INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.222.0

Amendment 1
(12/2002)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Transmission multiplexing and synchronization

---

Information technology – Generic coding of moving pictures and associated audio information: Systems

Amendment 1: Carriage of metadata over ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams

ITU-T Recommendation H.222.0 (2000) – Amendment 1

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	**H.220–H.229**
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
SYSTEMS AND TERMINAL EQUIPMENT FOR AUDIOVISUAL SERVICES	H.300–H.399
SUPPLEMENTARY SERVICES FOR MULTIMEDIA	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569

For further details, please refer to the list of ITU-T Recommendations.

INTERNATIONAL STANDARD ISO/IEC 13818-1
ITU-T RECOMMENDATION H.222.0

Information technology – Generic coding of moving pictures and associated audio information: Systems

Amendment 1

Carriage of metadata over ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams

Summary

This amendment extends the functionality of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 to support the carriage of metadata. It incorporates the modifications introduced by ITU-T Rec. H.222.0 (2000) Cor.3 (06/2003) | ISO/IEC 13818-1:2000 Tech. Cor. 3:2003, which corrected the omission of the 8-bit reserved field in the section syntax for transport of metadata.

Source

Amendment 1 to ITU-T Recommendation H.222.0 (2000) was prepared by ITU-T Study Group 16 (2001-2004) and approved on 14 December 2002. An identical text is also published as ISO/IEC 13818-1, Amendment 1.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Generic coding of moving pictures and associated audio information: Systems

Amendment 1

Carriage of metadata over ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams

1) Subclause 2.1

Insert the following new definitions and renumber existing ones accordingly:

2.1.28 metadata: Information to describe audiovisual content and data essence in a format defined by ISO or any other authority.

2.1.29 metadata access unit: A global structure within metadata that defines the fraction of metadata that is intended to be decoded at a specific instant in time. The internal structure of a metadata Access Unit is defined by the format of the metadata.

2.1.30 metadata application format: Identifies the format of the application that uses the metadata; signals application specific information for transport of metadata.

2.1.31 metadata decoder configuration information: Data needed by a receiver to decode a specific metadata service. Depending on the format of the metadata, decoder configuration information may or may not be needed.

2.1.32 metadata format: Identifies the coding format of metadata.

2.1.33 metadata service: Coherent set of metadata of the same format delivered to a receiver for a specific purpose.

2.1.34 metadata service id: Identifier of a specific metadata service; used for some transport methods of the metadata.

2.1.35 metadata stream: The concatenation or collection of metadata Access Units from one or more metadata services.

2) Table 2-18

Define a tag for a metadata stream by replacing Table 2-18 in clause 2:

Table 2-18 – Stream_id assignments

Stream_id	Note	Stream coding
1011 1100	(1)	program_stream_map
1011 1101	(2)	private_stream_1
1011 1110		padding_stream
1011 1111	(3)	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	(3)	ECM_stream
1111 0001	(3)	EMM_stream
1111 0010	(5)	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSM-CC_stream
1111 0011	(2)	ISO/IEC_13522_stream
1111 0100	(6)	ITU-T Rec. H.222.1 type A
1111 0101	(6)	ITU-T Rec. H.222.1 type B
1111 0110	(6)	ITU-T Rec. H.222.1 type C
1111 0111	(6)	ITU-T Rec. H.222.1 type D
1111 1000	(6)	ITU-T Rec. H.222.1 type E
1111 1001	(7)	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100		metadata stream
1111 1101		reserved data stream
1111 1110		reserved data stream
1111 1111	(4)	program_stream_directory

The notation x means that the values '0' or '1' are both permitted and results in the same stream type. The stream number is given by the values taken by the x's.

NOTE 1 – PES packets of type program_stream_map have unique syntax specified in 2.5.4.1.

NOTE 2 – PES packets of type private_stream_1 and ISO/IEC_13552_stream follow the same PES packet syntax as those for ITU-T Rec. H.262 | ISO/IEC 13818-2 video and ISO/IEC 13818-3 audio streams.

NOTE 3 – PES packets of type private_stream_2, ECM_stream and EMM_stream are similar to private_stream_1 except no syntax is specified after PES_packet_length field.

NOTE 4 – PES packets of type program_stream_directory have a unique syntax specified in 2.5.5.

NOTE 5 – PES packets of type DSM-CC_stream have a unique syntax specified in ISO/IEC 13818-6.

NOTE 6 – This stream_id is associated with stream_type 0x09 in Table 2-29.

NOTE 7 – This stream_id is only used in PES packets, which carry data from a Program Stream or an ISO/IEC 11172-1 System Stream, in a Transport Stream (refer to 2.4.3.7).

3) Table 2-26

Define a metadata_section tag by replacing Table 2-26 in clause 2:

Table 2-26 – table_id assignment values

Value	Description
0x00	Program_association_section
0x01	Conditional_access_section (CA_section)
0x02	TS_program_map_section
0x03	TS_description_section
0x04	ISO_IEC_14496_scene_description_section
0x05	ISO_IEC_14496_object_descriptor_section
0x06	Metadata_section
0x07-0x37	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved
0x38-0x3F	Defined in ISO/IEC 13818-6
0x40-0xFE	User private
0xFF	Forbidden

4) Table 2-29

Define tags for a metadata carried in PES packets, metadata_sections and DSM-CC by replacing Table 2-29 in clause 2:

Table 2-29 – Stream type assignments

Value	Description
0x00	ITU-T | ISO/IEC Reserved
0x01	ISO/IEC 11172 Video
0x02	ITU-T Rec. H.262 | ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x03	ISO/IEC 11172 Audio
0x04	ISO/IEC 13818-3 Audio
0x05	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 private_sections
0x06	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 PES packets containing private data
0x07	ISO/IEC 13522 MHEG
0x08	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A DSM-CC
0x09	ITU-T Rec. H.222.1
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 auxiliary
0x0F	ISO/IEC 13818-7 Audio with ADTS transport syntax
0x10	ISO/IEC 14496-2 Visual
0x11	ISO/IEC 14496-3 Audio with the LATM transport syntax as defined in ISO/IEC 14496-3/Amd.1
0x12	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in PES packets
0x13	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in ISO/IEC14496_sections.
0x14	ISO/IEC 13818-6 Synchronized Download Protocol

Table 2-29 – Stream type assignments

Value	Description
0x15	Metadata carried in PES packets using the Metadata Access Unit Wrapper defined in 2.12.4.1
0x16	Metadata carried in metadata_sections
0x17	Metadata carried in ISO/IEC 13818-6 (DSM-CC) Data Carousel
0x18	Metadata carried in ISO/IEC 13818-6 (DSM-CC) Object Carousel
0x19	Metadata carried in ISO/IEC 13818-6 Synchronized Download Protocol using the Metadata Access Unit Wrapper defined in 2.12.4.1
0x1A-0x7F	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved
0x80-0xFF	User Private

5) Table 2-39

Define tags for a metadata related descriptors by replacing Table 2-39 in clause 2:

Table 2-39 – Program and program element descriptors

descriptor_tag	TS	PS	Identification
0	n/a	n/a	Reserved
1	n/a	n/a	Reserved
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	Video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	System_clock_descriptor
12	X	X	Multiplex_buffer_utilization_descriptor
13	X	X	Copyright_descriptor
14	X		Maximum_bitrate_descriptor
15	X	X	Private_data_indicator_descriptor
16	X	X	Smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Defined in ISO/IEC 13818-6
27	X	X	MPEG-4_video_descriptor
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor
31	X	X	FMC_descriptor
32	X	X	External_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor

