INTERNATIONAL STANDARD

ISO/IEC 23008-1

First edition
2014-06-01

AMENDMENT 1
2015-04-01

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

Part 1: MPEG media transport (MMT)

AMENDMENT 1: Additional technologies for MPEG Media Transport (MMT)

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

Partie 1: Transport des médias MPEG

AMENDEMENT 1: Technologies supplémentaires pour le transport des médias MPEG

Reference number
ISO/IEC 23008-1:2014/Amd.1:2015(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23008 consists of the following parts, under the general title *Information technology — High efficiency coding and media delivery in heterogeneous environments:*

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

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

— *Part 3: 3D Audio*

— *Part 5: HEVC Conformance testing and reference software*

— *Part 8: Conformance Specification for HEVC*

— *Part 10: MPEG Media Transport Forward Error Correction (FEC) codes*

— *Part 11: MPEG Media Transport Composition Information*

— *Part 12: Image file format*

— *Part 13: MMT Implementation Guidelines*

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

Part 1:
MPEG media transport (MMT)

AMENDMENT 1: Additional technologies for MPEG Media Transport (MMT)

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

Add the following definitions to 3.1, suitably numbered.

3.1.33
asset delivery characteristics

description about required Quality of Service (QoS) for delivery of Assets. ADC is represented by the parameters agnostic to specific delivery environment.

Add the following abbreviated terms to 3.2, suitably numbered

ADC asset delivery characteristics

ARQ automatic repeat request

In 5.2, amend the 1st paragraph to read as follows.

As shown in Figure 4, a Package is a logical entity. A Package shall contain one or more presentation information documents such as one specified in ISO 23008-11, one or more Assets and for each Asset an associated Asset Delivery Characteristics (ADC). In other words, as processing of a Package is applied per MPU basis and an Asset is a collection of one or more MPUs that share the same Asset ID. It can be also considered that one Package is composed of one Presentation Information, one or more MPUs and associated ADC for each Asset.

In 5.2.1, replace Figure 4 with the following figure.

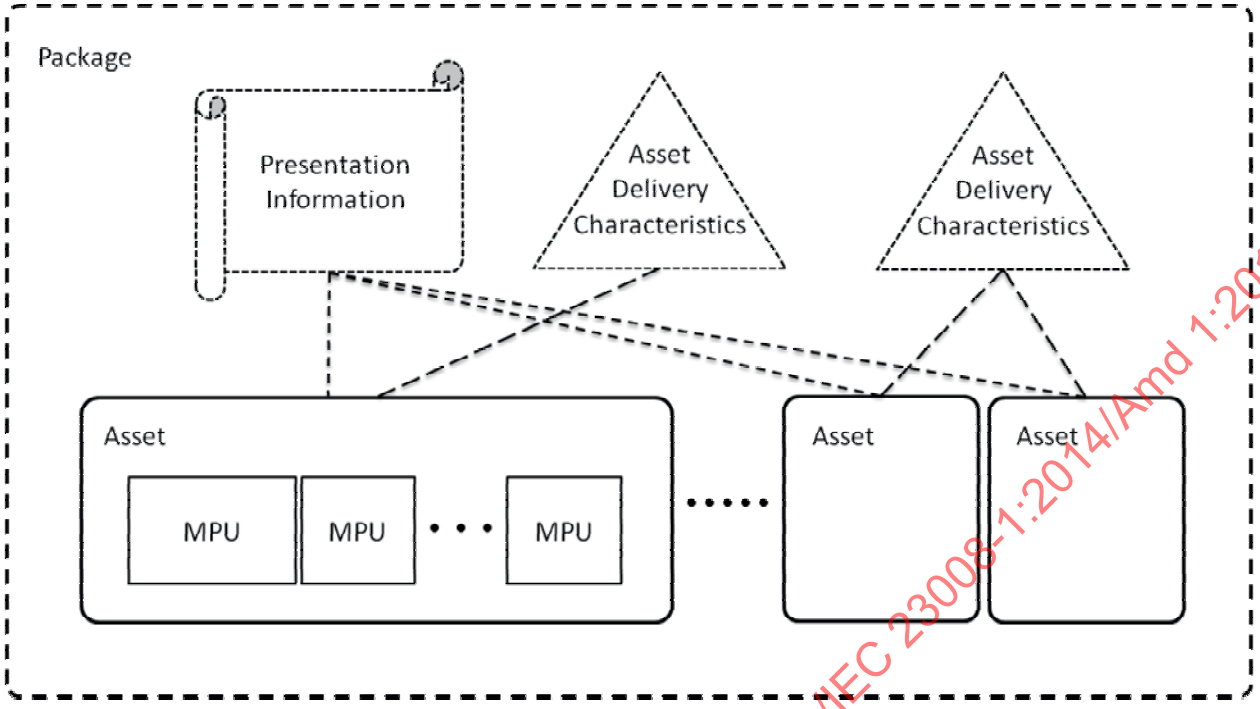

Figure — AMD1.1 — Overview of package

In 5.2, insert the following paragraph after the one beginning in "Presentation Information (PI) document specify … ".

Asset Delivery Characteristics (ADC) shall provide the required QoS information for transmission of Assets. Multiple Assets can be associated with a single ADC. However a single Asset shall not be associated with multiple ADCs. This information can be used by the entity packetizing the Package to configure the fields of the MMTP payload header and MMTP packet header for efficient delivery of the Assets.

Add the following sub-clause as 5.5.

5.5 Asset delivery characteristics

5.5.1 Introduction

The Asset Delivery Characteristics (ADC) describes the QoS requirements and statistics of Assets for delivery. Each Asset in a Package shall be associated with an ADC. The ADC for each Asset is used by an MMT sending entity to derive the appropriate QoS parameters and the transmission parameters to which a resource reservation and a delivery policy may apply. The ADC is represented in a protocol agnostic format to be generally used by QoS control service entity defined by other standard development organizations, such as IETF, 3GPP, IEEE, etc. It consists of a `QoS_descriptor` element and a `bitstream_descriptor` element. ADC is an XML file that conforms to the schema in section 5.5.3. The MIME type of ADC is defined as Annex H.

5.5.2 ADC Descriptors

5.5.2.1 QoS descriptor

The `QoS_descriptor` element defines required QoS levels on delay and loss for Asset delivery. It consists of `loss_tolerance` attribute, `jitter_sensitivity` attribute, `class_of_service` attribute and `bidirection_indicator` attribute.

5.5.2.2 Bitstream descriptor

The `bitstream_descriptor` element provides the statistics of Asset. It provides the parameters to implement token bucket traffic shaping such as sustainable rate and buffer size. In addition, peak rate and maximum MFU size represent burstiness of Asset as shown in Figure AMD1. 2., where burstiness is defined as a ratio between a `peak_rate` and `sustainable_rate`.

