
**Information technology — Generic
coding of moving pictures and
associated audio information —**

**Part 1:
Systems**

*Technologies de l'information — Codage générique des images
animées et du son associé —*

Partie 1: Systèmes





COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: 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.

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.

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by ITU-T (as ITU-T H.222.0) and drafted in accordance with its editorial rules, in collaboration with Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This ninth edition cancels and replaces the eighth edition (ISO/IEC 13818-1:2022), which has been technically revised. It also incorporates the Amendment ISO/IEC 13818-1:2022/Amd 1:2023 and the Technical Corrigendum ISO/IEC 13818-1:2022/Cor 1:2023.

The main changes are as follows:

- updates references to ISO/IEC 14496-1 where clause numbering and field naming has changed;
- clarifies a reference to ISO/IEC 23008-3, where the field 3dAudioSceneInfoID is named differently;
- removes semantic definitions for fields that do not exist in the respective syntax table (Table 2-123);
- improves the semantic definition for HDR_WCG_idc equal to '0';
- corrects a mismatch in field size between syntax table and semantic definition of SubstreamOffset[k][j][i];
- corrects a misleading semantic definition of the media_description_flag for the Media_service_kind descriptor.

A list of all parts in the ISO/IEC 13818 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

CONTENTS

	<i>Page</i>
SECTION 1 – GENERAL.....	1
1.1 Scope.....	1
1.2 Normative references	1
SECTION 2 – TECHNICAL ELEMENTS	4
2.1 Definitions.....	4
2.2 Symbols and abbreviations	12
2.3 Method of describing bit stream syntax	14
2.4 Transport stream bitstream requirements	15
2.5 Program stream bitstream requirements.....	63
2.6 Program and program element descriptors.....	77
2.7 Restrictions on the multiplexed stream semantics	166
2.8 Compatibility with ISO/IEC 11172	170
2.9 Registration of copyright identifiers	170
2.10 Registration of private data format.....	171
2.11 Carriage of ISO/IEC 14496 data	171
2.12 Carriage of metadata	183
2.13 Carriage of ISO 15938 data.....	192
2.14 Carriage of Rec. ITU-T H.264 ISO/IEC 14496-10 video	192
2.15 Carriage of ISO/IEC 14496-17 text streams	208
2.16 Carriage of auxiliary video streams	210
2.17 Carriage of HEVC.....	210
2.18 Carriage of green access units.....	224
2.19 Carriage of ISO/IEC 23008-3 MPEG-H 3D audio data.....	226
2.20 Carriage of Quality Access Units in MPEG-2 sections.....	228
2.21 Carriage of sample variants.....	229
2.22 Carriage of Media Orchestration Access Units.....	230
2.23 Carriage of VVC	230
2.24 Carriage of EVC	235
2.25 Carriage of LCEVC	238
Annex A CRC decoder model	240
A.1 CRC decoder model.....	240
Annex B Digital storage medium command and control (DSM-CC).....	241
B.1 Introduction.....	241
B.2 General elements.....	242
B.3 Technical elements.....	244
Annex C Program-specific information.....	250
C.1 Explanation of program-specific information in transport streams.....	250
C.2 Introduction.....	250
C.3 Functional mechanism	250
C.4 The mapping of sections into transport stream packets.....	251
C.5 Repetition rates and random access	251
C.6 What is a program?	252
C.7 Allocation of program_number	252
C.8 Usage of PSI in a typical system.....	252
C.9 The relationships of PSI structures	253
C.10 Bandwidth utilization and signal acquisition time	255
Annex D Systems timing model and application implications of this Recommendation International Standard.....	258
D.1 Introduction.....	258
Annex E Data transmission applications	267
E.1 General considerations.....	267
E.2 Suggestion.....	267
Annex F Graphics of syntax for this Recommendation International Standard.....	268

	<i>Page</i>
F.1 Introduction.....	268
Annex G General information	272
G.1 General information	272
Annex H Private data.....	273
H.1 Private data.....	273
Annex I Systems conformance and real-time interface	274
I.1 Systems conformance and real-time interface.....	274
Annex J Interfacing jitter-inducing networks to MPEG-2 decoders.....	275
J.1 Introduction.....	275
J.2 Network compliance models.....	275
J.3 Network specification for jitter smoothing	276
J.4 Example decoder implementations	277
Annex K Splicing transport streams	278
K.1 Introduction.....	278
K.2 The different types of splicing point	278
K.3 Decoder behaviour on splices	279
Annex L Registration procedure (see 2.9).....	281
L.1 Procedure for the request of a Registered Identifier (RID).....	281
L.2 Responsibilities of the Registration Authority.....	281
L.3 Responsibilities of parties requesting an RID	281
L.4 Appeal procedure for denied applications.....	281
Annex M Registration application form (see 2.9)	283
M.1 Contact information of organization requesting a Registered Identifier (RID).....	283
M.2 Statement of an intention to apply the assigned RID	283
M.3 Date of intended implementation of the RID	283
M.4 Authorized representative	283
M.5 For official use only of the Registration Authority	283
Annex N Registration Authority Diagram of administration structure (see 2.9).....	284
Annex O Registration procedure (see 2.10).....	285
O.1 Procedure for the request of an RID.....	285
O.2 Responsibilities of the Registration Authority	285
O.3 Contact information for the Registration Authority	285
O.4 Responsibilities of parties requesting an RID	285
O.5 Appeal procedure for denied applications.....	285
Annex P Registration application form	287
P.1 Contact information of organization requesting an RID	287
P.2 Request for a specific RID	287
P.3 Short description of RID that is in use and date system that was implemented	287
P.4 Statement of an intention to apply the assigned RID	287
P.5 Date of intended implementation of the RID	287
P.6 Authorized representative	287
P.7 For official use of the Registration Authority	287
Annex Q T-STD and P-STD buffer models for ISO/IEC 13818-7 ADTS	288
Q.1 Introduction.....	288
Q.2 Leak rate from transport buffer	288
Q.3 Buffer size.....	288
Q.4 Conclusion	289
Annex R Carriage of ISO/IEC 14496 scenes in Rec. ITU-T H.222.0 ISO/IEC 13818-1	291
R.1 Content access procedure for ISO/IEC 14496 program components within a program stream	291
R.2 Content access procedure for ISO/IEC 14496 program components within a transport stream	292
Annex S Carriage of JPEG 2000 part 1 video over MPEG-2 transport streams.....	296
S.1 Introduction.....	296

	<i>Page</i>
S.2 J2K video access unit, J2K video elementary stream, J2K video sequence and J2K still picture.....	296
S.3 Optional J2K block mode for high resolution support.....	296
S.4 Optional J2K stripe mode for Ultra-Low Latency	297
S.5 Elementary stream header (elsm) and mapping to PES packets	297
S.6 J2K transport constraints.....	300
S.7 Interpretation of flags in adaptation and PES headers for J2K video elementary streams.....	301
S.8 T-STD extension for J2K video elementary streams	301
Annex T MIME type for MPEG-2 transport streams	304
T.1 Introduction.....	304
T.2 MIME type and subtype.....	304
T.3 Security considerations	305
T.4 Parameters.....	305
Annex U Carriage of timeline and external media information over MPEG-2 transport streams	307
U.1 Introduction.....	307
U.2 TEMI access unit and TEMI elementary stream.....	308
U.3 AF descriptors	309
Annex V Transport of HEVC tiles	318
V.1 Introduction.....	318
V.2 HEVC tile substream identification example.....	319
V.3 Subregion layout example.....	319
Annex W Carriage of JPEG XS part 1 video over MPEG-2 Transport Streams	321
W.1 Introduction.....	321
W.2 JPEG XS video access unit, JPEG XS video elementary stream, JPEG XS video sequence and JPEG XS still picture	321
W.3 Elementary stream header (jxes) and mapping to PES packets.....	321
W.4 JPEG XS transport constraints	322
W.5 Interpretation of flags in adaptation field and PES packet for JPEG XS video elementary streams.....	323
W.6 T-STD extension for JPEG XS video elementary streams.....	323
Bibliography	326