Table 2-39 – Program and program element descriptors

descriptor_tag	TS	PS	Identification
35	X		MultiplexBuffer_descriptor
36	X	X	Content_labeling_descriptor
37 38 39	X X X	X X X	Metadata_pointer_descriptor Metadata_descriptor Metadata_STD_descriptor
40-63	n/a	n/a	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Reserved
64-255	n/a	n/a	User Private

6) New subclauses after subclause 2.6.55

Add the following subclauses after subclause 2.6.55:

2.6.56 Content labelling descriptor

The content labelling descriptor assigns a label to content; the label can be used by metadata to reference the associated content. This label, the content_reference_id_record, is metadata application format specific. The content labelling descriptor is associated with a content segment. For the purpose of this clause, a content segment is defined as a portion in time of a program, an elementary stream (such as audio or video) or any combination of programs or elementary streams. The descriptor may be included in the PMT in the descriptor loop for either the program or an elementary stream, but may also be contained in tables not defined in this Specification, for example tables to describe segments of programs or elementary streams. The content labelling descriptor also provides information on which content time base is used and on the offset between the content time base and the metadata time base. When the Normal Play Time (NPT) concept of DSM-CC, as specified in IEC/ISO 13818-6, is used as the content time base, the ID of the NPT time base is provided. The descriptor allows for carriage of private data.

Table Amd.1-1 – Content labelling descriptor

Syntax	No. of bits	Mnemonic
Content_labeling_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
content_reference_id_record_flag	**1**	**bslbf**
content_time_base_indicator	**4**	**uimsbf**
reserved	**3**	**bslbf**
if (content_reference_id_record_flag == '1'){		
content_reference_id_record_length	**8**	**uimsbf**
for (i=0; i<content_reference_id_record_length;i++){		
content_reference_id_byte	**8**	**bslbf**
}		
}		
if (content_time_base_indicator== 1|2){		
reserved	**7**	**bslbf**
content_time_base_value	**33**	**uimsbf**
reserved	**7**	**bslbf**
metadata_time_base_value	**33**	**uimsbf**
}		
if (content_time_base_indicator== 2){		
reserved	**1**	**bslbf**
contentId	**7**	**uimsbf**
}		
if (content_time_base_indicator==3|4|5|6|7){		
time_base_association_data_length	**8**	**uimsbf**
for (i=0; i< time_base_association_data_length;i++){		
reserved	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++){		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.57 Semantic definition of fields in content labelling descriptor

metadata_application_format: The metadata_application_format is a 16-bit field, coded as defined in Table Amd.1-2, that specifies the application responsible for defining usage, syntax and semantics of the content_reference_id record and of any other privately defined fields in this descriptor. See also subclause 2.12.1. The value 0xFFFF indicates that the format is signalled by the value carried in the metadata_application_format_identifier field.

Table Amd.1-2 – metadata_application_format

Value	Description
0x0000-0x00FF	Reserved
0x0100-0xFFFE	User defined
0xFFFF	Defined by the metadata_application_format_identifier field

metadata_application_format_identifier: The coding of this 32-bit field is fully equivalent to the coding of the format_identifier field in the registration_descriptor, as defined in subclause 2.6.8.

NOTE – The assigned Registration Authority for the format_identifier field is SMPTE.

content_reference_id_record_flag: The content_reference_id_record_flag is a 1-bit flag that signals the presence of a content_reference_id_record in this descriptor.

content_time_base_indicator: The content_time_base_indicator is a 4-bit field which specifies the used content time base. If the descriptor is associated with a program, then the content time base applies to all streams that are part of that program. A value of 1 indicates usage of the STC, while a value of 2 indicates usage of NPT, the Normal Play Time as defined in ISO/IEC 13818-6. The values between 8 and 15 indicate usage of a privately defined content time base. If coded with a value of 0, no content time base is defined in this descriptor. If no content time base is specified for a

program or stream, then the mapping of time references in the metadata to the content is not defined in this Specification.

Table Amd.1-3 – Content_time_base_indicator values

Value	Description
0	No content time base defined in this descriptor
1	Use of STC
2	Use of NPT
3-7	Reserved
8-15	Use of privately defined content time base

content_reference_id_record_length: The content_reference_id_record_length is an 8-bit field that specifies the number of content_reference_id_bytes immediately following this field. This field shall not be coded with the value 0.

content_reference_id_byte: The content_reference_id_byte is part of a string of one or more contiguous bytes that assigns one or more reference identifications (labels) to the content to which this descriptor is associated. The format of this byte string is defined by the body indicated by the coded value in the metadata_application_format field.

content_time_base_value: The content_time_base_value is a 33-bit field that specifies a value in units of 90 kHz of the content time base indicated by the content_time_base_indicator field.

metadata_time_base_value: The metadata_time_base_value is a 33-bit field that is coded in units of 90 kHz. The field is coded with the value of the metadata time base at the instant in time in which the time base indicated by content_time_base_indicator reaches the value encoded in the content_time_base_value field. Note that the metadata time base may use any time scale, but that its value is to be coded in units of 90 kHz. For example, if a SMPTE type of time code is used, then the number of hours, minutes, seconds and frames is expressed in the corresponding number of 90 kHz units.

contentId: The contentId is a 7-bit field that specifies the value of the content_Id field in the NPT Reference Descriptor for the applied NPT time base.

time_base_association_data_length: The time_base_association_data_length is an 8-bit field that specifies the number of reserved bytes immediately following this field. The reserved bytes can be used to carry time base association data for time bases defined in future.

private_data_byte: The private_data_byte is an 8-bit field. The private_data_bytes represent data, the format of which is defined privately. These bytes can be used to provide additional information as deemed appropriate. The use of these bytes is defined by the metadata application format.

2.6.58 Metadata pointer descriptor

The metadata pointer descriptor points to a single metadata service and associates this metadata service with audiovisual content in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream. The metadata is associated with the content within the context of the descriptor. The context is defined by the location of the descriptor. In a transport stream, the descriptor may be located in the PMT in the descriptor loop for either the program or an elementary stream, but may also be located in tables not defined in this Specification, such as tables describing bouquets of broadcast services. The metadata may be located in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, but the same metadata may also be provided on alternative locations, such as the Internet.

The descriptor may contain location information of metadata that is not carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream; the coding of the location information is metadata application format specific. The descriptor allows for carriage of private data.

For metadata carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, the descriptor specifies the tools used for such carriage. If the metadata is carried in PES packets, metadata sections, or ISO/IEC 13818-6 synchronized download sections, the metadata_service_id field identifies the metadata service in the referenced metadata stream. If a ISO/IEC 13818-6 carousel is used to carry the metadata, then the private data may provide information to signal the metadata service, such as the applied value of the module_id for carriage of the metadata in a data carousel, and the file name of the metadata when the object carousel is used.

Receivers should be aware that multiple metadata services may be pointed to from the same program or audiovisual stream (as defined by the context of the descriptor). A unique metadata pointer descriptor shall be used to point to each

metadata service used by the program or audiovisual stream. Similarly, the same metadata service can be pointed to from several programs or audiovisual streams by using a separate metadata pointer descriptors for each association.