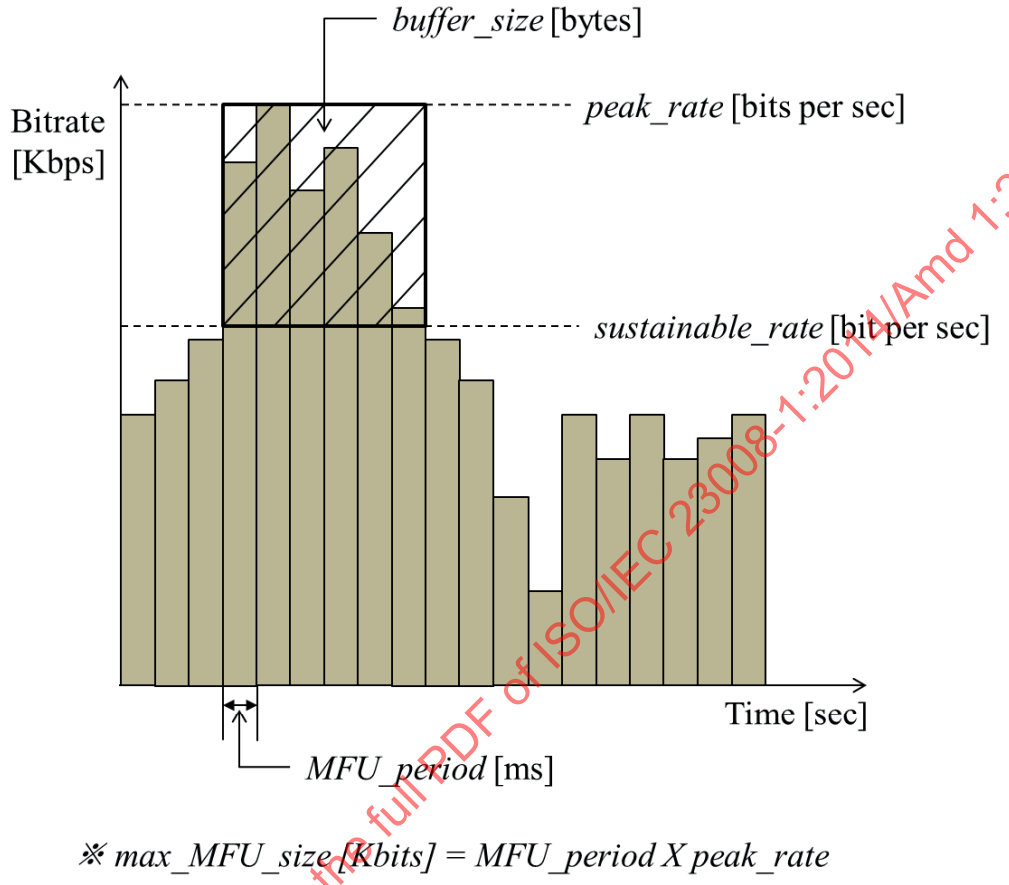

Figure — AMD1.2 — The bitstream_descriptor depicted for a variable bit-rate of Asset

5.5.3 Syntax

```
<complexType name="AssetDeliveryCharacteristic">
     <sequence>
          <element name="QoS_descriptor" type="mmt:QoS_descriptorType" />
          <element name="Bitstream_descriptor" type="mmt:Bitstream_descriptorType"/>
     </sequence>
</complexType>

<complexType name="QoS_descriptorType">
     <attribute name=" loss_tolerance " type="integer"/>
     <attribute name="jitter_sensitivity" type="integer"/>
     <attribute name="class_of_service" type="boolean"/>
     <attribute name="bidirection_indicator" type="boolean"/>
</complexType>

<complexType name="Bitstream_descriptorType">
     <choice>
          <complexType name="Bitstream_descriptorVBRType">
               <attribute name="sustainable_rate" type="float"/>
               <attribute name="buffer_size" type="float"/>
               <attribute name="peak_rate" type="float"/>
               <attribute name="max_MFU_size" type="integer"/>
               <attribute name="mfu_period" type="integer"/>
          </complexType>
```

```
            <complexType name="Bitstream_descriptorCBRType">
                <attribute name="peak_rate" type="float"/>
                <attribute name="max_MFU_size" type="integer"/>
                <attribute name="mfu_period" type="integer"/>
            </complexType>
        </choice>
</complexType>
```

5.5.4 Semantics

`loss_tolerance` — indicates required loss tolerance of the Asset for the delivery. The value of `loss_tolerance` attribute is listed in Table AMD1.1.

Table — AMD1.1 — Value of `loss_tolerance` attribute

Value	Description
0	This Asset requires lossless delivery
1	This Asset allows lossy delivery

`jitter_sensitivity` — indicates required jitter level of underlying delivery network for the Asset delivery between end-to-end. The value of `jitter_sensitivity` attribute is listed in Table AMD1.2.

Table — AMD1.2 — Value of `jitter_sensitivity` attribute

Value	Description
0	This Asset requires the preserve time variation between MMTP packets
1	This Asset doesn't require the preserve time variation between MMTP packets.

`class_of_service` — classifies the services in different classes and manage each type of bitstream with a particular way. For example, Media Aware Network Element (MANE) can manage each type of bitstream with a particular way. This field indicates the type of bitstream attribute as listed in Table AMD1.3.

Table — AMD1.3 — Value of `class_of_service` attribute

Value	Description
0	The Constant Bit Rate (CBR) service class shall guarantee peak bitrate at any time to be dedicated for transmission of the Asset. This class is appropriate for realtime services which require fixed bitrate such as VoIP without silence suppression.
1	The Variable Bit Rate (VBR) service class shall guarantee sustainable birate and allow peak bitrate for the Asset with delay constraints over shared channel. This class is appropriate for most realtime services such as video telephony, video conferencing, streaming service, etc.

`Bidirection_indicator` — If set to '1', bidirectional delivery is required. If set to '0', bidirectional delivery is not required.

`Bitstream_descriptorVBRType` — when `class_of_service` is '1', "`Bitstream_descriptorVBRType`" shall be used for "`Bitstream_descriptorType`".

`Bitstream_descriptorCBRType` — when `class_of_service` is '0', "`Bitstream_descriptorCBRType`" shall be used for "`Bitstream_descriptorType`".

`sustainable_rate` — defines the minimum bitrate that shall be guaranteed for continuous delivery of the Asset. The `sustainable_rate` corresponds to drain rate in token bucket model. The `sustainable_rate` is expressed in bits per second.

buffer_size — defines the maximum buffer size for delivery of the Asset. The buffer absorbs excess instantaneous bitrate higher than the sustainable_rate and the buffer_size shall be large enough to avoid overflow. The buffer_size corresponds to bucket depth in token bucket model. Buffer_size of a CBR (constant bit rate) Asset shall be zero. The buffer_size is expressed in bytes.

peak_rate — defines peak bitrate during continuous delivery of the Asset. The peak_rate is the highest bitrate during every MFU_period. The peak_rate is expressed in bits per second.

MFU_period — defines period of MFUs during continuous delivery of the Asset. The MFU_period is measured as the time interval of sending time between the first byte of two consecutive MFUs. The MFU_period is expressed in millisecond.

max_MFU_size — indicates the maximum size of MFU, which is MFU_period*peak_rate. The max_MFU_size is expressed in byte.

Add the following subclause as 5.6.

5.6 Bundle delivery characteristics

5.6.1 Introduction

The Bundle Delivery Characteristics (BDC) describes the QoS requirements and statistics of Bundle for delivery. Each Bundle in a Package shall be associated with an BDC. The BDC for each Bundle is used by an MMT sending entity to derive the appropriate QoS parameters and the transmission parameters to which a resource reservation and a delivery policy may apply. The BDC is represented in a protocol agnostic format to be generally used by QoS control service entity defined by other standard development organizations, such as IETF, 3GPP, IEEE, etc. It consists of a QoS_descriptor element and a bitstream_descriptor element as defined in ADC.

5.6.2 BDC descriptors

5.6.2.1 QoS descriptor

The QoS_descriptor element defines required QoS levels on delay and loss for Bundle delivery. It consists of loss_tolerance attribute, jitter_sensitivity attribute, class_of_serivce attribute and bidirection_indicator attribute.

5.6.2.2 Bitstream descriptor

The bitstream_descriptor element provides the statistics of Bundle. It provides the parameters to implement token bucket traffic shaping such as sustainable rate and buffer size. In addition, peak rate and maximum MFU size represent burstiness of Bundle where burstiness is defined as a ratio between a peak_rate and sustainable_rate.

5.6.3 Syntax

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"targetNamespace="mmt">
    <xs:element name="BundleDeliveryCharacteristic"
type="mmt:BundleDeliveryCharacteristicType">
        <xs:attribute name="MMT_package_id" type="xs:string"/>
    </xs:element>

    <xs:complexType name="mmt:BundleDeliveryCharacteristicType">
        <xs:sequence>
            <xs:element name="Bundle" type="mmt:BundleType"
minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="mmt:BundleType">
        <xs:sequence>
            <xs:element name="Element_Asset_id" type="asset_id_T" minOccurs="1">
                <xs:attribute name="Intra_Bundle_Priority" type="xs:integer"/>
```

```
                </xs:element>
            </xs:sequence>
            <xs:element name="Bundle_QoS_descriptor" type="mmt:QoS_descriptorType"/>
            <xs:element name="Bundle_Bitstream_descriptor" type="mmt:Bitstream_
descriptorType"/>
            <xs:attribute name="Bundle_id" type="xs:integer"/>
            <xs:attribute name="Inter_Bundle_Priority" type="xs:integer"/>
        </xs:complexType>
        <xs:complexType name="mmt:QoS_descriptorType">
            <xs:attribute name="loss_tolerance" type="xs:integer"/>
            <xs:attribute name="jitter_sensitivity" type="xs:integer"/>
            <xs:attribute name="class_of_service" type="xs:boolean"/>
            <xs:attribute name="distortion_levels" type="xs:integer"/>
            <xs:attribute name="bidirection_indicator" type="xs:boolean"/>
        </xs:complexType>

