 1 (FEC repair packet)

NOTE: If FEC type is set to 0, it indicates that FEC is not applied to this MMT packet or that FEC is applied to this MMT packet without adding source FEC payload ID.

Add the following sentence in the last part of packet_sequence_number of semantic in 8.2.3

`packet_sequence_number (32 bits)` – an integer value that is used to distinguish packets that have the same `packet_id`. The value of this field starts from arbitrary value and will be incremented by one for each MMTP packet received. It wraps around to '0' after the maximum value is reached. In FEC repair packet for FEC Payload ID Mode 1, this field shall be replaced with RS_ID.

Add the following sentence in the last sentence of Source_FEC_payload_ID semantics in 8.2.3

`Source_FEC_payload_ID (32 bits)` – This field shall be used only when the value of FEC type is set to '1' (see Annex.エラー! 参照元が見つかりません。). MMTP packet with `FEC type = 1` shall be used for AL-FEC protection for FEC Payload ID Mode 0 and this field shall be added to the MMTP packet after AL-FEC protection.

Add the following sub-clause as an 8.4.4

8.4.4 Header compression for MMTP packet

8.4.4.1 Introduction

Header compression provides the method to reduce the size of the header, technique such as Robust Header Compression (RoHC defined in RFC 3095) may be used. While such technique can severely reduce the size of headers, it has two major drawbacks:

— It relies on complex computations/coding techniques (described in protocol stacks profiles) that are quite heavy on the receiver's side
— It is not a transparent technique and headers need to be entirely decoded, even when it is only to do some filtering and most of the decoded packets are rejected.

It provides two types of headers as follows

- Full size headers are sent regularly and may be used as a reference for reduced-size headers.
- Reduced size headers contain differential information with regards to the last full size header received that is marked as a reference header.

Therefore, by sending only differential information instead of full information, bits savings can be achieved. Additionally, a link to the reference header packet is added in all compressed packets to make sure that the last full header (reference) packets received is indeed the one that shall be used as a reference. Such robustness mechanism is needed as reference packets may actually be lost.

Header compression method applies on the MMTP packet header. For this, a bit (B) is introduced at the beginning of the original header (or at least in the initial fixed part of the header that is common to full size header and reduced size header) to simply inform whether or not the present header is full size or reduced size.

Then many fields present in the full-size headers are either removed or replaced by much smaller fields that contain enough information for the receiver to reconstruct the original full-size header field.

In order to let the receiver know that a full size header will be used as a reference by further reduced size header, an extra bit (I) is also added at the beginning of the full size header in order to mark headers that will be used as a reference later.

Since packet losses may also happen in the network, it is important that even when reduced size headers are used, it is still possible to detect and identify packet losses. Thus, a smaller sequence number is introduced for MMTP packets and mandates that the full-size header is used in place of the reduced size header whenever the reduced sequence number is about to wrap around its initial value.

Similarly, although it is costly in number of bits, the timestamp information associated to packets must be preserved. For this, only the last 19 bits of the full size 32 bits NTP timestamp are used. This allows keeping the same timestamp precision with a wrap around duration of 8 seconds. Consequently, a full size header must be present at least every 8 seconds.