List of Tables

	<i>Page</i>
Table 2-1 – Transport stream.....	26
Table 2-2 – Transport packet of this Recommendation International Standard.....	26
Table 2-3 – PID table	27
Table 2-4 – Scrambling control values.....	27
Table 2-5 – Adaptation field control values	28
Table 2-6 – Transport stream adaptation field.....	28
Table 2-7 – Splice parameters Table 1 Simple Profile Main Level, Main Profile Main Level, SNR Profile Main Level (both layers), Spatial Profile High-1440 Level (base layer), High Profile Main Level (middle + base layers), Multi-view Profile Main Level (base layer) Video	35
Table 2-8 – Splice parameters Table 2 Main Profile Low Level, SNR Profile Low Level (both layers), High Profile Main Level (base layer), Multi-view Profile Low Level (base layer) Video.....	35
Table 2-9 – Splice parameters Table 3 Main Profile High-1440 Level, Spatial Profile High-1440 Level (all layers), High Profile High-1440 Level (middle + base layers), Multi-view Profile High-1440 Level (base layer) Video	36
Table 2-10 – Splice parameters Table 4 Main Profile High Level, High Profile High-1440 Level (all layers), High Profile High Level (middle + base layers), Multi-view Profile High Level (base layer) Video.....	36
Table 2-11 – Splice parameters Table 5 SNR Profile Low Level (base layer) Video	36
Table 2-12 – Splice parameters Table 6 SNR Profile Main Level (base layer) Video	36
Table 2-13 – Splice parameters Table 7 Spatial Profile High-1440 Level (middle + base layers) Video	37
Table 2-14 – Splice parameters Table 8 High Profile Main Level (all layers), High Profile High-1440 Level (base layer) Video	37
Table 2-15 – Splice parameters Table 9 High Profile High Level (base layer), Multi-view Profile Main Level (both layers) Video	37
Table 2-16 – Splice parameters Table 10 High Profile High Level (all layers), Multi-view Profile High-1440 Level (both layers) Video	38
Table 2-17 – Splice parameters Table 11 4:2:2 Profile Main Level Video	38
Table 2-18 – Splice parameters Table 12 Multi-view Profile Low Level (both layers) Video.....	38
Table 2-19 – Splice parameters Table 13 Multi-view Profile High Level (both layers) Video.....	38
Table 2-20 – Splice parameters Table 14 4:2:2 Profile High Level Video.....	39
Table 2-21 – PES packet	39
Table 2-22 – Stream_id assignments.....	42
Table 2-23 – PES scrambling control values.....	43
Table 2-24 – Trick mode control values.....	48
Table 2-25 – Field_id field control values.....	49
Table 2-26 – Coefficient selection values.....	49
Table 2-27 – Stream_id_extension assignments.....	51
Table 2-28 – Program-specific information.....	52
Table 2-29 – Program-specific information pointer	54
Table 2-30 – Program association section	54
Table 2-31 – table_id assignment values.....	55
Table 2-32 – Conditional access section.....	56
Table 2-33 – Transport stream program map section	57
Table 2-34 – Stream type assignments	58
Table 2-35 – Private section	61
Table 2-36 – The transport stream description table	62
Table 2-37 – Program stream	68
Table 2-38 – Program stream pack.....	68
Table 2-39 – Program stream pack header	69
Table 2-40 – Program stream system header.....	70
Table 2-41 – Program stream map.....	73
Table 2-42 – Program stream directory packet.....	75
Table 2-43 – Intra_coded indicator.....	76
Table 2-44 – Coding_parameters indicator	77
Table 2-45 – Program and program element descriptors.....	77
Table 2-46 – Video stream descriptor	79

	<i>Page</i>
Table 2-47 – Frame rate code	79
Table 2-48 – Audio stream descriptor	80
Table 2-49 – Hierarchy descriptor	80
Table 2-50 – Hierarchy_type field values	82
Table 2-51 – Registration descriptor	82
Table 2-52 – Data stream alignment descriptor	83
Table 2-53 – Video stream alignment values	83
Table 2-54 – AVC video stream alignment values	84
Table 2-55 – HEVC video stream alignment values	84
Table 2-56 – Audio stream alignment values	84
Table 2-57 – VVC video stream alignment values	85
Table 2-58 – EVC video stream alignment values	85
Table 2-59 – Target background grid descriptor	86
Table 2-60 – Video window descriptor	86
Table 2-61 – Conditional access descriptor	87
Table 2-62 – ISO 639 language descriptor	87
Table 2-63 – Audio type values	88
Table 2-64 – System clock descriptor	89
Table 2-65 – Multiplex buffer utilization descriptor	89
Table 2-66 – Copyright descriptor	90
Table 2-67 – Maximum bitrate descriptor	90
Table 2-68 – Private data indicator descriptor	91
Table 2-69 – Smoothing buffer descriptor	91
Table 2-70 – STD descriptor	92
Table 2-71 – IBP descriptor	92
Table 2-72 – MPEG-4 video descriptor	93
Table 2-73 – MPEG-4 audio descriptor	93
Table 2-75 – IOD descriptor	96
Table 2-76 – SL descriptor	96
Table 2-77 – FMC descriptor	97
Table 2-78 – External_ES_ID descriptor	97
Table 2-79 – Muxcode descriptor	98
Table 2-80 – M4MuxBufferSize descriptor	98
Table 2-81 – MultiplexBuffer descriptor	99
Table 2-82 – M4MuxTiming descriptor	99
Table 2-83 – Content labelling descriptor	100
Table 2-84 – Metadata_application_format	100
Table 2-85 – Content_time_base_indicator values	101
Table 2-86 – Metadata pointer descriptor	102
Table 2-87 – Metadata format values	102
Table 2-88 – MPEG_carriage_flags	103
Table 2-89 – Metadata descriptor	104
Table 2-90 – decoder_config_flags	105
Table 2-91 – Metadata STD descriptor	106
Table 2-92 – AVC video descriptor	106
Table 2-93 – AVC timing and HRD descriptor	108
Table 2-94 – MPEG-2 AAC_audio_descriptor	109
Table 2-95 – MPEG-2_AAC_additional_information field values	110
Table 2-96 – MPEG-4 text descriptor	110
Table 2-97 – MPEG-4 audio extension descriptor	110
Table 2-98 – Auxiliary video stream descriptor	111
Table 2-99 – SVC extension descriptor	112
Table 2-100 – MVC extension descriptor	113
Table 2-101 – J2K video descriptor	114

	<i>Page</i>
Table 2-102 – Example frame rates based on DEN_frame_rate and NUM_frame_rate values	116
Table 2-103 – MVC operation point descriptor.....	118
Table 2-104 – MPEG2_stereoscopic_video_format_descriptor syntax	119
Table 2-105 – Stereoscopic_program_info_descriptor syntax	119
Table 2-106 – Stereoscopic_service_type values	120
Table 2-107 – Stereoscopic_video_info_descriptor syntax	120
Table 2-108 – Upsampling factor values.....	121
Table 2-109 – Extension descriptor.....	121
Table 2-110 – Extension descriptor tag values.....	124
Table 2-111 – Transport_profile_descriptor syntax	125
Table 2-112 – Transport_profile values	125
Table 2-113 – HEVC video descriptor	126
Table 2-114 – Semantics of HDR_WGC_idc.....	128
Table 2-115 – HEVC timing and HRD descriptor.....	128
Table 2-116 – Adaptation field extension descriptor.....	129
Table 2-117 – HEVC operation point descriptor.....	130
Table 2-118 – HEVC hierarchy extension descriptor.....	132
Table 2-119 – Semantics of extension dimension bits.....	132
Table 2-120 – Green extension descriptor.....	133
Table 2-121 – MPEG-H 3D audio descriptor.....	134
Table 2-122 – MPEG-H 3D audio config descriptor.....	134
Table 2-123 – MPEG-H 3D audio scene descriptor	135
Table 2-124 – MPEG-H 3D audio text label descriptor	138
Table 2-125 – MPEG-H 3D audio multi-stream descriptor.....	140
Table 2-126 – MPEG-H 3D audio DRC and Loudness descriptor()	141
Table 2-127 – MPEG-H 3D audio command descriptor	143
Table 2-128 – Quality extension descriptor.....	144
Table 2-129 – Virtual segmentation descriptor	145
Table 2-130 – HEVC tile substream descriptor.....	146
Table 2-131 – HEVC subregion descriptor	147
Table 2-132 – JPEG XS video descriptor.....	149
Table 2-133 – VVC video descriptor.....	151
Table 2-134 – Semantics of HDR_WGC_idc.....	152
Table 2-135 – SDR widely used video property combinations	153
Table 2-136 – WCG widely used video property combinations.....	153
Table 2-137 – HDR/WCG widely used video property combinations	153
Table 2-138 – No Indication.....	154
Table 2-139 – VVC timing and HRD descriptor.....	154
Table 2-140 – EVC video descriptor	156
Table 2-141 – EVC timing and HRD descriptor	157
Table 2-142 – LCEVC video descriptor.....	158
Table 2-143 – LCEVC linkage descriptor	159
Table 2-144 – Media service kind descriptor	160
Table 2-145 – media_description_flag	160
Table 2-146 – Media type indicator.....	160
Table 2-147 – ID_length_code	161
Table 2-148 – ID_type.....	161
Table 2-149 – configuration type values	161
Table 2-150 – lang_len_indicator	162
Table 2-151 – Media service type values	162
Table 2-152 – Carriage of individual ISO/IEC 14496 streams in Rec. ITU-T H.222.0 ISO/IEC 13818-1	172
Table 2-153 – Section syntax for transport of ISO/IEC 14496 stream.....	177
Table 2-154 – ISO/IEC defined options for carriage of an ISO/IEC 14496 scene and associated streams in Rec. ITU-T H.222.0 ISO/IEC 13818-1	180
Table 2-155 – Metadata Access Unit Wrapper.....	187