        <xs:complexType name="Bitstream_descriptorType">
            <xs:choice>
                <xs:complexType name="Bitstream_descriptorVBRType">
                        <xs:attribute name="sustainable_rate" type="xs:float"/>
                        <xs:attribute name="buffer_size" type="xs:float"/>
                        <xs:attribute name="peak_rate" type="xs:float"/>
                        <xs:attribute name="max_MFU_size" type="xs:integer"/>
                        <xs:attribute name="mfu_period" type="xs:integer"/>
                </xs:complexType>
                <xs:complexType name="Bitstream_descriptorCBRType">
                        <xs:attribute name="peak_rate" type="xs:float"/>
                        <xs:attribute name="max_MFU_size" type="xs:integer"/>
                        <xs:attribute name="mfu_period" type="xs:integer"/>
                </xs:complexType>
            </xs:choice>
        </xs:complexType>
</xs:schema>
</xml>
```

5.6.4 Semantics

`loss_tolerance` — indicates required loss tolerance of the Bundle for the delivery. The value of loss_tolerance attribute is listed in Table AMD1.1.

Table — AMD1.4 — Value of `loss_tolerance` attribute

Value	Description
0	This Bundle requires lossless delivery
1	This Bundle allows lossy delivery

`jitter_sensitivity` — indicates required jitter level of underlying delivery network for the Bundle delivery between end-to-end. The value of jitter_sensitivity attribute is listed in Table AMD1.2.

Table — AMD1.5 — Value of `jitter_sensitivity` attribute

Value	Description
0	This Bundle requires the preserve time variation between MMT protocol packets
1	This Bundle doesn't require the preserve time variation between MMT protocol packets.

`class_of_service` — classifies the services in different classes and manage each type of bitstream with a particular way. For example, MANE can manage each type of bitstream with a particular way. This field indicates the type of bitstream attribute as listed in Table AMD1.3.

Table — AMD1.6 — Value of `class_of_service` attribute

Value	Description
0	The Constant Bit Rate (CBR) service class shall guarantee peak bitrate at any time to be dedicated for transmission of the Bundle. This class is appropriate for realtime services which require fixed bitrate such as VoIP without silence suppression.
1	The Variable Bit Rate (VBR) service class shall guarantee sustainable birate and allow peak bitrate for the Bundle with delay constraints over shared channel. This class is appropriate for most realtime services such as video telephony, videoconferencing, streaming service, etc.

`Bidirection_indicator` — If set to '1', bidirectional delivery is required. If set to '0', bidirectional delivery is not required.

`Bitstream_descriptorVBRType` — when `class_of_service` is '1', "`Bitstream_descriptorVBRType`" shall be used for "`Bitstream_descriptorType`".

`Bitstream_descriptorCBRType` — when `class_of_service` is '0', "`Bitstream_descriptorCBRType`" shall be used for "`Bitstream_descriptorType`".

`sustainable_rate` — defines the minimum bitrate that shall be guaranteed for continuous delivery of the Asset. The `sustainable_rate` corresponds to drain rate in token bucket model. The `sustainable_rate` is expressed in bytes per second.

`buffer_size` — defines the maximum buffer size for delivery of the Bundle. The buffer absorbs excess instantaneous bitrate higher than the `sustainable_rate` and the `buffer_size` shall be large enough to avoid overflow. The `buffer_size` corresponds to bucket depth in token bucket model. `Buffer_size` of a CBR (constant bit rate) Bundle shall be zero. The `buffer_size` is expressed in bytes

`peak_rate` — defines peak bitrate during continuous delivery of the Bundle. The peak_rate is the highest bitrate during every `MFU_period`. The peak_rate is expressed in bytes per second.

`MFU_period` — defines period of MFUs during continuous delivery of the Bundle. The `MFU_period` measured as the time interval of sending time between the first byte of two consecutive MFUs. The `MFU_period` is expressed in millisecond

`max_MFU_size` — indicates the maximum size of MFU, which is `MFU_period`*`peak_rate`. The `max_MFU_size` is expressed in byte.

`MMT_package_id` — this field is a unique identifier of the Package. This BDC describes delivery characteristics of all the possible Bundles within the scope of this package.

`Element_Asset_id` — identifier of asset which is an element of current bundle

`Bundle_id` — identifier of bundle which distinguish bundles within the package.

`Intra_Bundle_Priority` — defines the relative priority level among assets within a bundle, which ranges from 0(highest) to 12(lowest).

`Inter_Bundle_Priority` — defines the relative priority level among bundles, which ranges from 0(highest) to 12(lowest).

In 6.3.1, insert the following sentence in last sentence of the paragraph.

When it is required to store the ADC together with MPU, it shall be stored in the 'meta' box at the file level and its presence shall be indicated through the '`is_adc_present`' flag and the MIME type of the item that stores the ADC.

Replace the following syntax in 6.3.2.