8.4.4.2 Syntax

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=1|B|I|C|FEC|P|X|R|Q|F|R|E|typ|          packet_id            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     packet_sequence_number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         packet_counter                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TB  |  DS  |  TP  |  flow_label |      private_user_data      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       header_extension                    ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         payload_data                      ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     source_FEC_payload_ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure AMD2. 1 Structure of MMTP packet with full header (B=0)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



```
|B|C|FEC|P|X|Q|F|R|E|typ|   Ref    |         delta_timestamp
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|reduced_pckt_id|reduced_SeqNum |reduced_PckCnt |  ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  TB |  DS |  TP |  flow_label |       private_user_data         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       header_extension                    ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         payload_data                      ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     source_FEC_payload_ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure AMD2. 2 MMTP with reduced header (B=1)

8.4.4.3 Semantics

The full size MMTP packet header introduces new fields with their own semantic:

`Compression_flag (B: 1bit)` — This field is added at the beginning of the header in order to indicate whether or not header Compression is used. When set to 0, the full size header is used; when set to 1, the reduced size header is used.

`Indicator_flag (I: 1bit)` — This field is added to tell the receiver whether or not the current full header will later be used as a reference. This field shall be set to 1 when the full header will be used as a reference. This allows receivers discovering that this header information shall be stored as it will be later used as a reference by packets with reduced headers.

The reduced size MMTP packet header introduces new fields with their own semantic:

— The `delta_timestamp` field contains the difference between the `timestamp` field of the reference full size header and the value that would be in current packet `timestamp` field if full size header was used. This difference is coded in a way similar to the 19 least significant bits of an NTP timestamp. If the difference between these two timestamps is larger than 8 seconds (and therefore goes beyond the maximum duration that can be coded on 19 bits) then a packet with full header shall be sent in order to provide a new timestamp reference value for further packets with reduced size header.

— The `reduced_pckt_id` field contains the 8 least significant bits of the `packet_id` field that would be in the header if full size header was used. Therefore, this reduction from 16 bits long packet ids to 8 bits long ids restricts the use of the header compression for streams whose packet_id is between 0 and 255. In other words, header reduction mechanism can only be used on assets with a packet_id between 0 and 255.

— The `reduced_SeqNum` field contains the 8 least significant bits of the `packet_sequence_number` field that would be in the header if full size header was used. Since this new field is coded on 8 bits, MMT receiving entity shall take into account the number of times this field wrapped around 0 to compute the original `packet_sequence_number` value.

— The `reduced_PckCnt` field contains the 8 least significant bits of the `packet_counter` field that would be in the header if full size header was used. Since this new field is coded on 8 bits, MMT receiving entity shall take into account the number of times this field wrapped around 0 to compute the original `packet_counter` value.

— The `RefSeqNum(Ref)` field contains the 5 bits preceded the last 8bits of the packet sequence number (the value set to "reduced sequence number" filed) of the MMTP packet whose full header is used as a reference. This brings additional robustness by allowing the MMT receiving entity to check if the last full header received is actually the one that shall be used as a reference for the current reduced size header. Since MMTP packet may be dropped, RefSeqNum field allows making sure that MMT receiving entity will not try to improperly decode the reduced header when full header for reference has not been received.

The reduced size MMTP packet header also suppresses fields that are present in full size header:

— The `version` field is suppressed as reduced size headers shall have the same version as their referenced header.

— The `I` field is suppressed as only full size headers shall be used as a reference.
— The `RAP_flag` is removed as full size headers shall be sent whenenver the payload contains a Random Access Point

8.4.4.4 MMTP packet header compression rules and normative aspects

A packet with full MMTP header shall at least be sent when one of the following conditions is met:

1) The difference between the timestamps of the current packet and the reference packet is larger than 8 seconds (and therefore cannot be coded on the 19bits long delta_timestamp field),
2) Packet_id is not in the range 0-255,
3) The packet contains a Random Access Point (RAP)

Packet header compression is optional on MMTP sending entities and MMTP receiving entities. Consequently, MMT receivers shall ignore packets with B field set to 1 if they do not support MMTP header compression.

MMTP receiving entities shall not try to decode reduced size header for which the full reference header has not been received, whether because the receiver has just joined the stream or the packet with full reference header has been lost. MMTP receiving entities shall always wait for packets with a reference header (I field set to 1) before they can start or re-start (in case of packet loss of reference header) the header decoding.

Replace the following figure with figure 17. in 9.4