	<i>Page</i>
Table 2-156 – Metadata AU cell	187
Table 2-157 – Cell fragment indication.....	187
Table 2-158 – Section syntax for transport of metadata	188
Table 2-159 – Section fragment indication.....	189
Table 2-160 – View and dependency representation delimiter NAL unit	198
Table 2-161 – Implied hierarchy_layer_index if no hierarchy descriptors are used.....	218
Table 2-162 – Green access unit section syntax	225
Table 2-163 – Green access unit.....	225
Table 2-164 – Quality Access Unit	228
Table B.1 – DSM-CC syntax.....	245
Table B.2 – Command_id assigned values	245
Table B.3 – DSM-CC control.....	246
Table B.4 – Select mode assigned values	247
Table B.5 – DSM-CC Acknowledgement	248
Table B.6 – Time code	249
Table C.1 – Composite_descriptor	255
Table C.2 – Sub-descriptor.....	255
Table C.3 – Program association table bandwidth usage (bit/s) Number of programs per transport stream	256
Table C.4 – Program map table bandwidth usage (bit/s) Number of programs per transport stream.....	256
Table D.1 – Re-multiplexing strategy	263
Table E.1 – PES packet header example	267
Table S.1 – J2K Access unit elementary stream header	298
Table S.2 – Operating levels and maximum buffer size for JPEG 2000 broadcast profiles (from Table A.49 in Rec. ITU-T T.800 (2015) ISO/IEC 15444-1:2016)	303
Table T.1 – 'codecs' parameter values for some specific stream_type values	305
Table U.1 – Variable field length notation example.....	307
Table U.1bis – Table U.1 in equivalent full notation	308
Table U.2 – TEMI access unit	308
Table U.3 – AF descriptor tags.....	309
Table U.4 – TEMI location descriptor.....	310
Table U.5 – TEMI URL scheme types	310
Table U.6 – TEMI service types.....	311
Table U.7 – TEMI base URL descriptor.....	311
Table U.8 – TEMI timeline descriptor	312
Table U.9 – TEMI MPEG-H_3dAudio_extStreamID descriptor	314
Table U.10 – Boundary descriptor.....	315
Table U.11 – sequence_number_length_code interpretation.....	315
Table U.12 – Labelling Descriptor	316
Table U.13 – label_length_code interpretation.....	316
Table U.14 – label_type values	317
Table U.15 – HEVC tile substream af_descriptor	317
Table W.1 – JPEG XS Access unit elementary stream header (jxes header)	321

List of Figures

	<i>Page</i>
Figure Intro. 1 – Simplified overview of the scope of this Recommendation International Standard	xiv
Figure Intro. 2 – Prototypical transport demultiplexing and decoding example.....	xvi
Figure Intro. 3 – Prototypical transport multiplexing example.....	xvi
Figure Intro. 4 – Prototypical transport stream to program stream conversion	xvi
Figure Intro. 5 – Prototypical decoder for program streams.....	xvii
Figure 2-1 – Transport stream system target decoder notation.....	16
Figure 2-2 – Program stream system target decoder notation	63
Figure 2-3 – Target background grid descriptor display area.....	85
Figure 2-4 – Media Service Kind Descriptor semantics at program level.....	165
Figure 2-5 – Media Program Kind Descriptor semantics at elementary stream level	165
Figure 2-6 – T-STD model extensions for individual ISO/IEC 14496 elementary streams	172
Figure 2-7 – T-STD model for ISO/IEC 14496 content	178
Figure 2-8 – P-STD model for ISO/IEC 14496 Systems stream	181
Figure 2-9 – Timing model for delivery of content and metadata	184
Figure 2-10 – Delivery of metadata in PES packets	185
Figure 2-11 – Metadata signalling and referencing	191
Figure 2-12 – Metadata decoding in the STD.....	191
Figure 2-13 – T-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 video.....	195
Figure 2-14 – P-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 video	197
Figure 2-15 – T-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 Video with scalable video sub-bitstreams.....	198
Figure 2-16 – P-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 Video with scalable video sub-bitstreams.....	201
Figure 2-17 – T-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 Video with MVC video sub-bitstreams.....	203
Figure 2-18 – P-STD model extensions for Rec. ITU-T H.264 ISO/IEC 14496-10 Video with MVC video sub-bitstreams.....	207
Figure 2-19 – T-STD model extensions for ISO/IEC 14496-17 text streams.....	209
Figure 2-20 – T-STD model extensions for single layer HEVC.....	212
Figure 2-21 – T-STD model extensions for layered transport of HEVC temporal video subsets.....	214
Figure 2-22 – T-STD model extensions for bitstream-partition-specific CPB operation	216
Figure 2-23 – T-STD model extensions for transport of HEVC tiles through individual ESs	220
Figure 2-24 – T-STD model extensions for transport of HEVC tiles in a common ES using AF descriptors.....	222
Figure 2-25 – T-STD model extension for transport of HEVC tiles in a common ES ignoring AF descriptors	224
Figure 2-26 – T-STD model extension for green access units.....	226
Figure 2-27 – Transport stream system target decoder for multiple audio elementary streams	228
Figure 2-28 – Quality Access Unit decoder processing model.....	229
Figure 2-29 – T-STD model extensions for single layer VVC.....	231
Figure 2-30 – T-STD model extensions for layered transport of VVC temporal video subsets.....	233
Figure 2-31 – T-STD model extensions for single layer EVC	236
Figure A.1 – 32-bit CRC decoder model.....	240
Figure B.1 – Configuration of DSM-CC application.....	243
Figure B.2 – BSM-CC bitstream decoded as a stand-alone bitstream	243
Figure B.3 – DSM-CC bitstream decoded as part of the system bitstream	244
Figure C.1 – Program and network mapping relationships	253
Figure D.1 – Constant delay model	258
Figure D.2 – STC recovery using PLL	262
Figure F.1 – Transport stream syntax diagram	268
Figure F.2 – PES packet syntax diagram.....	269
Figure F.3 – Program association section diagram	269
Figure F.4 – Conditional access section diagram	269
Figure F.5 – TS program map section diagram	270
Figure F.6 – Private section diagram	270
Figure F.7 – Program stream diagram	271

	<i>Page</i>
Figure F.8 – Program stream map diagram	271
Figure J.1 – Sending system streams over a jitter-inducing network	276
Figure J.2 – Jitter-smoothing using network-layer timestamps	276
Figure J.3 – Integrated dejittering and MPEG-2 decoding	277
Figure R.1 – Example of ISO/IEC 14496 content in a program stream	292
Figure R.2 – Example of ISO/IEC 14496 content in a transport stream.....	293
Figure R.3 – Usage of MPEG-4 in a transport stream with BIFS scene referring to native PES	294
Figure R.4 – Usage of MPEG-4 in a transport stream with an ODUpdate_descriptor carrying an image ObjectDescriptor in the PMT	295
Figure S.1 – Structure and order of JPEG 2000 access units.....	300
Figure S.2 – T-STD model extensions for J2K Video.....	301
Figure U.1 – Stream partitioning into 2 and 5 second segments	315
Figure V.1 – Illustration of HEVC tiled encoding of panoramic content beyond UHD.....	318
Figure V.2 – Example of HEVC tile substream identification	319
Figure V.3 – Example of subregion layout for a 3 x 3 RoI	319
Figure W.1 – Structure and order of JPEG XS access units	322
Figure W.2 – T-STD model extensions for JPEG XS Video.....	324

Introduction

The systems part of this Recommendation | International Standard addresses the combining of one or more elementary streams of video and audio, as well as other data, into single or multiple streams which are suitable for storage or transmission. Systems coding follows the syntactical and semantic rules imposed by this Specification and provides information to enable synchronized decoding of decoder buffers over a wide range of retrieval or receipt conditions.