```
aligned(8) class MPUBox
   extends FullBox('mmpu', version, 0){
   unsigned int(1) is_complete;
   unsigned int(1) is_adc_present;
   unsigned int(6) reserved;
   unsigned int(32) mpu_sequence_number;
   AssetIdentifierBox();
}
```

Add the following semantics in 6.3.3.

`is_adc_present` — indicates whether the ADC is present as an XML box in a '`meta`' box. The MIME type of the ADC file as defined Annex H shall be indicated in an item informaiton box '`iinf`'.

Replace the following sentence in the 2nd paragraph in 8.1

The MMT protocol is an application layer transport protocol supporting delivery of Packages over heterogeneous packet-switched delivery networks, including IP-based network environments. The MMT protocol provides enhanced features for delivery of Packages such as protocol level multiplexing, which for example enables various Assets to be delivered over a single MMTP packet flow, and delivery timing model independent of presentation time to adapt to a wide range of network jitters

with the following sentence:

The MMT protocol is an application layer transport protocol supporting delivery of Packages over heterogeneous packet-switched delivery networks, including IP-based network environments. The MMT protocol provides enhanced features for delivery of Packages such as protocol level multiplexing, which for example enables various Assets to be delivered over a single MMTP packet flow, and delivery timing model independent of presentation time to adapt to a wide range of network jitters, and information to support Quality of Service (QoS).

In 8.2.1, replace the following sentence in the first paragraph

It supports several enhanced features, such as media multiplexing and network jitter calculation..

with the following sentence:

It supports several enhanced features, such as media multiplexing, network jitter calculation, and QoS indication.

In 8.2.1, add the following paragraph after the last paragraph.

MMT protocol provides priority related information to enable underlying network layers or the intermediate network entities to map the priority information in MMTP packet header such as type_of_bitrate, delay_sentivity, transmission_priority and flow_label to the network protocol according to predetermined priority mapping policy. When DiffServ [RFC2474] is used, this priority information may be used to set the 6-bit DSCP value of the DS field in the IP header. The underlying network entity supporting Diffserv shall then process the IP packets according to the mapping defined by the priority related information in MMTP packet header.

Insert the following figure after Figure 7.

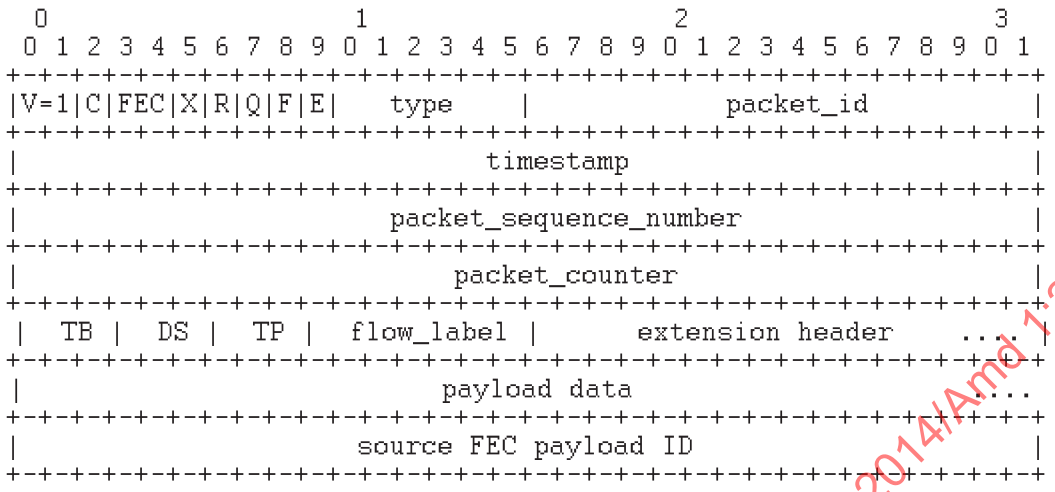

Figure — AMD1.3 — MMTP packet header, payload and footer

Add the following semantics in 8.2.3.

`Version (V:2bit)` — indicates the version number of the MMTP protocol. This field shall be set to '`01`' to comply with this specification.

`QoS_classifier_flag (Q:1bit)` — When set to '1' it indicates that QoS classifier information is used. QoS classifier contains `delay_sensitivity` field, `reliability_flag` field, and `tranmission_priority` field. It indicates the QoS class property. The application can perform per-class QoS operations according to the particular value of one property. The class values are universal to all independent sessions.

`flow_identifier_flag (F:1bit)` — When set to '1', it indicate that flow identifier information is used. flow identifier contains `flow_label` field, and `flow_extension_flag` field. It indicates the flow identifier. The application can perform per-flow QoS operations in which network resources are temporarily reserved during the session. A flow is defined to be a bitsream or a group of bitstreams whose network resources are reserved according to transport characteristics or ADC in Package.

`reliability_flag (r: 1bit)` — When "`reliability_flag`" is set to '0', it shall indicate that the data is loss tolerant (e.g. media data), and that the following "`transmission_priority`" field shall be used to indicate relative priority of loss. When "`reliability_flag`" is set to '1', the "`transmission_priority`" field will be ignored, and shall indicate that the data is not loss tolerant (e.g., signaling data, service data, or program data).

`flow_extension_flag (E: 1bit)` — If there are more than 127 individual flows, this bit is set to '1' and one more byte can be used in `extension_header`.

`type_of_bitrate (TB: 3bits)` — indicates the type of bitrate as listed in Table AMD1.4.

Table — AMD1.4 — Value of `type_of_bitrate`

Value	Description
000	Constant Bit Rate (CBR)
001	Non-Constant Bit Rate (nCBR)
010 ~ 111	reserved

`delay_sensitivity (DS: 3bits)` — indicates the delay sensitivity of the data between end-to-end delivery for the given service as listed in Table AMD1.5. This field is derived from the application as the same content may have different delay requirements in different applications.

Table — AMD1.5 — Value of `delay_sensitivity`

Value	Description
111	conversational service (~100ms)
110	live-streaming service (~1sec)
101	delay-sensitive interactive service (~2sec)
100	interactive service (~5sec)
011	streaming service (~10sec)
010	non-realtime service
001	reserved
000	reserved

`transmission_priority (TP: 3bits)` — provides the `transmission_priority` for the media packet, and it may be mapped to the NRI of NAL, DSCP of IETF, or other loss priority field in another network protocol. This field shall take values from '7' ('1112') to '0' ('0002'), where 7 is the highest priority, and '0' is the lowest priority.

`flow_label (7bits )` — indicates the flow identifier. The application can perform per-flow QoS operations in which network resources are temporarily reserved during the session. A flow is defined to be a bitsream or a group of bitstreams whose network resources are reserved according to transport characteristics or ADC in Package. It is an implicit serial number from '0' to '127'. An arbitrary number is assigned temporarily during a session and refers to every individual flow for whom a decoder (processor) is assigned and network resource could be reserved.

Change the second paragraph of 9.1 to the following:

Six types of signalling messages are defined for the consumption of Packages;

Add the following sentence after the one beginning in "Device Capability Information (DCI) message: ...", suitably numbered:

— Security Software Request (SSWR) message: it is used to request security software for comsuming Package or Asset by an MMT receiving entity. It can also include PA table or MP table (see 9.3.7).

Change the third paragraph of 9.1 to the following:

Six signaling messages are defined that relate to the delivery of the Package:

Add the following sentence after the one beginning in "HRBM: ...", suitably numbered:

— Automatic Repeat-Request(ARQ) Configuration (AC) message: It provides information required for ARQ configuration (see 9.4.3);

— Automatic Repeat-Request(ARQ) Feedback (AF) message: It provides information required for ARQ feedback (see 9.4.4);

— Measurement Configuration (MC) message: This message type is used for providing information to configure measurement of delivery quality (see 9.4.5);

— Reception Quality Feedback (RQF) message: This message type is used for measurement reporting by an MMT receiving entity (see 9.4.6);

— Asset Delivery Characteristic (ADC) message: This message type is used for providing information to configure the network delivery resource (see 9.4.7);

Replace Figure 16 with the following figure:

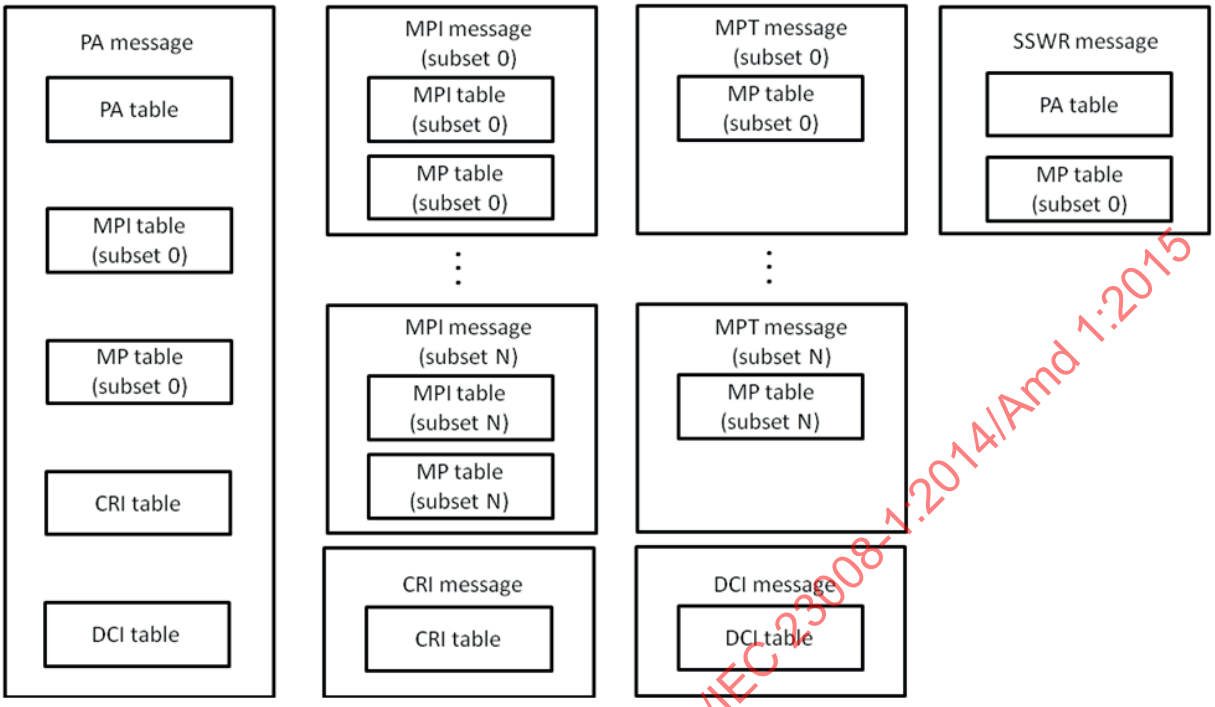