Table Amd.1-4 – Metadata pointer descriptor

Syntax	No. of bits	Mnemonic
Metadata_pointer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
metadata_locator_record_flag	**1**	**bslbf**
MPEG_carriage_flags	**2**	**uimsbf**
reserved	**5**	**bslbf**
if (metadata_locator_record_flag == '1'){		
metadata_locator_record_length	**8**	**uimsbf**
for (i=0; i< metadata_locator_record_length;i++){		
metadata_locator_record_byte	**8**	**bslbf**
}		
}		
if (MPEG_carriage_flags == 0|1|2){		
program_number	**16**	**uimsbf**
}		
if (MPEG_carriage_flags == 1){		
transport_stream_location	**16**	**uimsbf**
transport_stream_id	**16**	**uimsbf**
}		
for (i=0; i<N;i++){		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.59 Semantic definition of fields in metadata pointer descriptor

metadata_application_format: The metadata_application_format is a 16-bit field that specifies the application responsible for defining usage, syntax and semantics of the metadata_locator_record record and any other privately defined fields in this descriptor. The coding of this field is defined in Table Amd.1-2 in subclause 2.6.57.

metadata_application_format_identifier: The coding of this field is defined in subclause 2.6.57.

metadata_format: The metadata_format is an 8-bit field that indicates the format and coding of the metadata. The coding of this field is specified in Table Amd.1-5.

Table Amd.1-5 – Metadata format values

Value	Description
0x00-0x0F	Reserved
0x10	ISO/IEC 15938-1 TeM
0x11	ISO/IEC 15938-1 BiM
0x12-0x3E	Reserved
0x3F	Defined by metadata application format
0x40-0xFE	Private use
0xFF	Defined by metadata_format_identifier field

The values 0x10 and 0x11 identify ISO/IEC 15938-1 defined data. The value 0x3F indicates that the format is defined by the body indicated by the metadata_application_format field. The values in the inclusive range of 0x40 up to 0xFE

are available to signal use of private formats. The value 0xFF indicates that the format is signalled by the metadata_format_identifier field.

metadata_format_identifier: The coding of this 32-bit field is fully equivalent to the coding of the format_identifier field in the registration_descriptor, as defined in subclause 2.6.8.

NOTE – SMPTE is assigned as Registration Authority for the format_identifier field.

metadata_service_id: This 8-bit field references the metadata service. It is used for retrieving a metadata service from within a metadata stream.

metadata_locator_record_flag: The metadata_locator_record_flag is a 1-bit field which, when set to '1' indicates that associated metadata is available on a location outside of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, specified in a metadata_locator_record.

MPEG_carriage_flags: The MPEG_carriage_flags is a 2-bit field which specifies if the metadata stream containing the associated metadata service is carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, and if so, whether the associated metadata is carried in a Transport Stream or Program Stream. The coding of this field is defined in Table Amd.1-6.

Table Amd.1-6 – MPEG_carriage_flags

Value	Description
0	Carriage in the same Transport Stream where this metadata pointer descriptor is carried.
1	Carriage in a different Transport Stream from where this metadata pointer descriptor is carried.
2	Carriage in a Program Stream. This may or may not be the same Program Stream in which this metadata pointer descriptor is carried.
3	None of the above.

metadata_locator_record_length: The metadata_locator_record_length is an 8-bit field that specifies the number of metadata_locator_record_bytes immediately following. This field shall not be coded with the value 0.

metadata_locator_record_byte: The metadata_locator_record_byte is part of a string of one or more contiguous bytes that form the metadata locator record. This record specifies one or more locations outside of an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream. The format of the metadata locator record is defined by the metadata application signalled by the metadata_application_format field. The record may for example contain Internet URLs that specify where the metadata can be found, possibly in addition to their location(s) in the Transport Stream. If the MPEG_carriage_flags is coded with the value 0, 1 or 2 and the metadata locator record is present, then this signals alternative locations for the same metadata.

program_number: The program_number is a 16-bit field that identifies the program_number of the MPEG-2 program in the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream in which associated metadata is carried. If the MPEG_carriage_flags have the value 0, then the transport stream is the current one, and if the MPEG_carriage_flags have the value 1, it is the transport stream signalled by the fields transport_stream_location and transport_stream_id.

transport_stream_location: The transport_stream_location is a 16-bit field that is defined privately. For example, this field may be used by applications to signal the original_network_id defined by ETSI.

transport_stream_id: The transport_stream_id is a 16-bit field that identifies the Transport Stream in which associated metadata is carried.

private_data_byte: The private_data_byte is an 8-bit field. The private_data_bytes represent data, the format of which is defined privately. These bytes can be used to provide additional information as deemed appropriate.

2.6.60 Metadata descriptor

The metadata descriptor specifies parameters of a metadata service carried in an MPEG-2 TS or PS. In an MPEG-2 TS, the descriptor is included in the PMT in the descriptor loop for the elementary stream that carries the metadata service. The descriptor specifies the format of the associated metadata, and contains the value of the metadata_service_id to identify the metadata service to which the metadata descriptor applies. As needed, the descriptor can convey information to identify the metadata service from a collection of metadata transmitted in a DSM-CC carousel. Optionally metadata application format specific private data can be carried.

The metadata descriptor also signals whether decoder configuration is required and is able to carry the decoder configuration bytes, but this is only practical if the number of these bytes is small. If the decoder configuration information is too large to be carried by the descriptor, it shall be contained in a metadata service. This may be within the metadata service itself, or in another metadata service within the same program. Identification of the metadata service that contains the decoder configuration is provided by the metadata descriptor. If a DSM-CC carousel is used to carry the decoder configuration, then information can be provided how to retrieve the decoder configuration from the carousel.

Table Amd.1-7 – Metadata descriptor

Syntax	No. of bits	Mnemonic
Metadata_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format == 0xFFFF) {		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
decoder_config_flags	**3**	**bslbf**
DSM-CC_flag	**1**	**bslbf**
reserved	**4**	**bslbf**
if (DSM-CC_flag == '1'){		
service_identification_length	**8**	**uimsbf**
for(i=0; i<service_identification_length; i++) {		
service_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '001') {		
decoder_config_length	**8**	**uimsbf**
for(i=0; i<decoder_config_length; i++) {		
decoder_config_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '011') {		
dec_config_identification_record_length	**8**	**uimsbf**
for(i=0;i<dec_config_id_record_length;i++) {		
dec_config_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '100') {		
decoder_config_metadata_service_id	**8**	**uimsbf**
}		
if (decoder_config_flags == '101'|'110') {		
reserved_data_length	**8**	**uimsbf**
for(i=0;i<reserved_data_length;i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++) {		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.61 Semantic definition of fields in metadata descriptor

metadata_application_format: The metadata_application_format is a 16-bit field that specifies the application responsible for defining usage, syntax and semantics of the service_identification_record and any privately defined bytes in this descriptor. The coding of this field is defined in Table Amd.1-2 in subclause 2.6.57.

metadata_application_format_identifier: The coding of this field is defined in subclause 2.6.57.

metadata_format: The coding of this field is defined in subclause 2.6.59.

metadata_format_identifier: The coding of this field is defined in subclause 2.6.59.

metadata_service_id. This 8-bit field identifies the metadata service to which this metadata descriptor applies.

decoder_config_flags: The decoder_config_flags is a 3-bit field which indicates whether and how decoder configuration information is conveyed.

Table Amd.1-8 – decoder_config_flags

Value	Description
000	No decoder configuration is needed.
001	The decoder configuration is carried in this descriptor in the decoder_config_byte field.
010	The decoder configuration is carried in the same metadata service as to which this metadata descriptor applies.
011	The decoder configuration is carried in a DSM-CC carousel. This value shall only be used if the metadata service to which this descriptor applies is using the same type of DSM-CC carousel.
100	The decoder configuration is carried in another metadata service within the same program, as identified by the decoder_config_metadata_service_id field in this metadata descriptor.
101, 110	Reserved.
111	Privately defined.