System coding shall be specified in two forms: the transport stream and the program stream. Each is optimized for a different set of applications. Both the transport stream and program stream defined in this Recommendation | International Standard provide coding syntax which is necessary and sufficient to synchronize the decoding and presentation of the video and audio information, while ensuring that data buffers in the decoders do not overflow or underflow. Information is coded in the syntax using time stamps concerning the decoding and presentation of coded audio and visual data and time stamps concerning the delivery of the data stream itself. Both stream definitions are packet-oriented multiplexes.

The basic multiplexing approach for single video and audio elementary streams is illustrated in Figure Intro. 1. The video and audio data is encoded as described in Rec. ITU-T H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3. The resulting compressed elementary streams are packetized to produce PES packets. Information needed to use PES packets independently of either transport streams or program streams may be added when PES packets are formed. This information is not needed and need not be added when PES packets are further combined with system level information to form transport streams or program streams. This systems standard covers those processes to the right of the vertical dashed line.

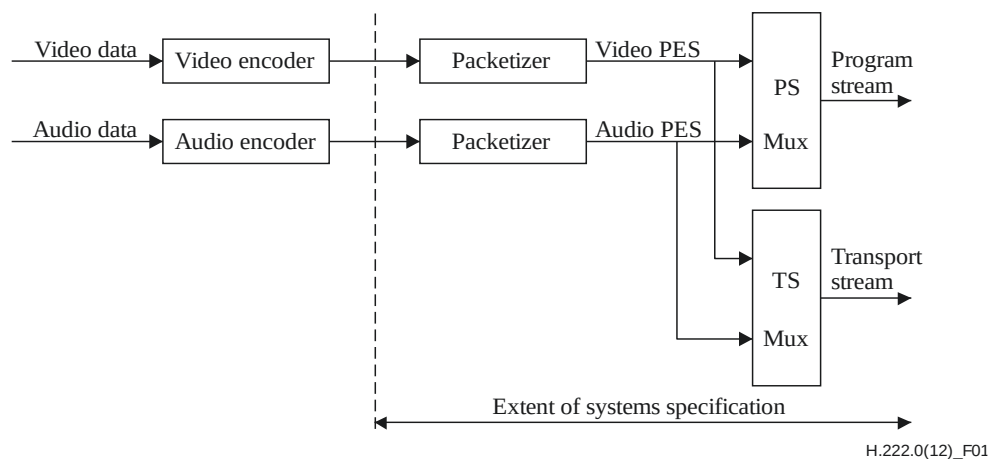


Figure Intro. 1 – Simplified overview of the scope of this Recommendation | International Standard

The program stream is analogous and similar to the ISO/IEC 11172 systems layer. It results from combining one or more streams of PES packets, which have a common time base, into a single stream.

For applications that require the elementary streams that comprise a single program to be in separate streams that are not multiplexed, the elementary streams can also be encoded as separate program streams, one per elementary stream, with a common time base. In this case the values encoded in the SCR fields of the various streams shall be consistent.

Like the single program stream, all elementary streams can be decoded with synchronization.

The program stream is designed for use in relatively error-free environments and is suitable for applications which may involve software processing of system information such as interactive multi-media applications. Program stream packets may be of variable and relatively great length.

The transport stream combines one or more programs with one or more independent time bases into a single stream. PES packets made up of elementary streams that form a program share a common time base. The transport stream is designed for use in environments where errors are likely, such as storage or transmission in lossy or noisy media. Transport stream packets are 188 bytes in length.

Program and transport streams are designed for different applications and their definitions do not strictly follow a layered model. It is possible and reasonable to convert from one to the other; however, one is not a subset or superset of the other. In particular, extracting the contents of a program from a transport stream and creating a valid program stream is possible and is accomplished through the common interchange format of PES packets, but not all of the fields needed in a program stream are contained within the transport stream; some must be derived. The transport stream may be used to span a range of layers in a layered model, and is designed for efficiency and ease of implementation in high bandwidth applications.

The scope of syntactical and semantic rules set forth in the systems specification differs: the syntactical rules apply to systems layer coding only, and do not extend to the compression layer coding of the video and audio specifications; by contrast, the semantic rules apply to the combined stream in its entirety.

The systems specification does not specify the architecture or implementation of encoders or decoders, nor those of multiplexors or demultiplexors. However, bit stream properties do impose functional and performance requirements on encoders, decoders, multiplexors and demultiplexors. For instance, encoders must meet minimum clock tolerance requirements. Notwithstanding this and other requirements, a considerable degree of freedom exists in the design and implementation of encoders, decoders, multiplexors, and demultiplexors.

Intro. 1 Transport stream

The transport stream is a stream definition which is tailored for communicating or storing one or more programs of coded data according to Rec. ITU-T H.262 | ISO/IEC 13818-2 and ISO/IEC 13818-3 and other data in environments in which significant errors may occur. Such errors may be manifested as bit value errors or loss of packets.

Transport streams may be either fixed or variable rate. In either case the constituent elementary streams may either be fixed or variable rate. The syntax and semantic constraints on the stream are identical in each of these cases. The transport stream rate is defined by the values and locations of program clock reference (PCR) fields, which in general are separate PCR fields for each program.

There are some difficulties with constructing and delivering a transport stream containing multiple programs with independent time bases such that the overall bit rate is variable. Refer to 2.4.2.3.

The transport stream may be constructed by any method that results in a valid stream. It is possible to construct transport streams containing one or more programs from elementary coded data streams, from program streams, or from other transport streams which may themselves contain one or more programs.

The transport stream is designed in such a way that several operations on a transport stream are possible with minimum effort. Among these are:

- 1) Retrieve the coded data from one program within the transport stream, decode it and present the decoded results as shown in Figure Intro. 2.
- 2) Extract the transport stream packets from one program within the transport stream and produce as output a different transport stream with only that one program as shown in Figure Intro. 3.
- 3) Extract the transport stream packets of one or more programs from one or more transport streams and produce as output a different transport stream (not illustrated).
- 4) Extract the contents of one program from the transport stream and produce as output a program stream containing that one program as shown in Figure Intro. 4.
- 5) Take a program stream, convert it into a transport stream to carry it over a lossy environment, and then recover a valid, and in certain cases, identical program stream.

Figure Intro. 2 and Figure Intro. 3 illustrate prototypical demultiplexing and decoding systems which take as input a transport stream. Figure Intro. 2 illustrates the first case, where a transport stream is directly demultiplexed and decoded. Transport streams are constructed in two layers:

- a system layer; and
- a compression layer.

The input stream to the transport stream decoder has a system layer wrapped about a compression layer. Input streams to the video and audio decoders have only the compression layer.

Operations performed by the prototypical decoder which accepts transport streams either apply to the entire transport stream ("multiplex-wide operations"), or to individual elementary streams ("stream-specific operations"). The transport stream system layer is divided into two sub-layers, one for multiplex-wide operations (the transport stream packet layer), and one for stream-specific operations (the PES packet layer).

A prototypical decoder for transport streams, including audio and video, is also depicted in Figure Intro. 2 to illustrate the function of a decoder. The architecture is not unique – some system decoder functions, such as decoder timing control, might equally well be distributed among elementary stream decoders and the channel-specific decoder – but this figure is useful for discussion. Likewise, indication of errors detected by the channel-specific decoder to the individual audio and video decoders may be performed in various ways and such communication paths are not shown in the diagram. The prototypical decoder design does not imply any normative requirement for the design of a transport stream decoder. Indeed non-audio/video data is also allowed, but not shown.