Figure — AMD1.4 — Structure of the signaling messages and tables for Package consumption

Change the last sentence in 9.3.1 to the following:

The relationship between a message and a table is shown in Figure AMD1.4.

Add the following subclause and change the following chapter numbers from 9.3.12.

6.3.12 Security Software Request (SSWR) Message

6.3.12.1 Introduction

The overall operation of downloadable DRM and CAS for MMT is described in Annex D. There are 5 steps in Annex D. Among them, the message for the security software request is sent from a receiving MMT entity to downloadable DRM/CAS server. The message for DRM and CAS SW request is defined in this subclause.

6.3.12.2 Syntax

The syntax of SSWR message is defined in Table AMD1. 6.

Table — AMD1.6 —SSWR Message syntax

Syntax	Values	No. of bits	Mnemonic
SSWR_message () {			
message_id		**16**	**uimsbf**
version		**8**	**uimsbf**
length		**16**	**uimsbf**
message_payload {			
token_ID **{**			**uimsbf**
token_ID_URI_length	**N1**	**8**	**bslbf**
for (i=0; i<N1; i++) {			
token_ID_URI_byte		**8**	
}			

Table — *(continued)*

Syntax	Values	No. of bits	Mnemonic
Number_of_deviceID	**N2**	**8**	**uimsbf**
for (i=0; i<N2; i++) { device_ID {			**uimsbf**
deviceID_length	**N3**	**8**	
for (j=0; j<N3; j++) {			
length_byte		**8**	
}			**uimsbf**
}		**8**	
}			
tokenIssure_ID {			
tokenIssuer_ID_URI_length	**N4**	**8**	
for (i=0; i<N4; i++) {			
tokenIssuer_URI_bytes		**8**	
}			
}		**64**	
tokenIssueTime		**64**	**uimsbf**
tokenExpireTime			
information_table_info {			**uimsbf**
number_of_tables			**uimsbf**
for (i=0; i<N5; i++) {	**N5**	**8**	
MMT_signaling_table_id		**8**	
MMT_signaling_table_version		**8**	
}			**uimsbf**
}			**uimsbf**
}			**uimsbf**
}			
}			

6.3.12.3 Semantics

`message_id` — indicates the type of MMT Signaling messages.

`version` — indicates the version of MMT Signaling messages.

`length` — indicates the length of MMT signaling messages. The value '0' is never used for this field.

`tokenID` — Identification of Token and is provided by Token Provider. Token should be provided by the trustable entity. It has sub elements of Device ID, Token Issuer ID, Issue Time and Expire Time

`deviceID` — provides identification of device(s) under Token. If MMT client want to consume Asset/Package two different devices, then multiple Device ID should be provided.

`tokenIssureID` — Identification of trust entity that issues a token. This filed is to be used by D-DRM/D-CAS server to verify the validity of Token.

`tokenIssueTime` — a time at which the Token is issued. The unit of this field is second. NTC format will be used.

`tokenExpireTime` — a time at which the Token is expired. The unit of this field is second. NTC format will be used.

`MMT_signaling_table_info` — provides the information of MPT related to Package/ Asset to be descripted by downloaded DRM or CAS.

Replace the following paragraph to 9.4:

Signaling messages for delivery are HRBM Message, ARQ Configuration (AC) Message, ARQ Feedback (AF) Message, Measurement Configuration (MC) Message, and Reception Quality Feedback (RQF) Message. They are shown at Figure AMD1. 5.

Replace Figure 17 with the following figure:

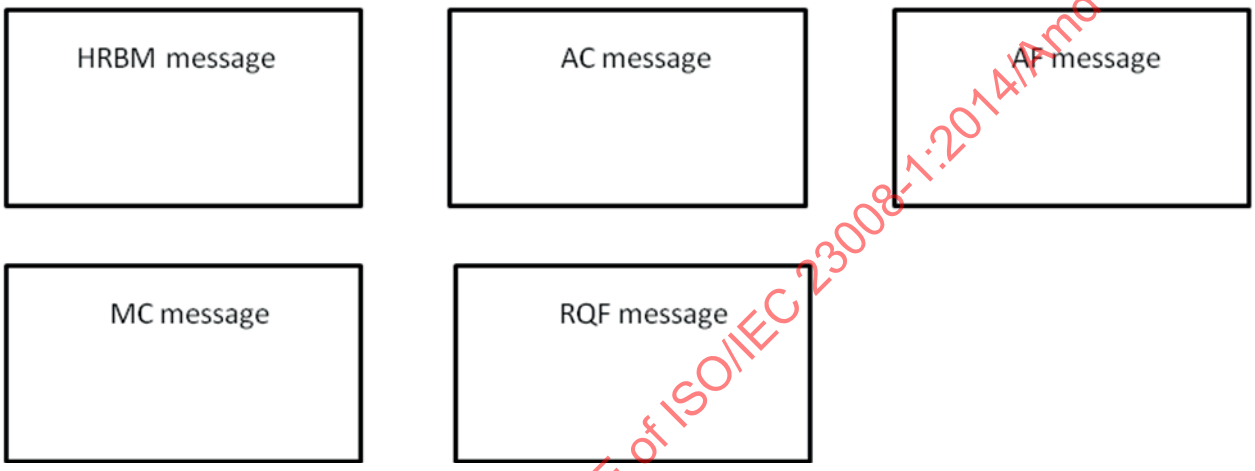

Figure — AMD1.5 —Structure of the signaling messages for delivery

6.4.3 The measurement configuration (mc) message

6.4.3.1 Introduction

MC messages are used for transport metrics measurement. It provides the information on measurement metrics (e.g. a receiving entity buffer status, round trip delay, NAM parameter), measurement condition such as a measurement starting time and a period and measurement report. The syntax of an MC message is shown in Table AMD1.7 and its semantics are described in 9.4.3.2.

6.4.3.2 Syntax

Syntax of the MC message is shown in Table AMD1.7.

Table — AMD1.7 — MC message syntax

Syntax	Values	No. of bits	Mnemonic
Measurement _message() {			
message_id		**16**	**uimsbf**
version		**8**	**uimsbf**
length		**16**	**uimsbf**
message_payload {			
reserved	'1111 11'	**6**	**bslbf**
measurement_mode		**2**	**bslbf**

Table — *(continued)*

Syntax	Values	No. of bits	Mnemonic
if(measurement_mode !=11) { if(measurement_mode ==01){			
measurement_start_time		**32**	**uimsbf**
}else if(measurement_mode ==10) { measurement_start_condition()			
}			
measurement_stop_time		**32**	**uimsbf**
measurement_period		**32**	**uimsbf**
measurement_report{ server_address{			
MMT_general_location_info()		**32**	
}			
report_type		**8**	**bslbf**
}			
}			
}			
}			

6.4.3.3 Semantics

`message_id` — indicates the message ID. The length of this field is 16 bits.

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

`length` — indicates the length of the messages in bytes, counting from the beginning of the next field to the last byte of the MC message. The value '0' shall not be used for this field.

`measurement_mode` — indicates when the MMT receiving entity should start measuring the item indicated by `measurement_item_flag`. Valid values for this filed is described in Table AMD1.9.

Table — AMD1.9 — Value of measurement_mode

Value	Description
00	start measurement immediately and stop measurement at the appointed time
01	start and stop measurement at the appointed time
10	start measurement at the measurement condition
11	stop measurement immediately

`measurement_start_time` — indicates a UTC time in NTP format corresponding to the measurement start time. This field is the 32 bits MSB from the full resolution NTP timestamp.

`measurement_start_condition` — indicates a specific condition which MMT receiving entity starts a measurement. The example of specific condition is receiver buffer status or the reception channel status. The example of condition is given at Table x.

`measurement_stop_time` — indicates a UTC time in NTP format corresponding to the measurement stop time. This field is the 32 bits MSB from the full resolution NTP timestamp. The value '0x0000' means the MMT receiving entity measures periodically with the `measurement_period` until receiving "immediately stop measurement" indication.

`measurement_period` — indicates how frequently the MMT receiving entity should measure the item indicated by `measurement_item_flag`. The value '0x0000' means the MMT receiving entity to execute the measurement only once. Other values mean the period for measurement. The unit is seconds

`measurement_report` — this field provides information for the measurement report. It has a server address where the MMT receiving entity should send measurement results and a template to be used for the measurement report.

`server_address` — indicates the location of server that receives transport data measurement results. The syntax and semantics are the same of `MMT_general_location_info` that is defined in 9.6.1

`report_type` — indicates type of measurement report request as shown in Table AMD1.10.

Table — AMD1.10 — Value of `report_type` field

Value	Description
0000 0000	report type is reception_quality_feedback
0000 0001	report type is NAM_feedback
0000 0010	report type is reception_quality_feedback and NAM_feedback
0000 0011 ~ 1111 1111	reserved for future use

Add the following sub-clause as 9.4.4.

9.4.4 ARQ Configuration (AC) message

9.4.4.1 Introduction

ARQ configuration information, which includes the policy to be adopted by the MMT sending entity and MMT receiving entity in the event of packet loss, shall be transmitted at the beginning of a session as the ARQ configuration (AC) message from the transmitting MMT sending entity to the MMT receiving entity either in-band or out-of-band. The syntax for AC message is shown in Table AMD1.11.