DSM-CC_flag: This is a one-bit flag that is set to '1' if the stream with which this descriptor is associated is carried in an ISO/IEC 13818-6 data or object carousel.

NOTE 1 – The use of the object or data carousel is indicated by the applied stream-type value for this metadata stream.

service_identification length: This field specifies the number of service_identification_record_bytes immediately following.

service_identification_record_byte: This byte is part of a string of one or more contiguous bytes that specify the service_identification_record. This record contains data on retrieval of the metadata service from a DSM-CC carousel. The format of the metadata locator record is defined by the application indicated by the metadata application format. When a DSM-CC object carousel is used, the record may for example comprise the unique object identifier (the IOP:IOR() from ISO/IEC 13818-6 DSM-CC, subclauses 11.3.1 and 5.7.2.3) for the metadata service. Similarly, in case of a DSM-CC data carousel, the record can for example provide the transaction_id and the module_id of the metadata service.

decoder_config_length: This field specifies the number of decoder_config_bytes immediately following.

decoder_config_byte: These bytes comprise the decoder configuration information. This sequence of bytes comprises the configuration information needed by the receiver to decode this service. It is intended that carriage in the metadata descriptor is only used when the configuration information is very small.

decoder_config_DSM-CC_id: This is the download identifier of the decoder configuration information when it is transmitted in a DSM-CC data carousel, or the object identifier of the decoder configuration information if it is carried in a DSM-CC object carousel.

NOTE 2 – The use of the object or data carousel is indicated by the applied stream-type value for this metadata stream.

dec_config_identification_record_length: This field specifies the immediately following number of dec_config_identification_record_bytes.

dec_config_identification_record_byte: This byte is part of a string of one or more contiguous bytes that specify the dec_config_identification_record. This record specifies how to retrieve the required decoder configuration from a DSM-CC carousel. The format of the metadata locator record is defined by the metadata application format. When a DSM-CC object carousel is used, the record may for example comprise the unique object identifier (the IOP:IOR() from ISO/IEC 13818-6 DSM-CC, subclauses 11.3.1 and 5.7.2.3) for the decoder configuration. Similarly, in case of a DSM-CC data carousel, the record may for example provide the transaction_id and the module_id of the decoder configuration.

decoder_config_metadata_service_id: This is the value of the metadata_service_id that is assigned to the metadata service that contains the decoder configuration. The metadata service indicated by the decoder_config_metadata_service_id and the metadata service that uses that decoder configuration shall be in the same program. Hence in a Transport Stream, the metadata descriptors for both these metadata services shall be in the same

PMT. The metadata descriptor of the metadata service indicated by the decoder_config_metadata_service_id shall have a decoder_config_flag field with a value of either '001', '010' or '011'.

reserved_data_length: This field specifies the number of reserved bytes immediately following.

private_data_byte: The private_data_byte is an 8-bit field. The private_data_bytes represent data, the format of which is defined privately. These bytes can be used to provide additional information as deemed appropriate.

2.6.62 Metadata STD descriptor

This descriptor defines parameters of the STD model (defined in 2.12.10) for the processing of the metadata stream to which this descriptor is associated.

Table Amd.1-9 – Metadata STD descriptor

Syntax	No. of bits	Mnemonic
Metadata_STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
reserved	**2**	**bslbf**
metadata_input_leak_rate	**22**	**uimsbf**
reserved	**2**	**bslbf**
metadata_buffer_size	**22**	**uimsbf**
reserved	**2**	**bslbf**
metadata_output_leak_rate	**22**	**uimsbf**
}		

2.6.63 Semantic definition of fields in metadata STD descriptor

metadata_input_leak_rate: The metadata_input_leak_rate is a 22-bit field that specifies the leak rate for the associated metadata stream in the T-STD model out of the buffer TB_n into buffer B_n . The leak rate is specified in units of 400 bits/s. For metadata carried in a program stream, the coding of the metadata_input_leak_rate field is not specified, as the rate into B_n equals the rate of the program stream.

metadata_buffer_size: The metadata_buffer_size is a 22-bit field that specifies the size of buffer B_n in the STD model for the associated metadata stream. The size of B_n is specified in units of 1024 bytes.

metadata_output_leak_rate: The metadata_output_leak_rate is a 22-bit field that specifies for the associated metadata service the leak rate in the STD model out of buffer B_n to the decoder. The leak rate is specified in units of 400 bits/s. For metadata streams transported synchronously (stream-type 0x15 or 0x19), the metadata access units are instantaneously removed from B_n under the control of PTS timestamps and in that case the coding of the metadata_output_leak_rate field is not specified .

7) New subclauses after subclause 2.11

Add subclause 2.12 immediately after subclause 2.11:

2.12 Carriage of Metadata

2.12.1 Introduction

An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream can carry metadata. The format of the metadata may be defined by ISO or by any other authority. This subclause defines how to carry the metadata; transport mechanisms are defined as well as metadata related signalling, the applied metadata timing model and extensions of the STD model for decoding of metadata.

A metadata service is defined to be a coherent set of metadata of the same format delivered to a receiver for a specific purpose. Metadata services are contained in metadata streams; each metadata stream carries one or more metadata services. This Specification assumes the notion of metadata Access Units within a metadata service. The definition of a Metadata Access Unit is metadata format specific, but each metadata service is assumed to represent a concatenation (or a collection) of metadata Access Units.

When transporting a metadata service over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, a unique metadata service id is assigned to each such service. A metadata service id references uniquely a metadata service among all the

metadata services available on the same Transport or Program Stream, and *not* unique *solely* within a metadata stream. The metadata service identifier is used to retrieve the metadata service and all the information needed to decode it.

Decoding of metadata may require the availability of decoder configuration data. If a metadata service carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream requires decoder configuration data for decoding, then this metadata decoder configuration data shall be carried within the same program of the same ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream.

Subclause 2.12.2 discusses metadata timing, while subclause 2.12.3 provides an overview of tools that are defined for transport of metadata over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream. The use of available transport tools is specified in 2.12.4 up to 2.12.8, and subclause 2.12.9 specifies metadata related signalling. Finally, the STD model for metadata decoding is specified in 2.12.10.

Since many forms of metadata may be carried, it is essential to signal both the precise format and encoding of the metadata, and the semantic meaning the metadata conveys. The former is signalled by the metadata format, while the latter is signalled by the metadata application format. In other words, the metadata format conveys how the metadata shall be decoded, while the metadata application format conveys how to use the metadata, essentially which application uses the metadata. This division is important since it separates the encoding or representation of the metadata from its meaning, thereby allowing an application to be agnostic of the means by which its metadata is conveyed.

2.12.2 Metadata time-line model

Metadata may refer to time codes associated to the content, for example to indicate the beginning of a content segment. Each time indication made in the metadata refers to a certain metadata content time line specific to the actual metadata format and/or metadata application format. For example, one metadata (application) format may use UTC, while another metadata application format may use SMPTE time codes. To allow for transport of the content at any time over any media, the metadata content time line is expected but not required to be transport agnostic.

When transporting content and the associated metadata over ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams, accurate time references from the metadata to the content are to be maintained. The same is needed if the metadata is delivered over other means. To achieve this, the time line model of Figure Amd.1-1 is assumed in this Specification.