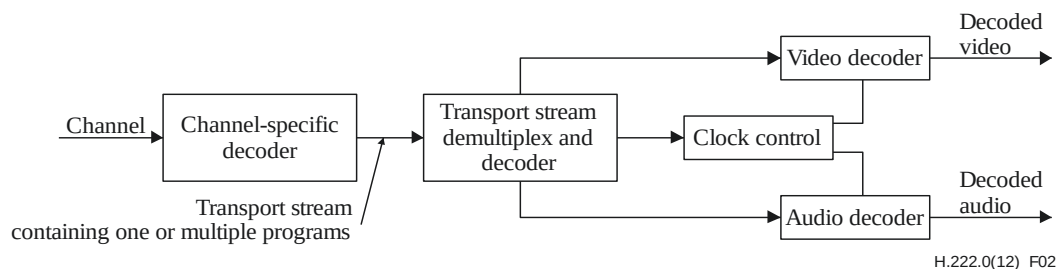


Figure Intro. 2 – Prototypical transport demultiplexing and decoding example

Figure Intro. 3 illustrates the second case, where a transport stream containing multiple programs is converted into a transport stream containing a single program. In this case the re-multiplexing operation may necessitate the correction of program clock reference (PCR) values to account for changes in the PCR locations in the bit stream.

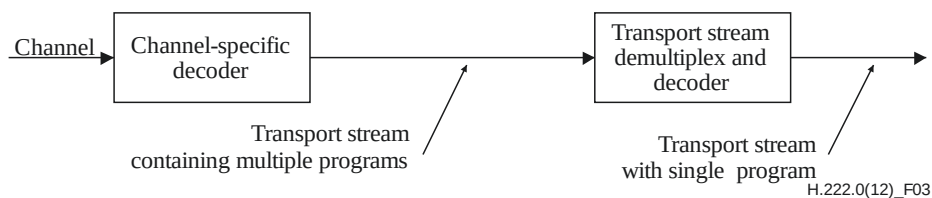


Figure Intro. 3 – Prototypical transport multiplexing example

Figure Intro. 4 illustrates a case in which a multi-program transport stream is first demultiplexed and then converted into a program stream.

Figures Intro. 3 and Intro. 4 indicate that it is possible and reasonable to convert between different types and configurations of transport streams. There are specific fields defined in the transport stream and program stream syntax which facilitate the conversions illustrated. There is no requirement that specific implementations of demultiplexors or decoders include all of these functions.

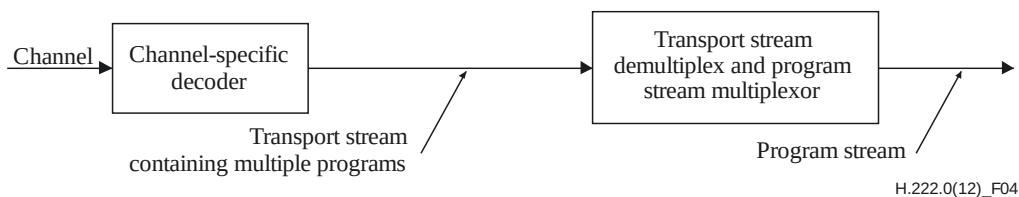


Figure Intro. 4 – Prototypical transport stream to program stream conversion

Intro. 2 Program stream

The program stream is a stream definition which is tailored for communicating or storing one program of coded data and other data in environments where errors are very unlikely, and where processing of system coding, e.g., by software, is a major consideration.

Program streams may be either fixed or variable rate. In either case, the constituent elementary streams may be either fixed or variable rate. The syntax and semantics constraints on the stream are identical in each case. The program stream rate is defined by the values and locations of the system clock reference (SCR) and mux_rate fields.

A prototypical audio/video program stream decoder system is depicted in Figure Intro. 5. The architecture is not unique – system decoder functions including decoder timing control might as equally well be distributed among elementary stream decoders and the channel-specific decoder – but this figure is useful for discussion. The prototypical decoder design does not imply any normative requirement for the design of a program stream decoder. Indeed, non-audio/video data is also allowed, but not shown.

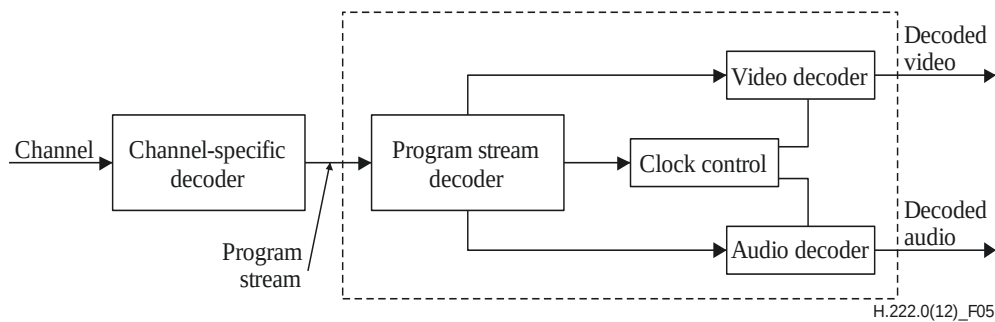


Figure Intro. 5 – Prototypical decoder for program streams

The prototypical decoder for program streams shown in Figure Intro. 5 is composed of system, video and audio decoders conforming to Parts 1, 2 and 3, respectively, of ISO/IEC 13818. In this decoder, the multiplexed coded representation of one or more audio and/or video streams is assumed to be stored or communicated on some channel in some channel-specific format. The channel-specific format is not governed by this Recommendation | International Standard, nor is the channel-specific decoding part of the prototypical decoder.

The prototypical decoder accepts as input a program stream and relies on a program stream decoder to extract timing information from the stream. The program stream decoder demultiplexes the stream, and the elementary streams so produced serve as inputs to video and audio decoders, whose outputs are decoded video and audio signals. Included in the design, but not shown in the figure, is the flow of timing information among the program stream decoder, the video and audio decoders, and the channel-specific decoder. The video and audio decoders are synchronized with each other and with the channel using this timing information.

Program streams are constructed in two layers: a system layer and a compression layer. The input stream to the program stream decoder has a system layer wrapped about a compression layer. Input streams to the video and audio decoders have only the compression layer.

Operations performed by the prototypical decoder either apply to the entire program stream ("multiplex-wide operations"), or to individual elementary streams ("stream-specific operations"). The program stream system layer is divided into two sub-layers, one for multiplex-wide operations (the pack layer), and one for stream-specific operations (the PES packet layer).

Intro. 3 Conversion between transport stream and program stream

It may be possible and reasonable to convert between transport streams and program streams by means of PES packets. This results from the specification of transport stream and program stream as embodied in 2.4.1 and 2.5.1 of the normative requirements of this Recommendation | International Standard. PES packets may, with some constraints, be mapped directly from the payload of one multiplexed bit stream into the payload of another multiplexed bit stream. It is possible to identify the correct order of PES packets in a program to assist with this if the `program_packet_sequence_counter` is present in all PES packets.

Certain other information necessary for conversion, e.g., the relationship between elementary streams, is available in tables and headers in both streams. Such data, if available, shall be correct in any stream before and after conversion.

Intro. 4 Packetized elementary stream

Transport streams and program streams are each logically constructed from PES packets, as indicated in the syntax definitions in 2.4.3.6. PES packets shall be used to convert between transport streams and program streams; in some cases the PES packets need not be modified when performing such conversions. PES packets may be much larger than the size of a transport stream packet.

A continuous sequence of PES packets of one elementary stream with one stream ID may be used to construct a PES Stream. When PES packets are used to form a PES stream, they shall include elementary stream clock reference (ESCR) fields and elementary stream rate (ES_Rate) fields, with constraints as defined in 2.4.3.8. The PES stream data shall be contiguous bytes from the elementary stream in their original order. PES streams do not contain some necessary system information which is contained in program streams and transport streams. Examples include the information in the pack header, system header, program stream map, program stream directory, program map table, and elements of the transport stream packet syntax.

The PES stream is a logical construct that may be useful within implementations of this Recommendation | International Standard; however, it is not defined as a stream for interchange and interoperability. Applications requiring streams containing only one elementary stream can use program streams or transport streams which each contain only one elementary stream. These streams contain all of the necessary system information. Multiple program streams or transport streams, each containing a single elementary stream, can be constructed with a common time base and therefore carry a complete program, i.e., with audio and video.

Intro. 5 Timing model

Systems, video and audio all have a timing model in which the end-to-end delay from the signal input to an encoder to the signal output from a decoder is a constant. This delay is the sum of encoding, encoder buffering, multiplexing, communication or storage, demultiplexing, decoder buffering, decoding, and presentation delays. As part of this timing model all video pictures and audio samples are presented exactly once, unless specifically coded to the contrary, and the inter-picture interval and audio sample rate are the same at the decoder as at the encoder. The system stream coding contains timing information which can be used to implement systems which embody constant end-to-end delay. It is possible to implement decoders which do not follow this model exactly; however, in such cases it is the decoder's responsibility to perform in an acceptable manner. The timing is embodied in the normative specifications of this Recommendation | International Standard, which must be adhered to by all valid bit streams, regardless of the means of creating them.