9.4.4.2 Syntax

Syntax of the AC message is shown in Table AMD1.11.

Table — AMD1.11 —AC message syntax

Syntax	Values	No. of bits	Mnemonic
AC_message() {			
message_id		16	
version		8	
length		16	
message_payload{			
flow_label_flag		1	boolean
if(flow_lable_flag == 1){			
fb_flow_label		7	uimsbf
} else {			
reserved	**'1111 11'**	7	bslbf
}			
delay_constrained_ARQ_flag	**N1**	1	boolean

Table — *(continued)*

Syntax	Values	No. of bits	Mnemonic
number_of_packet_id		7	uimsbf
for(i=0; i<N1; i++) {			
packet_id		16	uimsbf
rtx_window_timeout		32	uimsbf
}			
arq_server_address{			
MMT_general_location_info()		32	uimsbf
}			
}			
}			

9.4.4.3 Semantics

`message_id` — indicates AC message ID. The length of this field is 16 bits.

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

`length` — indicates the length of AC messages. The length of this field is 16 bits. It indicates the length of the AC message counted in bytes starting from the next field to the last byte of the AC message. The value '0' shall not be used.

`flow_label_flag` — indicates whether `fb_flow_label` exists. If the value is set to 1, the value of `rtx_flow_label` parameter is present.

`fb_flow_label` — indicates the flow label to be used when MMT receiving entity sends AF message. The `flow_label` will be allocated to guarantee higher priority for AF message along the delivery path. This parameter will be presented only if the value of `flow_label_flag` parameter is set to 1.

`delay_constrained_ARQ_flag` — when set to '1' this flag indicates that the server supports delay constrained ARQ.

`number_of_packet_id` — indicates the number of packet id that has lost packets.

`rtx_window_timeout` — indicates the retransmit window timeout. An MMT sending entity will keep an MMTP packet in buffer until the timeout, and thus available for retransmission. The timeout for a certain MMTP packet starts when MMT sending entity sends the MMTP packet. Thus MMT receiving entity can infer whether the MMTP packet is available by referencing the timestamp field in MMTP packet header. The unit is milliseconds

`arq_server_address` — indicates address of servers to which MMT receiving entity can send AF message to request lost MMTP packets.

NOTE `location_type of MMT_general_location_info()` will be restricted to 0x01, 0x02 and 0x05 for simplicity.

9.4.5 ARQ Feedback (AF) message

9.4.5.1 Introduction

In the event of packet loss, this loss is detected by MMT receiving entity. An ARQ Feedback (AF) message is generated according to the information of AC message, and then transmitted to MMT sending entity. When the MMT receiving entity detects that one or more packets have been lost, it forms a mask of up to 255 bytes where each bit in a byte corresponds to a sequence number of lost MMTP packets. This allows

AF message can report up to 255 × 8 lost packets in one AF message. The syntax for AF message is shown in Table AMD1.12.

The method of packet loss detection is out of scope of this standard.

9.4.5.2 Syntax

Syntax of the AF message is shown in Table AMD1.12.

Table — AMD1.12 —AF message syntax

Syntax	Values	No. of bits	Mnemonic
AF_message() {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
message_payload {			
argument_type		1	Boolean
delay_constrained_ARQ_mode		2	uimsbf
if(argument_type == 0) {			
if (delay_constrained_ARQ_mode == 01){		32	uimsbf
ARQ_feedback_timestamp		32	uimsbf
}		32	uimsbf
if (delay_constrained_ARQ_mode == 10){		8	uimsbf
propagation_delay		16	uimsbf
}		8	uimsbf
packet_counter			
masklength			
if (delay_constrained_ARQ_mode ==01){			
arrival_deadline			
}			
for(i=0; i<masklength; i++){			
mask_byte			
}			
}	N1		
if(argument_type == 1) {		7	uimsbf
number_of_packet_id		32	uimsbf
if (delay_constrained_ARQ_mode == 01){		32	uimsbf
ARQ_feedback_timestamp		16	uimsbf
}		32	uimsbf
if (delay_constrained_ARQ_mode == 10){		8	uimsbf
propagation_delay		16	uimsbf
}		8	uimsbf
for(i=0; i<N1; i++){			
packet_id			
packet_sequence_number			
masklength			
if (delay_constrained_ARQ_mode == 01){			
arrival_deadline			
}			
for(i=0; i<masklength; i++){			
mask_byte			
}			
}			
}			
}			
}			

9.4.5.3 Semantics

`message_id` — It indicates AF message ID. The length of this field is 16 bits.

`version` — It indicates the version of AF 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 AF messages. The length of this field is 16 bits. It indicates the length of the AF message counted in bytes starting from the next field to the last byte of the AF message. The value '0' shall not be used.

argument_type – This field indicates the type of argument the MMT receiving entity is using when requesting the lost packets to server. Valid values for this field are described in Table AMD1.4

Table — AMD1.7 — Value of argument_type

Value	Description
0	Packet counter based ARQ. MMT receiving entity send AF message with *packet_counter*.
1	Packet sequence number based ARQ. MMT receiving entity send AF message with *packet_id* and *packet_sequence_number*.

delay_constrained_ARQ_mode – This field indicates the type of delay constrained ARQ. Valid values for this field are described in Table AMD1.8.

Table — AMD1.8 — Value of delay_constrained_ARQ_mode

Value	Description
00	No time constrained ARQ. ARQ server does not need to consider any delay constraints when retransmitting the lost packets for this request.
01	Playout-time constrained ARQ. MMT receiving entity sends AF message with *ARQ_feedback_timestamp* and *arrival_deadline*, to help server decide whether to retransmit or not.
10	Delivery-time constrained ARQ. MMT receiving entity sends AF message with *propagation_delay* only, to help server decide whether to retransmit or not.
11	Reserved for future use

number_of_packet_id — This field indicates the number of packet id that has lost packets.

delay_constrained_ARQ_flag — This flag indicates the present of arrival_deadline field information.

ARQ_feedback_timestamp — indicates the NTP time at which the ARQ feedback is sent from the MMT receiving entity.

propagation_delay — propagation delay for the MMT packet to arrive at the MMT receiving entity. The MMT receiving entity calculates the propagation_delay by the subtracting the NTP time at the delivery instant of a MMT packet from the NTP time at the arrival instant of the MMT packet. The propagation_delay can be an average result of a propagation delay measured within the measurement_duration.

packet_id — This field is the integer value assigned to each Asset to distinguish packets of one Asset from another. Separate value will be assigned to signaling messages and FEC parity flows.

packet_sequence_number — This field corresponds to the packet_sequence_number of the first packet indicated by the mask_byte that is identified as having been detected to be lost and hence requiring re-transmission.

mask_length — It indicates the length of the data behind the mask in bytes.

arrival_deadline — indicates the deadline by which the retransmitted packet for the first lost packet should arrive at the MMT receiving entity for timely processing. This parameter represents the time increment from the ARQ_feedback_timestamp. The first 8 bits represents integer part and the last 8 bits represents fractional part.

`mask_byte` — Mask field, each bit correspond to a MMTP packet. If the packet behind the packet with `packet_id` is lost, then the corresponding bit will be set to '1'.

9.4.6 The reception quality feedback (RQF) message

9.4.6.1 Introduction