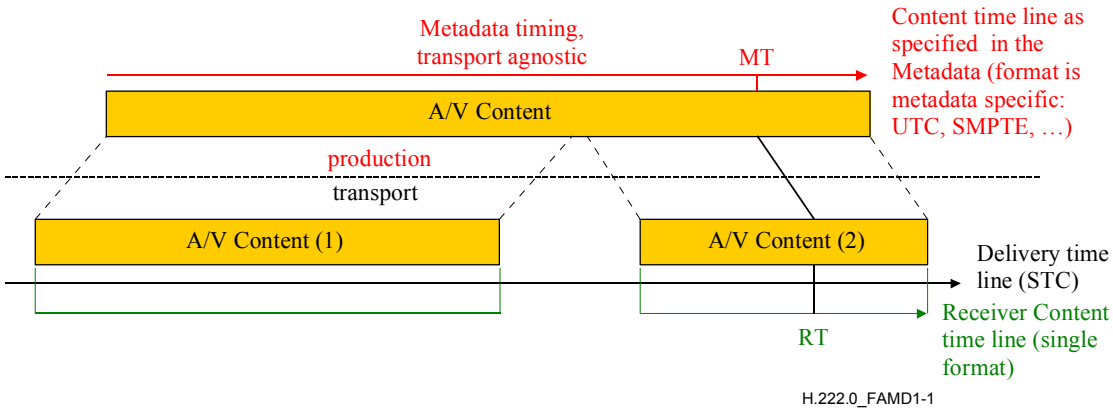

Figure Amd.1-1 – Timing model for delivery of content and metadata

Metadata is associated to the audiovisual content, usually in a transport agnostic way, at production or any other stage prior to transport. Where needed, time information is embedded in the metadata to indicate for example specific segments within the content, using the metadata content time line used in the metadata. For example UTC or SMPTE time codes may be used. The time line format is independent of any time code that may or may not be embedded in the audiovisual stream itself. For example, the metadata time line may utilize UTC, while SMPTE time codes are embedded in the video stream.

The following requirements shall be met for each metadata stream:

- no time discontinuities shall occur in the metadata content time line;
- the metadata content time line shall be locked to the sampling clock of the content;
- each time reference in the metadata stream refers to the same metadata content time line.

At transport, a transport specific timing is associated with the content; this is the delivery time line. In the case of transport over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, the delivery time line is provided by the System Time Clock, the STC. The content may be delivered as a contiguous piece of information, but it is also possible to interrupt

the delivery of the content, for example in the case of news-flash interruptions of a program; in such and other cases time line discontinuities may occur.

When time references are used in the metadata, in the System Target Decoder (STD) these time references are to be associated unambiguously with time values in the received content. To achieve this, a receiver content time line is required. The STC can be used as the receiver content time line, but due to STC discontinuities that may occur, the STC does not necessarily offer an unambiguous time association. Therefore the NPT (Normal Play Time) concept from ISO/IEC 13818-6 DSM-CC is also available for use as the receiver content time line. In any playback mode, such as normal, reverse, slow motion, fast forward, fast backward and still picture, the NPT provides an unambiguous time association, independent of STC discontinuities, and independent of insertions of other content. Note that a new NPT_reference_descriptor needs to be transmitted when the STC rolls over.

To maintain the accurate time references from metadata to the content, information is needed how to map a metadata time, MT, defined on the metadata content time line to the corresponding receiver time, RT, of the receiver content time line. This is achieved by providing the offset in time (in 90 kHz units) between the metadata content time line and the receiver content time line. The offset is provided in the content labelling descriptor. The offset conveys the value of the metadata time base at the instant in time at which the receiver content time base reaches a specified value. See also Figure Amd.1-1.

The timing in metadata systems may refer to a specific picture or audio frame, for example using SMPTE time codes. The offset in time between the metadata content time line and the receiver content time line is expressed in units of 90 kHz, and consequently the metadata time reference will translate into a 90 kHz value in receivers. To accommodate for inaccuracies, receivers shall assume that when reference is made to a picture or audio frame the closest match shall be used. For example, the translated 90 kHz metadata time reference shall be matched with the picture or frame whose PTS value is closest to the translated value.

When using NPT, during playback in any mode at any point in time the offset remains constant between the metadata time base and the NPT time base. As long as neither STC discontinuities nor insertions with other content occur, the same is true for the offset in time between the metadata time base and the STC time base, but only in normal playback mode. For privately defined time lines the offset is also required to be constant, but possibly within constraints not defined in this Specification.

When synchronous transport of metadata is applied in PES packets or by using the synchronized DSM-CC download protocol, PTSs are assigned to the metadata. Such PTS may for example indicate the point in time at which the metadata becomes valid. This implies *a priori* knowledge of how to associate the metadata to the delivery timing. However, synchronously transported metadata may also contain time references, which are to be mapped from the metadata content time line to the receiver content time line using the specified offset between both time lines. See also Figure Amd.1-2.

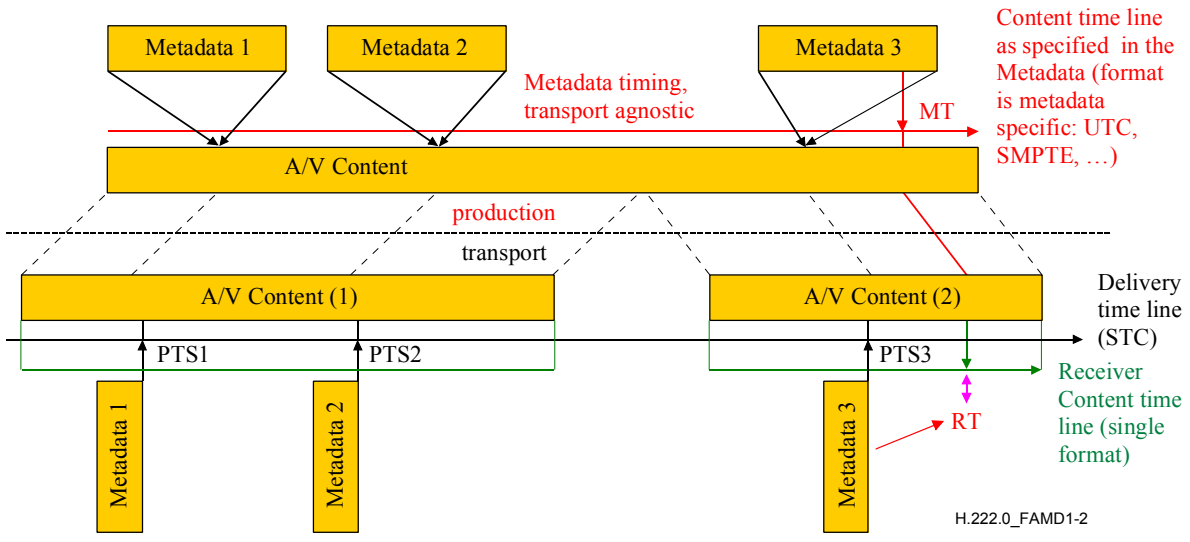

Figure Amd.1-2 – Delivery of metadata in PES packets

2.12.3 Options for Transport of Metadata

To acknowledge the very diverse characteristics of metadata, a variety of tools is defined to transport the metadata over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream.

This Specification defines two tools for synchronous delivery of the metadata:

- carriage in PES packets;
- use of DSM-CC synchronized download protocol.

In addition, this Specification defines three tools for asynchronous delivery of metadata:

- carriage in metadata sections;
- use of DSM-CC data carousels;
- use of DSM-CC object carousels.