All timing is defined in terms of a common system clock, referred to as a system time clock (STC). In the program stream this clock may have an exactly specified ratio to the video or audio sample clocks, or it may have an operating frequency which differs slightly from the exact ratio while still providing precise end-to-end timing and clock recovery.

In the transport stream the system clock frequency is constrained to have the exactly specified ratio to the audio and video sample clocks at all times; the effect of this constraint is to simplify sample rate recovery in decoders.

Intro. 6 Conditional access

Encryption and scrambling for conditional access to programs encoded in the program and transport streams is supported by the system data stream definitions. Conditional access mechanisms are not specified here. The stream definitions are designed so that implementation of practical conditional access systems is reasonable, and there are some syntactical elements specified which provide specific support for such systems.

Intro. 7 Multiplex-wide operations

Multiplex-wide operations include the coordination of data retrieval of the channel, the adjustment of clocks, and the management of buffers. The tasks are intimately related. If the rate of data delivery of the channel is controllable, then data delivery may be adjusted so that decoder buffers neither overflow nor underflow; but if the data rate is not controllable, then elementary stream decoders must slave their timing to the data received from the channel to avoid overflow or underflow.

Program streams are composed of packs whose headers facilitate the above tasks. Pack headers specify intended times at which each byte is to enter the program stream Decoder from the channel, and this target arrival schedule serves as a reference for clock correction and buffer management. The schedule need not be followed exactly by decoders, but they must compensate for deviations about it.

Similarly, transport streams are composed of transport stream packets with headers containing information which specifies the times at which each byte is intended to enter a transport stream decoder from the channel. This schedule provides exactly the same function as that which is specified in the program stream.

An additional multiplex-wide operation is a decoder's ability to establish what resources are required to decode a transport stream or program stream. The first pack of each program stream conveys parameters to assist decoders in this task. Included, for example, are the stream's maximum data rate and the highest number of simultaneous video channels. The transport stream likewise contains globally useful information.

The transport stream and program stream each contain information which identifies the pertinent characteristics of, and relationships between, the elementary streams which constitute each program. Such information may include the language spoken in audio channels, as well as the relationship between video streams when multi-layer video coding is implemented.

Intro. 8 Individual stream operations (PES packet layer)

The principal stream-specific operations are:

- 1) demultiplexing; and
- 2) synchronizing playback of multiple elementary streams.

Intro. 8.1 Demultiplexing

On encoding, program streams are formed by multiplexing elementary streams, and transport streams are formed by multiplexing elementary streams, program streams, or the contents of other transport streams. Elementary streams may include private, reserved, and padding streams in addition to audio and video streams. The streams are temporally subdivided into packets, and the packets are serialized. A PES packet contains coded bytes from one and only one elementary stream.

In the program stream both fixed and variable packet lengths are allowed subject to constraints as specified in 2.5.1 and 2.5.2. For transport streams the packet length is 188 bytes. Both fixed and variable PES packet lengths are allowed, and will be relatively long in most applications.

On decoding, demultiplexing is required to reconstitute elementary streams from the multiplexed program stream or transport stream. Stream_id codes in program stream packet headers, and packet ID codes in the transport stream make this possible.

Intro. 8.2 Synchronization

Synchronization among multiple elementary streams is accomplished with presentation time stamps (PTSs) in the program stream and transport streams. Time stamps are generally in units of 90 kHz, but the system clock reference (SCR), the program clock reference (PCR) and the optional elementary stream clock reference (ESCR) have extensions with a resolution of 27 MHz. Decoding of N-elementary streams is synchronized by adjusting the decoding of streams to a common master time base rather than by adjusting the decoding of one stream to match that of another. The master time base may be one of the N-decoders' clocks, the data source's clock, or it may be some external clock.

Each program in a transport stream, which may contain multiple programs, may have its own time base. The time bases of different programs within a transport stream may be different.

Because PTSs apply to the decoding of individual elementary streams, they reside in the PES packet layer of both the transport streams and program streams. End-to-end synchronization occurs when encoders save time stamps at capture time, when the time stamps propagate with associated coded data to decoders, and when decoders use those time stamps to schedule presentations.

Synchronization of a decoding system with a channel is achieved through the use of the SCR in the program stream and by its analogue, the PCR, in the transport stream. The SCR and PCR are time stamps encoding the timing of the bit stream itself, and are derived from the same time base used for the audio and video PTS values from the same program. Since each program may have its own time base, there are separate PCR fields for each program in a transport stream containing multiple programs. In some cases it may be possible for programs to share PCR fields. Refer to 2.4.4, program-specific information (PSI), for the method of identifying which PCR is associated with a program. A program shall have one and only one PCR time base associated with it.

Intro. 8.3 Relation to compression layer

The PES packet layer is independent of the compression layer in some senses, but not in all. It is independent in the sense that PES packet payloads need not start at compression layer start codes, as defined in Parts 2 and 3 of ISO/IEC 13818. For example, video start codes may occur anywhere within the payload of a PES packet, and start codes may be split by a PES packet header. However, time stamps encoded in PES packet headers apply to presentation times of compression layer constructs (namely, presentation units). In addition, when the elementary stream data conforms to Rec. ITU-T H.262 | ISO/IEC 13818-2 or ISO/IEC 13818-3, the PES_packet_data_bytes shall be byte aligned to the bytes of this Recommendation | International Standard.

Intro. 9 System reference decoder

Part 1 of ISO/IEC 13818 employs a "system target decoder" (STD), one for transport streams (refer to 2.4.2) referred to as "transport system target decoder" (T-STD) and one for program streams (refer to 2.5.2) referred to as "program system target decoder" (P-STD), to provide a formalism for timing and buffering relationships. Because the STD is parameterized in terms of Rec. ITU-T H.222.0 | ISO/IEC 13818-1 fields (for example, buffer sizes) each elementary stream leads to its own parameterization of the STD. Encoders shall produce bit streams that meet the appropriate STD's constraints. Physical decoders may assume that a stream plays properly on its STD. The physical decoder must compensate for ways in which its design differs from that of the STD.

Intro. 10 Applications

The streams defined in this Recommendation | International Standard are intended to be as useful as possible to a wide variety of applications. Application developers should select the most appropriate stream.

ISO/IEC 13818-1:2023

Modern data communications networks may be capable of supporting Rec. ITU-T H.222.0 | ISO/IEC 13818-1 video and ISO/IEC 13818 audio. A real-time transport protocol is required. The program stream may be suitable for transmission on such networks.

The program stream is also suitable for multimedia applications on CD-ROM. Software processing of the program stream may be appropriate.

The transport stream may be more suitable for error-prone environments, such as those used for distributing compressed bit-streams over long-distance networks and in broadcast systems.

Many applications require storage and retrieval of Rec. ITU-T H.222.0 | ISO/IEC 13818-1 bitstreams on various digital storage media (DSM). A digital storage media command and control (DSM-CC) protocol is specified in Annex B and Part 6 of ISO/IEC 13818 in order to facilitate the control of such media.

**INTERNATIONAL STANDARD
ITU-T RECOMMENDATION**

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

SECTION 1 – GENERAL

1.1 Scope

This Recommendation | International Standard specifies the system layer of the coding. It was developed principally to support the combination of the video and audio coding methods defined in Parts 2 and 3 of ISO/IEC 13818. The system layer supports six basic functions:

- 1) the synchronization of multiple compressed streams on decoding;
- 2) the interleaving of multiple compressed streams into a single stream;
- 3) the initialization of buffering for decoding start up;
- 4) continuous buffer management;
- 5) time identification;
- 6) multiplexing and signalling of various components in a system stream.

A Rec. ITU-T H.222.0 | ISO/IEC 13818-1 multiplexed bit stream is either a transport stream or a program stream. Both streams are constructed from PES packets and packets containing other necessary information. Both stream types support multiplexing of video and audio compressed streams from one program with a common time base. The transport stream additionally supports the multiplexing of video and audio compressed streams from multiple programs with independent time bases. For almost error-free environments the program stream is generally more appropriate, supporting software processing of program information. The transport stream is more suitable for use in environments where errors are likely.