An MMT receiving entity can send the reception quality feedback to the MMT sending entity to inform the reception quality of the received MMTP packet flow by using RQF messages. An MMT receiving entity needs to keep track of reception quality per MMT sending entity.

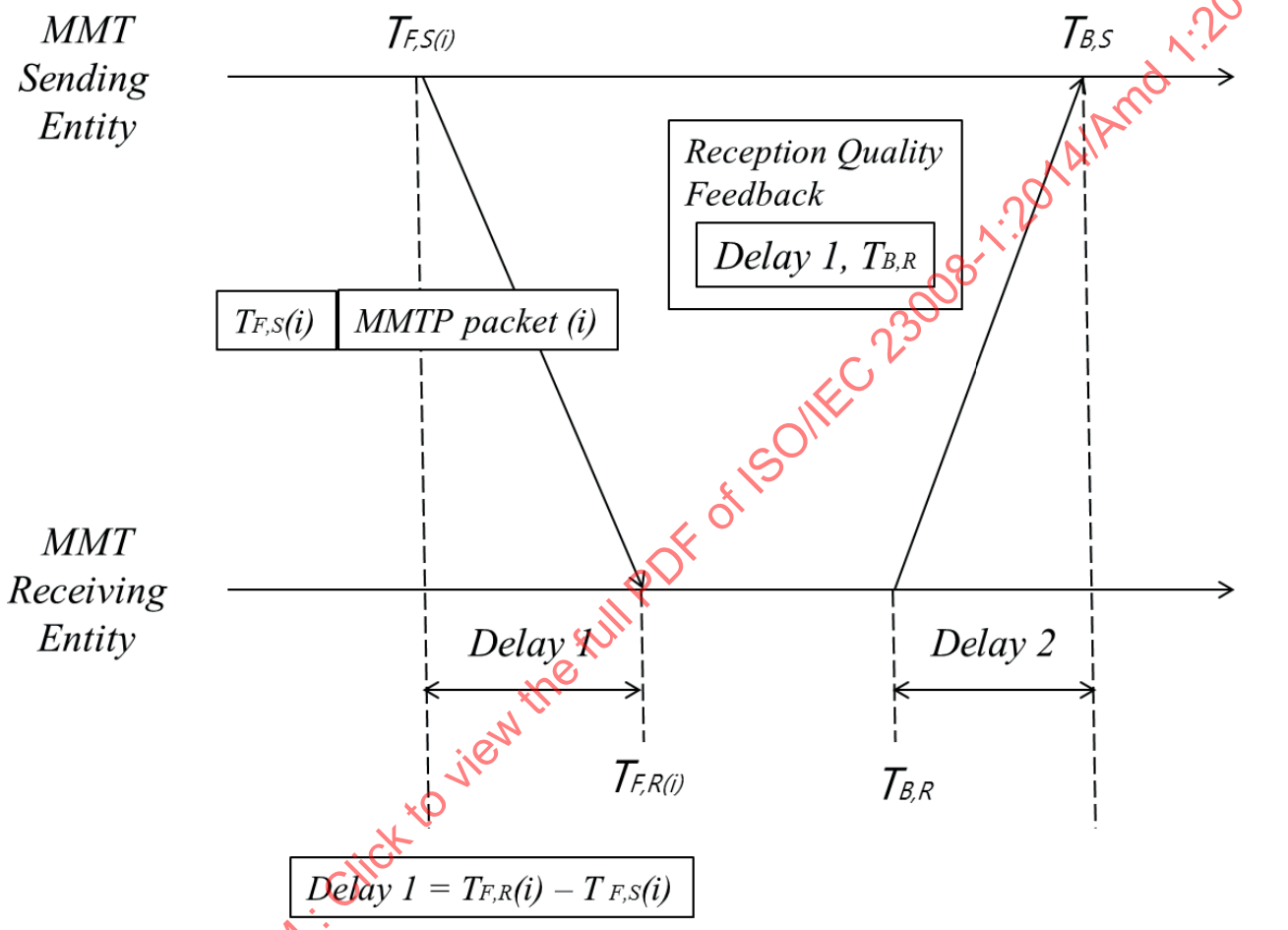

Figure — AMD1.6 — Round Trip Time (RTT) calculation

An MMT receiving entity provides information in the feedback to allow the MMT sending entity to calculate the Round-Trip Time (RTT) and the process is as shown in Figure AMD1. 6. In this Figure AMD1. 6, Delay1 is the delivery time of the MMTP packet from MMT sending entity to MMT receiving entity. Delay1 is calculated by the subtracting $T_{F,S}(i)$ (NTP time at delivery instant of i-th MMTP packet) from $T_{F,R}(i)$ (NTP time at arrival instant of i-th MMTP packet). The Delay2 is the delivery time for the MMTP packet from MMT receiving entity to MMT sending entity. The Delay2 is calculated by subtracting $T_{B,R}$ (NTP time at delivery instant of the feedback report, i.e. feedback_timestamp) from the $T_{B,S}$ (NTP time at arrival instant of the feedback report). Thus, the MMT sending entity can calculate the RTT by adding Delay1 and Delay2.

9.4.6.2 Syntax

The syntax of the RQF message is shown in Table AMD1. 13.

Table — AMD1.13 —RQF message syntax

Syntax	Values	No. of bits	Mnemonic
RQF_message ( ) {			
message_id		**16**	
version		**8**	unsigned short
length		**16**	
message_payload {			
packet_loss_ratio		**8**	**unsigned char**
inter_arrival_jitter		**32**	**unsigned integer**
max_transmission_delay		**32**	**unsigned integer**
min_transmission_delay		**32**	**unsigned integer**
RTT_parameter( ) {			
propagation_delay		**32**	**unsigned integer**
feedback_timestamp		**32**	**unsigned integer**
}			
}			
}			

9.4.6.3 Semantics

`message_id` — identifier of the RQF message. The length of this field is 16 bits.

`version` — version of the RQF message. An MMT receiving entity can use this field to check the version of a received message. The length of this field is 8 bits.

`length` — length of the RQF in bytes, counting from the first byte of next field to the last byte of the RQF message. The length of this field is 16 bits and the value '0' is not valid for this field.

`packet_loss_ratio` — ratio between the number of lost MMTP packets and the total number of transmitted packets. This value is equivalent to taking the integer part after multiplying the loss fraction by 256. The `packet_loss_ratio` is a result measured within the `measurement_duration`.

inter_arrival_jitter — deviation of the difference in packet spacing at the MMT receiving entity compared to the MMT sending entity for a pair of packets, measured in timestamp units. It can be estimated based on the time difference between the arrivals of adjacent MMTP packets. The `inter_arrival_jitter` is an average result measured within the `measurement_duration`.

`RTT_parameter` — parameter used for calculating the round trip time (RTT). RTT is the length of time required for the MMTP packet to be sent and the length of time it takes for an acknowledgement to be received. When computing the RTT, MMT sending entity records the time when the feedback is received. RTT is calculated by subtracting the `feedback_timestamp` from the recorded time and adding the `propagation_delay`.

`propagation_delay` — propagation delay for the MMTP packet to arrive at the MMT receiving entity. The MMT receiving entity calculates the `propagation_delay` by the subtracting the NTP time at the delivery instant of a MMTP packet from the NTP time at the arrival instant of the MMTP packet. The `propagation_delay` can be an average result of a propagation delay measured within the `measurement_duration`.

`feedback_timestamp` — NTP time at which the feedback is sent from the MMT receiving entity. This parameter is used to measure the propagation delay from the MMT receiving entity to the MMT sending entity.

`max_transmission_delay` — the maximum transmission delay for the MMTP packet to arrive at the MMT receiving entity. The MMT receiving entity calculates the transmission delay by the subtracting

the NTP time at the delivery instant of a MMTP packet from the NTP time at the arrival instant of the MMTP packet. The `max_transmission_delay` is the maximum `transmission_delay` measured within the `measurement_duration`.

`min_transmission_delay` — the minimum transmission delay for the MMTP packet to arrive at the MMT receiving entity. The MMT receiving entity calculates the transmission delay by the subtracting the NTP time at the delivery instant of a MMTP packet from the NTP time at the arrival instant of the MMTP packet. The `min_transmission_delay` is the minimum `transmission_delay` measured within the `measurement_duration`.

9.4.7 The ADC Message

9.4.7.1 Introduction

An ADC message carries information on ADC which defines QoS requirements and statistics of Asset for delivery, and their associated QoE quality information in alignment with ISO 23001-10 *Carriage of Quality Information in ISOBMFF File Format*. Additional operating points can be derived from stream sub-representation via a new sample_group_index structure. This is especially useful for low delay streaming applications where quality can be traded for delay reduction. This information can be used by the MMT-aware intermediate network entities for QoS-managed delivery of Assets. To provide more accurate information, MMT sending entity can update parameter values within ADC message and send it periodically or aperiodically. ADC information delivery can be done both in per-Asset basis and per-MPU basis considering its size.

9.4.7.2 Syntax

Syntax of the ADC message is defined in Table AMD1.15.

Table — AMD1.15 — ADC message syntax

Syntax	Value	No. of bits	Mnemonic
ADC_message () {			
message_id		**16**	**uimsbf**
version		**8**	**uimsbf**
length		**32**	**uimsbf**
message_payload {			
validity_start_time		**32**	**uimsbf**
validity_duration		**32**	**uimsbf**
ADC_level_flag		**1**	**boolean**
flow_label_flag		**1**	**boolean**
reserved		**6**	**uimsbf**
if (ADC_level_flag == 1){			
MPU_sequence_number		**32**	**uimsbf**
}			
packet_id		**16**	**uimsbf**
qos_descriptor{			**uimsbf**
loss_tolerance		**8**	**bslbf**
jitter_sensitivity		**8**	**bslbf**
class_of_service		**1**	**boolean**
bidirection_indicator		**1**	**boolean**
reserved		**6**	**bslbf**
}			
qoe_descriptor{			
n_samples		**16**	**uimsbf**
for (i=0;i<N1; i++) {			
sample_group_index		**16**	**uimsbf**
}			
spatial_quality		**16**	**uimsbf**
temporal_quality		**16**	**uimsbf**
aggregate_rate		**32**	**uimsbf**
}			
if (class_of_service == 1) bitstream_descriptor_vbr{			
sustainable_rate		**16**	**uimsbf**
buffer_size		**16**	**uimsbf**
peak_rate		**16**	**uimsbf**
max_MFU_size		**8**	**uimsbf**
mfu_period		**8**	**uimsbf**
}else bitstream_descriptor_cbr{			
peak_rate		**16**	**uimsbf**
max_MFU_size		**8**	**uimsbf**
mfu_period		**8**	**uimsbf**
}			
If(flow_label_flag == 1) {			
flow_label		**7**	**uimsbf**
reserved		**1**	**uimsbf**
}			
}			
}			

9.4.7.3 Semantics

message_id — indicates the identifier of the ADC messages.

version — indicates the version of the ADC messages.

length — a 32-bit field for conveying the length of the ADC message in bytes, counting from the beginning of the next field to the last byte of the ADC message. The value '0' is not valid for this field.

validity_start_time — indicates the time when the updated ADC message starts to be valid in UTC.