Note that some of the asynchronous transport options support carousels and file structures. The choice of transport tool depends on the requirements that apply to the delivery of the metadata, and the requirements of the tools, as described in the following subclauses.

Metadata may also be carried by private means such as PES packets with stream id value 0xBD or 0xBF (private_stream_id_1 or private_stream_id_2) or private sections. This Specification does not specify how to use private means for carriage of metadata, but allows for signalling of such metadata using the descriptors defined in subclause 2.6.56 up to subclause 2.6.63.

The basic referencing of metadata services is the same for all tools, using the metadata service id. However, there are differences per tool. When PES packets, metadata sections, or synchronized DSM-CC download sections are used, data from each metadata service is explicitly signalled within a metadata stream, using the metadata_service_id field. However, when using DSM-CC carousels, this signalling is left at the discretion of metadata applications. Note that this Specification allows for carriage of a metadata service in a DSM-CC carousel, but does not constrain how many metadata services can be carried in one DSM-CC carousel.

Metadata decoder configuration data is signalled explicitly when carried in a metadata desciptor, in PES packets with stream_type 0x15 and stream_id 0xFC, in metadata sections or in synchronized DSM-CC download sections. When metadata decoder configuration data is carried in a DSM-CC carousel, the signalling of such data is required, but not defined by this Specification; instead, such signalling is left at the discretion of applications.

2.12.4 Use of PES packets to transport metadata

PES packets provide a mechanism for synchronous transport of metadata. By means of the PTS in the PES packet header the metadata access units are associated to a certain instant of the STC, without the need for time references in the metadata. This implies *a priori* knowledge of how to associate the metadata to the delivery timing. Specific stream_id and stream_type values are assigned to signal PES packets carrying metadata; see subclause 2.12.9.

When using PES packets with a stream_type of 0x15 and a stream_id of 0xFC to transport the metadata, a Metadata Access Unit Wrapper shall be used as the tool to align PES packets and the metadata Access Units, using metadata_AU_cells. This allows random access indication, whose meaning depends on the format of the metadata, and a cell sequence counter to identify loss of metadata_AU_cells. Each metadata Access Unit is carried and, if appropriate, fragmented in one or more metadata_AU_cells. In each PES packet that carries metadata, the first PES_packet_data_byte shall be the first byte of a Metadata_AU_cell. For each metadata Access Unit contained in the same PES packet, the PTS in the PES header applies. The PTS signals the time at which the metadata Access Units are decoded instantaneously and removed from buffer B_n in the STD. Note that the relationship between a decoded metadata Access Unit and audiovisual content is beyond the scope of this Specification.

A PES packet may contain a single metadata_AU_cell. This is useful if a metadata Access Unit does not fit into a single PES packet, in which case the fragmentation of the metadata Access Unit is handled by the metadata_AU_cell.

When metadata is carried by PES packets in a Program Stream, and if a Program Stream Map is applied in that Program Stream, then the Program Stream Map shall specify which PES packets contain the associated metadata.

2.12.4.1 Metadata Access Unit Wrapper

The metadata Access Unit Wrapper shall be used when carrying metadata Access Units in PES packets with a stream_type of 0x15 and a stream_id value of 0xFC or in synchronized DSM-CC download sections of stream_type 0x19. The wrapper defines a structure consisting of a concatenated number of Metadata_AU_cells. By coding the size of the contained metadata in each metadata_AU_cell, metadata agnostic parsing is possible in receivers: the parser can retrieve the metadata and provide it to a metadata decoder without *a priori* knowledge on any detail of the metadata. The Metadata_AU_cell shall be aligned with the transport; that is the first byte of the payload of the PES packet or synchronized DSM-CC download section shall be the first byte of a Metadata_AU_cell.

If a metadata Access Unit does not fit entirely into a metadata_AU_cell, then the metadata Access Unit shall be fragmented into multiple metadata_AU_cells, where the fragmentation_indication in each such metadata_AU_cell signals that the metadata_AU_cell contains a fragment.

To each Metadata_AU_cell that is contained in the same PES packet or synchronized download section, the PTS as coded in the header of the PES packet or synchronized download section, respectively, applies.

Table Amd.1-10 – Metadata Access Unit Wrapper

Syntax	No. of bits	Mnemonic

Metadata_AU_wrapper () {
for (i=0; i<N;i++){
Metadata_AU_cell ()
}
}		

Table Amd.1-11 – Metadata AU cell

Syntax	No. of bits	Mnemonic
Metadata_AU_cell () {		
metadata_service_id	**8**	**uimsbf**
sequence_number	**8**	**uimsbf**
cell_fragment_indication	**2**	**bslbf**
decoder_config_flag	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
reserved	**4**	**bslbf**
AU_cell_data_length	**16**	**uimsbf**
for (i=0; I<AU_cell_data_length;i++){		
AU_cell_data_byte	**8**	**bslbf**
}		
}		

metadata_service_id: This 8-bit field identifies the metadata service associated to the metadata Access Unit carried in this metadata AU cell.

sequence_number: This 8-bit field specifies the sequence number of the metadata_AU_cell. This number increments by one for each successive metadata_AU_cell constituting the metadata_AU_wrapper, independent of the coded value of the metadata_service_id.

cell_fragment_indication: This 2-bit field conveys information on the metadata Access Unit carried in this metadata_AU_cell, corresponding to Table Amd.1-12.

Table Amd.1-12 – Cell fragment indication

Value	Description
11	A single cell carrying a complete metadata Access Unit.
10	The first cell from a series of cells with data from one metadata Access Unit.
01	The last cell from a series of cells with data from one metadata Access Unit.
00	A cell from a series of cells with data from one metadata Access Unit, but neither the first nor the last one.

random_access_indicator: This 1-bit field, when coded with the value '1', indicates that the metadata carried in this metadata_AU_cell represents an entry point to the metadata service where decoding is possible without information from previous metadata_AU_cells. The meaning of a random access point is defined by the format of the metadata.

decoder_config_flag: This 1-bit field signals the presence of decoder configuration information in the carried metadata Access Unit. Note that this does not preclude the presence of metadata in the Access Unit next to decoder configuration data.

AU_cell_data_length: This 16-bit field specifies the number AU_cell_data_bytes immediately following.

AU_cell_data_byte: This 8-bit field contains contiguous bytes from a metadata Access Unit.

2.12.5 Use of the DSM-CC synchronized download protocol to transport metadata

For synchronized transport, in addition to PES packets, the DSM-CC synchronized download protocol can be used. When using synchronized DSM-CC download sections to transport the metadata, the Metadata Access Unit Wrapper defined in 2.12.4.1 shall be used as the tool to encapsulate metadata Access Units. This allows random access indication, whose meaning depends on the format of the metadata, and a cell sequence counter to identify loss of metadata_AU_cells. In each DSM-CC synchronized download section that carries metadata, the first byte of the payload shall be the first byte of a Metadata_AU_cell. For each metadata Access Unit contained in the same DSM-CC synchronized download section, the PTS in the section header applies. The PTS signals the time at which the metadata Access Units are decoded instantaneously and removed from buffer B_n in the STD. Note that the relationship between a decoded metadata Access Unit and audiovisual content is beyond the scope of this Specification. A specific stream_type value (as detailed in Table 2-29) is assigned to signal carriage of metadata in DSM-CC synchronized download sections.

2.12.6 Use of metadata sections to transport metadata

If asynchronous transport of metadata Access Units without a carousel delivery mechanism is needed, metadata sections can be utilized. The syntax and semantics of metadata sections are defined in subclause 2.12.6. Each metadata section shall carry either one complete metadata Access Unit or a single part of one metadata Access Unit, as signalled by the section_fragment_indication field.