A Rec. ITU-T H.222.0 | ISO/IEC 13818-1 multiplexed bit stream, whether a transport stream or a program stream, is constructed in two layers: the outermost layer is the system layer, and the innermost is the compression layer. The system layer provides the functions necessary for using one or more compressed data streams in a system. The video and audio parts of this Specification define the compression coding layer for audio and video data. Coding of other types of data is not defined by this Specification, but is supported by the system layer provided that the other types of data adhere to the constraints defined in 2.7.

1.2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

1.2.1 Identical Recommendations | International Standards

- Recommendation ITU-T H.262 (2012) | ISO/IEC 13818-2:2013, *Information technology – Generic coding of moving pictures and associated audio information: Video*.
- Recommendation ITU-T T.800 (2019) | ISO/IEC 15444-1:2019, *Information technology – JPEG 2000 image coding system: Core coding system*.

1.2.2 Paired Recommendations | International Standards equivalent in technical content

- Recommendation ITU-T H.264 (2019), *Advanced video coding for generic audiovisual services*.
ISO/IEC 14496-10:2020, *Information technology – Coding of audio-visual objects – Part 10: Advanced video coding*.
- Recommendation ITU-T H.265 (2019), *High efficiency video coding*.
ISO/IEC 23008-2:2020, *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding*.

- Recommendation. ITU-T H.273 (2021), *Coding-independent code points for video signal type identification*.
ISO/IEC 23091-2:2021, *Information technology — Coding-independent code points — Part 2: Video*.
- Recommendation ITU-T T.171 (1996), *Protocols for interactive audiovisual services: coded representation of multimedia and hypermedia objects*.
ISO/IEC 13522-1:1997, *Information technology – Coding of Multimedia and Hypermedia information – Part 1: MHEG object representation – Base notation (ASN.1)*.
- Recommendation ITU-T H.266 (2020), *Versatile video coding*.
ISO/IEC 23090-3:2021, *Information technology – Coded Representation of Immersive Media – Part 3: Versatile video coding*.
- Recommendation ITU-T H.274 (2020), *Versatile supplemental enhancement information messages for coded video bitstreams*.
ISO/IEC 23002-7:2021 – *Information Technology – MPEG Video technologies – Part 7: Versatile supplemental enhancement information messages for coded video bitstreams*.

1.2.3 Additional references

- Recommendation ITU-R BT.709-6 (2015), *Parameter values for the HDTV standards for production and international programme exchange*.
- Recommendation ITU-R BT.1886 (2011), *Reference electro-optical transfer function for flat panel displays used in HDTV studio production*.
- Recommendation ITU-R BT.2020 (2015), *Parameter values for ultra-high definition television systems for production and international programme exchange*.
- Recommendation ITU-R BT.2100-2 (2018), *Image parameter values for high dynamic range television for use in production and international programme exchange*.
- ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code*.
- ISO 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
- ISO 15706-1:2002, *Information and documentation – International Standard Audiovisual Number (ISAN) – Part 1: Audiovisual work identifier*.
- ISO 15706-2:2007, *Information and documentation – International Standard Audiovisual Number (ISAN) – Part 2: Version identifier*.
- ISO/IEC 11172-1:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 1: Systems*.
- ISO/IEC 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video*.
- ISO/IEC 11172-3:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*.
- ISO/IEC 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*.
- ISO/IEC 13818-6:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC*.
- ISO/IEC 13818-7:2006, *Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*.
- ISO/IEC 13818-11:2004, *Information technology – Generic coding of moving pictures and associated audio information – Part 11: IPMP on MPEG-2 systems*.
- ISO/IEC 14496-1:2010, *Information technology – Coding of audio-visual objects – Part 1: Systems*.
- ISO/IEC 14496-2:2004, *Information technology – Coding of audio-visual objects – Part 2: Visual*.
- ISO/IEC 14496-3:2019, *Information technology – Coding of audio-visual objects – Part 3: Audio*.
- ISO/IEC 14496-17:2006, *Information technology, Coding of audio-visual objects – Part 17: Streaming text format*.
- ISO/IEC 21122-1:2019, *JPEG XS low-latency lightweight image coding system – Part 1: Core coding system*.

- ISO/IEC 21122-2:2019, *JPEG XS low-latency lightweight image coding system – Part 2: Profiles and buffer models*.
- ISO/IEC 21122-3:2019, *JPEG XS low-latency lightweight image coding system – Part 3: Transport and container formats*.
- ISO/IEC 23001-8:2016, *Information technology – MPEG systems technologies – Part 8: Coding-independent code-points*.
- ISO/IEC 23001-10:2020, *Information technology – MPEG systems technologies – Part 10: Carriage of timed metadata metrics of media in ISO base media file format*.
- ISO/IEC 23001-11:2019, *Information technology – MPEG systems technologies – Part 11: Energy-efficient media consumption (Green Metadata)*.
- ISO/IEC 23001-12:2018, *Information technology – MPEG systems technologies – Part 12: Sample Variants*.
- ISO/IEC 23001-13:2019, *Information technology – MPEG systems technologies – Part 13: Media Orchestration*.
- ISO/IEC 23003-3:2020, *Information technology – MPEG audio technologies – Part 3: Unified speech and audio coding*.
- ISO/IEC 23003-4:2020, *Information technology – MPEG audio technologies – Part 4: Dynamic Range Control*.
- ISO/IEC 23008-3:2019, *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 3: 3D audio*.
- ISO/IEC 23091-2:2019, *Coding-independent code points – Part 2: Video*.
- ISO/IEC 23094-1:2020, *Information technology – General video coding – Part 1: Essential video coding*.
- ISO/IEC 23094-2:2021, *Information technology – General video coding – Part 2: Low complexity enhancement video coding*.
- ANSI/SCTE 35 (2019), *Digital Program Insertion Cueing Message for Cable*.
- IETF BCP 47: IETF RFC 4647 (2006) *Matching of Language Tags* combined with IETF RFC 5646 (2009), *Tags for Identifying Languages*.
- IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax*.
- IETF RFC 5484 (2009), *Associating Time-Codes with RTP Streams*.

SECTION 2 – TECHNICAL ELEMENTS

2.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply. If specific to a Part, this is parenthetically noted.

2.1.1 access unit (system): A coded representation of a presentation unit. In the case of audio, an access unit is the coded representation of an audio frame, whereby each audio frame carries data from one or more audio channels; an audio frame may carry for example one mono channel, or two stereo channels or seven surround sound channels.

In the case of video, an access unit includes all the coded data for a picture, and any stuffing that follows it, up to but not including the start of the next access unit. If a picture is not preceded by a `group_start_code` or a `sequence_header_code`, the access unit begins with the picture start code. If a picture is preceded by a `group_start_code` and/or a `sequence_header_code`, the access unit begins with the first byte of the first of these start codes. If it is the last picture preceding a `sequence_end_code` in the bitstream, all bytes between the last byte of the coded picture and the `sequence_end_code` (including the `sequence_end_code`) belong to the access unit.

For the definition of an access unit for Rec. ITU-T H.264 | ISO/IEC 14496-10 video, see the AVC access unit definition in 2.1.3.

In the case of an ISO/IEC 14496-17 text stream, see ISO/IEC 14496-17 for the definition of an access unit.

2.1.2 AVC 24-hour picture (system): An advanced video coding (AVC) access unit with a presentation time that is more than 24 hours in the future. For the purpose of this definition, AVC access unit n has a presentation time that is more than 24 hours in the future if the difference between the initial arrival time $t_{ai}(n)$ and the DPB output time $t_{o,dpb}(n)$ is more than 24 hours.

2.1.3 AVC access unit (system): An access unit as defined for byte streams in Rec. ITU-T H.264 | ISO/IEC 14496-10 with the constraints specified in 2.14.1.

2.1.4 AVC slice (system): A `byte_stream_nal_unit` as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10 with `nal_unit_type` values of 1 or 5, or a `byte_stream_nal_unit` data structure with `nal_unit_type` value of 2 and any associated `byte_stream_nal_unit` data structures with `nal_unit_type` equal to 3 and/or 4.

2.1.5 AVC still picture (system): An AVC still picture consists of an AVC access unit containing an IDR picture, preceded by SPS and PPS NAL units that carry sufficient information to correctly decode the IDR picture. Preceding an AVC still picture, there shall be another AVC still picture or an end of sequence NAL unit terminating a preceding coded video sequence unless the AVC still picture is the very first access unit in the video stream.