validity_period_duration — indicates the validity period duration of the updated ADC message from the validity_start_time in milliseconds. The value of this parameter within this ADC message is valid until this duration and the MANE doesn't need to capture the newer ADC message necessarily.

ADC_level_flag (1 bit) — indicates whether included ADC information is for an Asset or for a MPU. If set to '0', ADC signaling message includes information for an Asset. If set to '1', it includes ADC information for single MPU.

loss_tolerance — indicates required loss tolerance of the Asset for the delivery. The value of loss_tolerance attribute is listed in Table AMD1.1.

jitter_sensitivity — indicates required jitter level of underlying delivery network for the Asset delivery between end-to-end. The value of jitter_sensitivity attribute is listed in Table AMD1.2.

class_of_service — classifies the services in different classes and manage each type of bitstream with a particular way. For example, MANE can manage each type of bitstream with a particular way. This field indicates the type of bitstream attribute as listed in Table AMD1.3.

Bidirection_indicator — If set to '1', bidirectional delivery is required. If set to '0', bidirectional delivery is not required.

Bitstream_descriptorVBRType — when class_of_service is '1', "Bitstream_descriptorVBRType" shall be used for "Bitstream_descriptorType".

Bitstream_descriptorCBRType — when class_of_service is '0', "Bitstream_descriptorCBRType" shall be used for "Bitstream_descriptorType".

n_samples — defines the samples associated with this particular operating point.

spatial_quality — defines the spatial quality associated with samples that is conforming to the ISOBMFF quality format. Examples are PSNR, MSE.

temporal_quality — defines the temporal quality, or distortion associated with samples that is computed from ISOBMFF frame significance values.

aggregate_rate — defines the bit rate associated with the operating point.

sustainable_rate — defines the minimum bitrate that shall be guaranteed for continuous delivery of the Asset. The sustainable_rate corresponds to drain rate in token bucket model. The sustainable_rate is expressed in bytes per second.

buffer_size — defines the maximum buffer size for delivery of the Asset. The buffer absorbs excess instantaneous bitrate higher than the sustainable_rate and the buffer_size shall be large enough to avoid overflow. The buffer_size corresponds to bucket depth in token bucket model. Buffer_size of a CBR (constant bit rate) Asset shall be zero. The buffer_size is expressed in bytes

peak_rate — defines peak bitrate during continuous delivery of the Asset. The peak_rate is the highest bitrate during every MFU_period. The peak_rate is expressed in bytes per second.

MFU_period — defines period of MFUs during continuous delivery of the Asset. The MFU_period measured as the time interval of sending time between the first byte of two consecutive MFUs. The MFU_period is expressed in millisecond.

max_MFU_size — indicates the maximum size of MFU, which is MFU_period*peak_rate. The max_MFU_size is expressed in byte.

flow_label — indicates the flow identifier. The application can perform per-flow QoS operations in which network resources are temporarily reserved during the session. A flow is defined to be a bitstream or a group of bitstreams whose network resources are reserved according to transport characteristics or ADC in Package. It is an implicit serial number from '0' to '127'. An arbitrary number is assigned temporarily during a session and refers to every individual flow for whom a decoder (processor) is assigned and network resource could be reserved.

packet_id (16 bits) — this field is an integer value that can be used to distinguish one Asset from another. The value of this field is derived from the asset_id of the Asset where this packet belongs to. The mapping between the packet_id and the asset_id is signaled by the Package Table as part of a signalling message (see 9.3.4). The packet_id is unique throughout the lifetime of the delivery session and for all MMT flows delivered by the same MMT sending entity. For AL-FEC, the mapping between packet_id and the FEC repair flow is provided in the AL-FEC message (see C.6).

Replace the Table 34 in 9.7 with Table AMD1.9.

Table — AMD1.9 — Message Identifier (message_id) Values

Value	Description
0x0000	PA message
0x0001 ~ 0x000F	MPI messages. For a Package, 16 contiguous values are allocated to MPI messages. If the value % 16 equals 15, the MPI message carries complete PI. If the value %16 equals N where N = 0 ~ 14, the MPI message carries Subset-N PI.
0x0010 ~ 0x001F	MPT messages. For a package, 16 contiguous values are allocated to MPT messages. If the value % 16 equals 15, the MPT message carries complete MPT. If the value %16 equals N where N = 0 ~ 14, the MPT message carries Subset-N MPT.
0x0200	CRI message
0x0201	DCI message
0x0202	SSWR message
0x0203	AL_FEC messgage
0x0204	HRBM message
0x0205	MC message
0x0206	AC message
0x0207	AF message
0x0208	RQF message
0x0209	ADC message
0x020A ~0x7FFF	reserved for ISO use
0x8000 ~ 0xFFFF	reserved for private use

Replace the Table 35 in 9.7 with: Table AMD1.10.

Table — AMD1.10 — Table Identifier (table_id) Values

Value	Description
0x00	PA table
0x01	Main PI (Subset-0 MPI table)
0x02 ~ 0x0F	Subset-1 MPI table ~ Subset-14 MPI table
0x10	Complete MPI table
0x11 ~ 0x1F	Subset-0 MP table (SUBSET_0_MPT_TABLE_ID) ~ Subset-14 MP table
0x20	Complete MP table
0x21	CRI table
0x22	DCI table
0x23	SIT table
0x24 ~0x7F	reserved for ISO use
0x80 ~ 0xFF	reserved for private use

Add the following XML syntax in Annex B.

Annex B
(normative)
XML syntax for ADC

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xs:schema xmlns:xs=”http://www.w3.org/2001/
XMLSchema” xmlns=”urn:mpeg:MMT:schema:Signalling:2013”
targetNamespace=”urn:mpeg:MMT:schema:Signalling:2013” elementFormDefault=”qualified”
attributeFormDefault=”unqualified”>
      <xs:annotation>
           <xs:appinfo>MMT Signalling</xs:appinfo>
           <xs:documentation xml:lang=”en”>This schema defines the syntax for MMT
Signalling messaging that </xs:documentation>
      </xs:annotation>
      <xs:element name=”Message”>
           <xs:complexType>
                <xs:choice minOccurs=”1” maxOccurs=”1”>

                ……
                ……
                     <xs:element name=”ADC_message”>
                          <xs:complexType>
                               <xs:attribute name=”ADC_level_flag” type=”xs:boolean”/>
                               <xs:attribute name=”MPU_sequence_number ‟ type=”xs:integer”
minOccurs=”0”/>
                               <xs:attribute name=”packet_id” type=”xs:integer”/>

                               <xs:attribute name=”loss_tolerance” type=”xs:integer”/>
                              <xs:attribute name=”jitter_sensitivity” type=”xs:integer”/>
                              <xs:attribute name=”class_of_service” type=”xs:boolean”/>

                              <xs:attribute name=”bidirection_indicator”
type=”xs:boolean”/>
                              <xs:attribute name=”flow_label” type=”xs:integer”/>
                              <xs:attribute name=”sustainable_rate” type=”xs:integer”
minOccurs=”0”/>
                              <xs:attribute name=”buffer_size” type=”xs:integer”
minOccurs=”0”/>
                              <xs:attribute name=”peak_rate” type=”xs:integer”/>
                              <xs:attribute name=”max_MFU_size” type=”xs:integer”/>
                              <xs:attribute name=”mfu_period” type=”xs:integer”/>

                              <xs:complexType name=”mmt:OperationPointCharacteristics”>
```