For transport in metadata sections, the metadata Access Units are structured in one or more Metadata Tables. Each Metadata Table contains one or more complete metadata Access Units from one or more metadata services. Conceptually, the transport mechanism of Metadata Tables is comparable to the transport mechanism of Program Map Tables and Program Association Tables. Each Metadata Table may be made up of multiple metadata sections. Each Metadata Table may contain metadata from multiple metadata services.

Specific stream_type and table_id values are assigned to metadata sections. Metadata decoder configuration data can also be carried in sections, signalled by a metadata description value, as assigned by the metadata decoder configuration descriptor.

Table Amd.1-13 – Section syntax for transport of metadata

Syntax	No. of bits	Mnemonic
Metadata_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
decoder_config_flag	**1**	**bslbf**
metadata_section_length	**12**	**uimsbf**
metadata_service_id	**8**	**uimsbf**
reserved	**8**	**bslbf**
section_fragment_indication	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i=1; i<N; i++){		
metadata_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id: The table_id is an 8-bit field that shall be set to '0x06' for each metadata section.

section_syntax_indicator: This 1-bit field shall be set to '1'.

private_indicator: This 1-bit field is not specified by this Specification.

random_access_indicator: This 1-bit field, when coded with the value '1', indicates that the metadata carried in this metadata section represents an access point to the metadata service where decoding is possible without information from previous metadata sections. The meaning of a random access point is defined by the format of the metadata.

decoder_config_flag: This 1-bit field, when coded with the value '1', indicates that decoder configuration information is present in the metadata Access Unit carried in this metadata section.

metadata_section_length: This 12-bit field shall specify the number of remaining bytes in the section immediately following the metadata_section_length field, and including the CRC. The value of this field shall not exceed 4093 (0xFFD).

metadata_service_id: This 8-bit field identifies the metadata service associated to the metadata Access Unit carried in this metadata section. Each Metadata Table may contain metadata from multiple metadata services.

section_fragmentation_indication: This 2-bit field conveys information on the fragmentation of the metadata Access unit carried in this metadata section, corresponding to Table Amd.1-14.

Table Amd.1-14 – Section fragment indication

Value	Description
11	A single metadata section carrying a complete metadata Access Unit.
10	The first metadata section from a series of metadata sections with data from one metadata Access Unit.
01	The last metadata section from a series of metadata sections with data from one metadata Access Unit.
00	A metadata section from a series of metadata sections with data from one metadata Access Unit, but neither the first nor the last one.

version_number: This 5-bit field is the version number of the whole Metadata Table. The version number shall be incremented by 1 modulo 32 whenever the information contained within the Metadata Table changes. When the current_next_indicator is set to '1', then the version_number shall be that of the currently applicable Metadata Table. When the current_next_indicator is set to '0', then the version_number shall be that of the next applicable Metadata Table.

current_next_indicator: A 1-bit field, which when set to '1' indicates that the Metadata Table sent is currently applicable. When the bit is set to '0', it indicates that the Metadata Table sent is not yet applicable and shall be the next Metadata Table to become valid.

section_number: This 8-bit field gives the number of the metadata section. The section_number of the first section in a Metadata Table shall be 0x00. The section_number shall be incremented by 1 with each additional section in this Metadata Table.

last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the complete Metadata Table of which this section is a part.

metadata_byte: This 8-bit contains contiguous bytes from a metadata Access Unit.

CRC_32: This 32-bit field shall contain the CRC value that gives a zero output of the registers in the decoder defined in Annex A of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 after processing the entire metadata_section.

2.12.7 Use of the DSM-CC data carousel to transport metadata

The DSM-CC tools as defined in ISO/IEC 13818-6 for Data Carousels can be used if a carousel delivery mechanism is required without the need to express the hierarchical organization of the metadata structure in the transport mechanism. Information on the carousel in which the metadata is contained, is included in the metadata descriptor defined in sub-subclauses 2.6.60 and 2.6.62. A specific stream_type value is assigned to signal carriage of metadata in the DSM-CC data carousel. Note that signalling of metadata services within a DSM-CC data carousel is required, but not defined by this Specification.

2.12.8 Use of the DSM-CC object carousel to transport metadata

If a carousel delivery mechanism is required with the capability to express the hierarchical organization of the metadata structure in the transport, then the DSM-CC tools and file structures as defined in ISO/IEC 13818-6 for User to User Object Carousels can be used. These file structures provide the tools to structure the metadata as deemed appropriate for efficient parsing of the metadata and for expressing the hierarchical organization of the metadata. Information needed to identify the carousel in which the metadata is contained, is included in the metadata descriptor defined in subclauses 2.6.60 and 2.6.61. This may be the IOP:IOR() as defined in ISO/IEC 13818-6 DSM-CC, subclauses 11.3.1 and 5.7.2.3. A specific stream_type value is assigned to signal carriage of metadata in the DSM-CC object carousel. Note that signalling of metadata services within a DSM-CC object carousel is required, but not defined by this Specification.

2.12.9 Metadata related signalling

Metadata related signaling covers four distinct areas:

- signalling of metadata services and streams;
- signalling of content for use by a metadata system;
- association of metadata to content; and
- signalling of decoder configuration data.

2.12.9.1 Signalling of metadata services and streams

Carriage of metadata is signalled by a stream_type value in the inclusive range between 0x15 and 0x19, specifying which of the five methods described in subclauses 2.12.4 to 2.12.8 is used to transport the metadata, and if appropriate, by a stream_id value of 0xFC indicating a metadata stream.

To uniquely identify a metadata service a metadata_service_id value is assigned to each such service by the transport; the assigned value shall be unique within the Transport or Program Stream carrying the metadata service. If the metadata is carried in PES packets with a stream_id of 0xFC, or in metadata sections, or in ISO/IEC 13818-6 synchronized download sections, the assigned metadata_service_id value is signalled explicitly in the header of the metadata_AU_cell or the metadata section. If a ISO/IEC 13818-6 carousel is used to carry the metadata, then the signalling of metadata services is left to the application. The metadata descriptor specifies the format of the metadata and provides information on the decoder configuration data, and is linked to the metadata service by carrying information on the metadata service it is associated with.

2.12.9.2 Signalling of content for use by a metadata system

In subclauses 2.6.56 and 2.6.57 a content labelling descriptor is defined that can be used to assign a metadata application format specific reference, the content_reference_id_record, to audiovisual or any other content carried over an MPEG-2 Transport Stream or Program Stream. The content_reference_id_record can be used by the metadata system as a label to refer to such content. The content may represent, for example, a program or a stream or segments thereof. The content labelling descriptor also provides information on the content time base used for time referencing from the metadata, including the constant offset in time between the metadata time base and the applied content time base. The descriptor allows carriage of private data. The metadata_application_format may define constraints on the content_reference_record, such as constraints on the time period during which it is valid.

2.12.9.3 Association of metadata to content

In subclauses 2.6.58 and 2.6.59 the metadata pointer descriptor is defined to associate a single metadata service to audiovisual or any other content in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream. The metadata is associated to the content within the context as defined by the location of the descriptor. In a transport stream, the descriptor may be located in the PMT in the descriptor loop for either the program or an elementary stream, but may also be located in tables not defined in this Specification, such as tables describing bouquets of broadcast services.