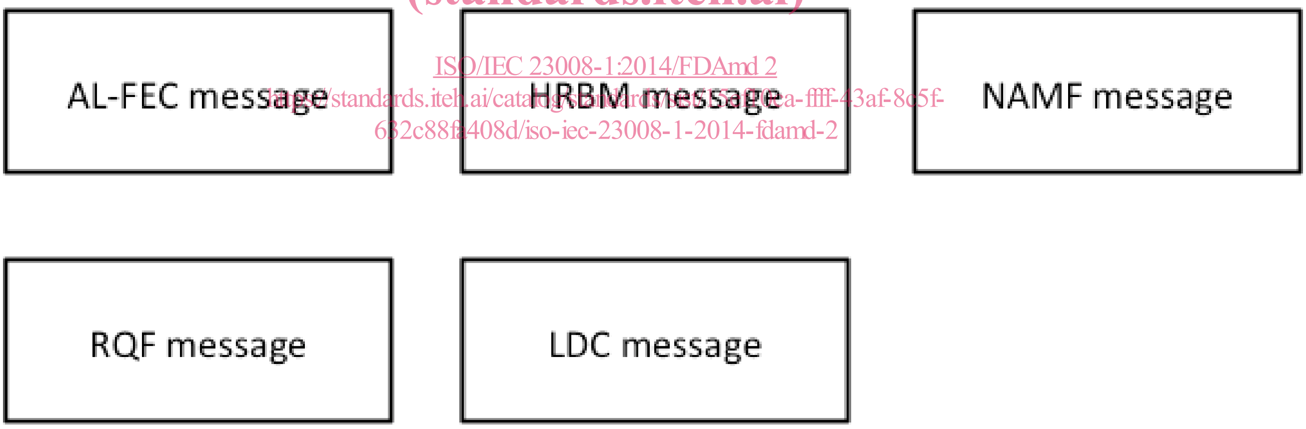

Figure AMD2.3 — Structure of the Signalling Messages for MMT Delivery

Add the following sub-clause as an 9.4.7

9.4.7 NAM Feedback (NAMF message)

9.4.7.1 Syntax

The syntax for the NAM feedback is shown in Table AMD2. 2.

Table AMD2. 2 — NAM_Feedback Message Syntax

Syntax	Values	No. of bits	Mnemonic
NAMF_message ( ) {			
message_id		16	unsigned short
version		8	unsigned char
length		16	unsigned short
NAM_flag		1	unsigned integer
reserved		7	unsigned integer
if(NAM_flag == 0)			
{		8	unsigned integer
message_payload{		8	float
CLI_id		8	float
relative_available_bitrate		8	float
relative_buffer_fullness		16	unsigned integer
relative_peak_bitrate		32	float
average_bitrate_period		32	float
current_delay		32	float
generation_time			
BER			
}			
}			
else if(NAM_flag == 1)			
{			
message_payload{			
CLI_id		8	unsigned integer
available_bitrate		32	float
buffer_fullness		32	float
peak_bitrate		32	float
current_delay		32	float
average_bitrate_period		16	unsigned interger
SDU_size		32	unsigned integer
SDU _loss_ratio		8	unsigned integer
generation_time		32	float
BER		32	float

} } }			

9.4.7.2 Semantics

`message_id` – It indicates NAMF message ID. The length of this field is 16 bits.

`version` – It indicates the version of NAMF messages. MMT receiving entity may check whether the received message is new or not. The length of this field is 8 bits.

`length` – It indicates the length of NAMF messages. The length of this field is 16 bits. It indicates the length of the NAM Feedback message counted in bytes starting from the next field to the last byte of the NAM Feedback message. The value ‘0’ shall not be used.

`NAM_flag` – It indicates whether NAMF message contains absolute NAM information or relative NAM information. The value ‘1’ should be set, if NAMF message contains absolute NAM information.

`CLI_id` - The `CLI_id` is an arbitrary integer number to identify this NAM among the underlying network.

`relative_available_bitrate` - The available bitrate change ratio(%) between the current NAM information and the previous NAM information.

`relative_buffer_fullness` - The remaining buffer fullness change ratio(%) between the current NAM information and the previous NAM information.

`relative_peak_bitrate` - The peak bitrate change ratio(%) between the current NAM information and the previous NAM information.

`available_bitrate` - the `available_bitrate` is bitrate that the scheduler of the underlying network can guarantee to the MMT stream. The `available_bitrate` is expressed in kilobits per second. Overhead for the protocols of the underlying network is not included.

`buffer_fullness` - the buffer is used to absorb excess bitrate higher than the `available_bitrate`. The `buffer_fullness` is expressed in bytes.

`peak_bitrate` - the `peak_bitrate` is maximum allowable bitrate that the underlying network can assign to the MMT stream. The `peak_bitrate` is expressed in kilobits per second. Overhead for the protocols of the underlying network is not included.

`current_delay` - the current_delay parameter indicates the last hop transport delay. The `current_delay` expressed in milliseconds.

`average_bitrate_period` – It provides the period of time over which the average bitrate of the input of MMT procotol session that carries the MMTP packet shall be calculated. The `average_bitrate_period` is provided in units of milliseconds.

`SDU_size` - SDU (Service Data Unit) is data unit in which the underlying network delivers the MMT data. The `SDU_size` specifies the length of the SDU and is expressed in bits. Overhead for the protocols of the underlying network is not included.

`SDU_loss_rate` - The `SDU_loss_ratio` is fraction of SDUs lost or detected as errorneous. Loss ratio of MMT packets can be calculated as a function of `SDU_loss_ratio` and `SDU_size`. The `SDU_loss_ratio` is expressed in percentile.

`generation_time` - The time when the parameters are generated. The `generation_time` is expressed in milliseconds.

`BER` - Bit Error Rate obtained from PHY or MAC layer. For `BER` from PHY layer, this value is presented as a positive value. For `BER` from MAC layer, this value is presented as a negative value which can be used as an absolute value.

Add the following sub-clause as an 9.4.8

9.4.8 Low Delay Consumption (LDC) message

9.4.8.1 Introduction

The LDC Message provides information required to decode and present media data by the MMT receiving entity before it receives metadata such as movie fragment headers. This message indicates that the duration of each sample is fixed as signaled by `default_sample_duration` in Track Extends Box. and the coding dependency structure is fixed across an Asset. When this message is used, the value of decoding time of the first sample of MPU is smaller than the presentation time of the first sample of the MPU by the sum of `base_presentation_time_offset` and the largest value of `sample_composition_time_offset_value` paired `sample_composition_time_offset_sign` is `‘1.’`

9.4.8.2 Syntax

The syntax for Low Delay Consumption Message is shown in Table AMD2. 3.

Table AMD 2. 3 — Low Delay Consumption Message Syntax

Syntax	Values	No. of bits	Mnemonic
LDC_message ( ){			
message_id		16	
Version		8	
Length		16	
base_presentation_time_offset		**31**	
coding_dependency_structure_flag		**1**	
if (coding_dependency_structure_flag == 1){			
period_of_intra_coded_sample	**N1**	**8**	
for (i=0 ; i<N1;i++){		**8**	
sample_composition_time_offset_sign		**1**	
sample_composition_time_offset_value		**31**	
}			
}			
}			

9.4.8.3 Semantics

`message_id` - indicates the identifier of the LDC Message.

`version` - version of the LDC messages. An MMT receiving entity can use this field to check the version of the received LDC message.

`length` - length of the LDC messages in bytes, counting from the first byte of the next field to the last byte of the LDC message. The value ‘0’ is not valid for this field.

`base_presentation_time_offset` - provides information about the time difference between decoding time and presentation time in microseconds. Presentation time of each sample shall be