The metadata pointer descriptor points from the content's context to the metadata service associated to that content. The descriptor provides the value of the metadata_service_id that is assigned to the associated metadata service, as well as one or more locations of the associated metadata. The location may for example be within the same Transport Stream as the content, or within another Transport Stream, but also at a non-ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream location such as the Internet.

2.12.9.4 Signalling decoder configuration data

Decoding of metadata may require the availability of metadata decoder configuration data. If needed, decoder configuration data shall be contained in one of the metadata services in the same program in the same ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream as the metadata service. If decoder configuration data is needed to decode a metadata service, then the metadata descriptor either carries such data or provides the information on retrieval of the decoder configuration data from the same or another metadata service. In a transport stream such other service can be found by searching in the PMT for a metadata_descriptor with the metadata_service_id as specified in the decoder_config_metadata_service_id field (and with the same metadata_format and the same metadata_application format).

2.12.9.5 Overview of metadata signalling

Figure Amd.1-3 provides an example of metadata signalling, in which a single program carries the content (or essence), the "content program", and the metadata is carried in a separate program, the "metadata program". In this example, the metadata program and the content program exist on the same transport stream.

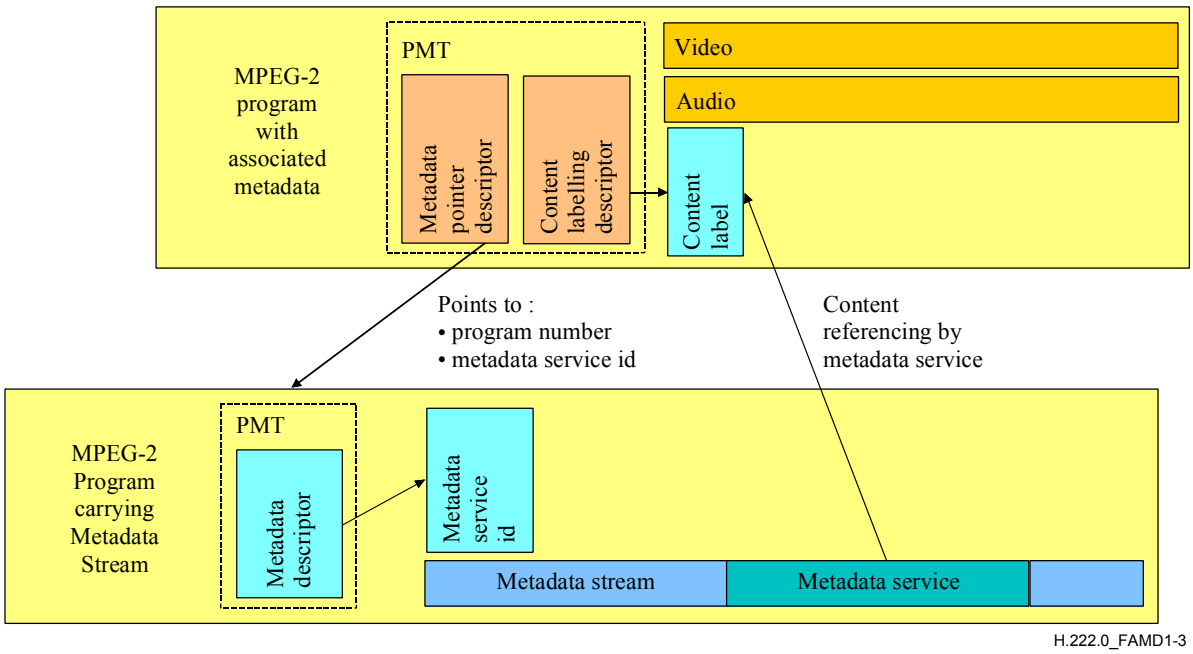

Figure Amd.1-3 – Metadata signalling and referencing

In the content program there are two metadata related descriptors, the content_labeling descriptor and the metadata_pointer descriptor. The content_labeling descriptor associates a label, illustrated in the diagram by "content label" and encoded in the descriptor in the content_reference_id fields, with the content. The label can than be used by the metadata service to refer to the essence, either in whole, in part, or by a time-described segment. For example, the content_labeling descriptor could provide the label "News of 1/1/02", and the metadata could then refer to a specific story item in the "News of 1/1/02", for example by providing the specific timing of the story item.

The metadata pointer descriptor provides information of where the metadata service can be found for the given content. In this example, the metadata is carried in a separate program, but it would be equally valid to have the metadata carried in the same program as the content, or provided by some means beyond the scope of this Specification, for instance from a URL. This descriptor also provides the metadata service id value that is assigned to the metadata service. This is required since a metadata stream could carry multiple metadata services for many different programs and each program needs to be able to uniquely identify its own metadata service.

In the metadata program, the metadata descriptor signals to which metadata service within a metadata stream it applies. If used, the metadata descriptor provides details of where to find the decoder configuration information.

Upon identifying a metadata pointer descriptor in the PMT by a receiver decoding the content program, the receiver retrieves the metadata descriptor from the metadata program. If needed first the decoder configuration data is retrieved, then the decoder is configured accordingly, after which the metadata service can start being decoded.

2.12.10 STD Model for Metadata

The STD model specifies normative constraints on ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams that carry metadata. For decoding of metadata in the STD, the regular T-STD and P-STD models are applicable with buffer B_n, input rate Rx_n of the metadata into B_n and output rate $R_{metadata}$ out of B_n and into $D_{metadata}$, the metadata decoder. See Figure Amd.1-4.

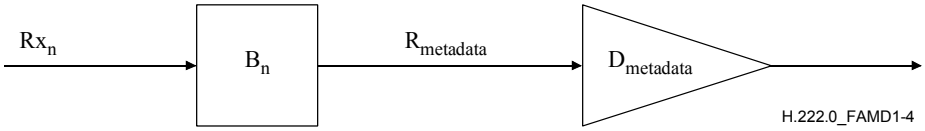

Figure Amd.1-4 – Metadata decoding in the STD

The metadata enters buffer B_n at rate Rx_n. In the P-STD, rate Rx_n equals the rate of the program stream. In the T-STD, rate Rx_n is the rate out of TB_n and equal to the rate defined by the metadata_input_leak_rate field in the metadata STD descriptor. The size BS_n of buffer B_n is equal to the size defined in the metadata_buffer_size field in the metadata STD

descriptor. In case of synchronous delivery, metadata decoding is instantaneous and controlled by PTSs. At decode time, that is when the STC equals the PTS, the associated metadata is removed instantaneously from B_n. In case of asynchronous delivery, the metadata is removed from B_n at a rate $R_{metadata}$ equal to the rate defined by the metadata_output_leak_rate field in the metadata STD descriptor. Buffer B_n shall not overflow.

Note that the STD model defines constraints on the delivery of the metadata, without specifying any constraint on the timing used in the metadata.

2.13 Carriage of ISO 15938 data

2.13.1 Introduction

Carriage of metadata over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 as defined in subclause 2.12 allows for carriage of ISO 15938 data by appropriate coding of the metadata_format field. In this subclause, for the purpose to transport ISO 15938 data, a specific instance is defined. Carriage of ISO 15938 data shall meet each requirement defined in 2.12, but in addition the requirements defined in this subclause shall apply for transport of ISO 15938 data.

2.13.2 ISO 15938 decoder configuration data

Decoding of ISO 15938 data requires the availability of decoder configuration data. Consequently, when ISO 15938 data is carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, then the metadata descriptor shall signal carriage of associated decoder configuration data in the same ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream by coding a value of the decoder_config_flags of either '001' or '010' or '011' or '100'.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Geneva, 2003