2.1.6 AVC video sequence (system): Coded video sequence as defined in 3.30 of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.7 AVC video stream (system): A Rec. ITU-T H.264 | ISO/IEC 14496-10 stream. An AVC video stream consists of one or more AVC video sequences. An AVC video stream may also result from re-assembling video sub-bitstreams.

2.1.8 AVC video sub-bitstream of MVC: The video sub-bitstream that contains the base view as defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10, containing all VCL NAL units associated with the minimum value of view order index present in each AVC video sequence of the AVC video stream. The AVC video sub-bitstream of MVC may additionally contain the associated NAL units with `nal_unit_type` syntax element equal to 14 (prefix NAL units), as defined for MVC in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.9 AVC video sub-bitstream of MVCD: The video sub-bitstream that contains the base view as defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10, containing all VCL NAL units associated with the minimum value of view order index present in each AVC video sequence of the AVC video stream. The AVC video sub-bitstream of MVCD may additionally contain the associated NAL units with `nal_unit_type` syntax element equal to 14 (prefix NAL units), as defined for MVC in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.10 AVC video sub-bitstream of SVC: The video sub-bitstream that contains the base layer as defined in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10 and that shall additionally contain NAL units with `nal_unit_type` equal to 14 (prefix NAL units) as defined for scalable video coding (SVC) in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10. The AVC video sub-bitstream of SVC contains all VCL NAL units associated with `dependency_id` equal to 0.

2.1.11 bitrate: The rate at which the compressed bit stream is delivered from the channel to the input of a decoder.

2.1.12 byte aligned: A bit in a coded bit stream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.

2.1.13 channel: A digital medium that stores or transports a Rec. ITU-T H.222.0 | ISO/IEC 13818-1 stream.

- 2.1.14 coded B-frame:** A B-frame picture or a pair of B-field pictures.
- 2.1.15 coded frame:** A coded frame is a coded I-frame, coded B-frame or a coded P-frame.
- 2.1.16 coded I-frame:** An I-frame picture or a pair of field pictures where the first field picture is an I-picture and the second field picture is either an I-picture or a P-picture.
- 2.1.17 coded P-frame:** A P-frame picture or a pair of P-field pictures.
- 2.1.18 coded representation:** A data element as represented in its encoded form.
- 2.1.19 compression:** Reduction in the number of bits used to represent an item of data.
- 2.1.20 constant bitrate:** Operation where the bitrate is constant from start to finish of the compressed bit stream.
- 2.1.21 constrained system parameter stream (CSPS) (system):** A program stream for which the constraints defined in 2.7.9 apply.
- 2.1.22 cyclic redundancy check (CRC):** The CRC to verify the correctness of data.
- 2.1.23 data element:** An item of data as represented before encoding and after decoding.
- 2.1.24 decoded stream:** The decoded reconstruction of a compressed bit stream.
- 2.1.25 decoder:** An embodiment of a decoding process.
- 2.1.26 decoding (process):** The process defined in this Recommendation | International Standard that reads an input-coded bit stream and outputs decoded pictures or audio samples.
- 2.1.27 decoding time-stamp (DTS) (system):** A field that may be present in a PES packet header that indicates the time that an access unit is decoded in the system target decoder.
- 2.1.28 digital storage media (DSM):** A digital storage or transmission device or system.
- 2.1.29 DSM-CC:** Digital storage media command and control.
- 2.1.30 entitlement control message (ECM):** Entitlement control messages are private conditional access information which specify control words and possibly other, typically stream-specific, scrambling and/or control parameters.
- 2.1.31 entitlement management message (EMM):** Entitlement management messages are private conditional access information which specify the authorization levels or the services of specific decoders. They may be addressed to single decoders or groups of decoders.
- 2.1.32 editing:** The process by which one or more compressed bit streams are manipulated to produce a new compressed bit stream. Edited bit streams meet the same requirements as streams which are not edited.
- 2.1.33 elementary stream (ES) (system):** A generic term for one of the coded video, coded audio or other coded bit streams in PES packets. One elementary stream is carried in a sequence of PES packets with one and only one stream_id.
- 2.1.34 elementary stream clock reference (ESCR) (system):** A time stamp in the PES stream from which decoders of PES streams may derive timing.
- 2.1.35 encoder:** An embodiment of an encoding process.
- 2.1.36 encoding (process):** A process, not specified in this Recommendation | International Standard, that reads a stream of input pictures or audio samples and produces a coded bit stream conforming to this Recommendation.
- 2.1.37 entropy coding:** Variable length lossless coding of the digital representation of a signal to reduce redundancy.
- 2.1.38 EVC video stream:** A raw bitstream as specified in ISO/IEC 23094-1 Annex B.
- 2.1.39 EVC access unit:** An access unit as defined in ISO/IEC 23094-1 with the constraints specified in 2.24.1.
- 2.1.40 EVC slice:** A slice as specified in ISO/IEC 23094-1.
- 2.1.41 EVC video sequence (system):** coded video sequence as defined in 23094-1.
- 2.1.42 EVC still picture:** an EVC access unit containing an IDR picture, preceded by APS, SPS, PPS and other NAL units that carry sufficient information to correctly decode that picture, and also optionally preceded by another EVC still picture or an end of sequence NAL unit terminating a preceding coded video sequence when the current picture is not the first access unit in the video bitstream.
- 2.1.43 EVC video sub-bitstream:** A subset of the NAL units of an EVC video stream in their original order.

2.1.44 EVC temporal video sub-bitstream: An EVC video sub-bitstream that contains all VCL NAL units and associated non-VCL NAL units of the temporal sub-layer, as specified in ISO/IEC 23094-1, associated to TemporalId equal to 0 and which may additionally contain all VCL NAL units and associated non-VCL NAL units of all temporal sub-layers associated to a contiguous range of TemporalId.

2.1.45 event: An event is defined as a collection of elementary streams with a common time base, an associated start time, and an associated end time.

2.1.46 fast forward playback (video): The process of displaying a sequence, or parts of a sequence, of pictures in display-order faster than real-time.

2.1.47 forbidden: The term "forbidden", when used in the clauses of this Recommendation | International Standard defining the coded bit stream, indicates that the value specified shall never be used.

2.1.48 green access unit – An access unit that contains dynamic metadata in a message format as defined in 6.2.1 of ISO/IEC 23001-11.

2.1.49 HEVC 24-hour picture (system): An HEVC access unit with a presentation time that is more than 24 hours in the future. For the purpose of this definition, HEVC access unit n has a presentation time that is more than 24 hours in the future if the difference between the initial arrival time $t_{ai}(n)$ and the DPB output time $t_{o,dpb}(n)$ is more than 24 hours.

2.1.50 HEVC access unit: An access unit as defined in Annex F of Rec. ITU-T H.265 | ISO/IEC 23008-2 with the constraints specified in 2.17.1.

2.1.51 HEVC base layer: HEVC layer with `nuh_layer_id` equal to 0.

2.1.52 HEVC base sub-partition: HEVC video sub-bitstream that is also a conforming bitstream as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2, which contains all VCL NAL units and the associated non-VCL NAL units of an HEVC base layer up to a target highest TemporalId identified by a target HEVC operation point.

2.1.53 HEVC complete temporal representation: A sub-layer representation as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2 that contains all temporal sub-layers up to the temporal sub-layer with TemporalId equal to `sps_max_sub_layers_minus1+1` as included in the active sequence parameter set, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.54 HEVC dependent slice segment: An HEVC slice segment with the syntax element `dependent_slice_segment_flag` in the slice header set to a value equal to 1, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.55 HEVC enhancement sub-partition: One HEVC layer with a particular value of `nuh_layer_id` greater than 0 in the NAL unit header syntax element or an HEVC temporal video sub-bitstream or HEVC temporal video subset thereof, of which the HEVC layer aggregation with an HEVC base sub-partition and zero or more other HEVC sub-partitions, according to HEVC layer list, results in a valid HEVC layered video stream.

2.1.56 HEVC highest temporal sub-layer representation: The sub-layer representation of the temporal sub-layer with the highest value of TemporalId, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2, in the associated HEVC temporal video sub-bitstream or HEVC temporal video subset.

2.1.57 HEVC independent slice segment: An HEVC slice segment with the syntax element `dependent_slice_segment_flag` in the slice header set to a value 0 or inferred to be equal to 0, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.58 HEVC layer: HEVC video sub-bitstream that contains all VCL NAL units with a particular value of `nuh_layer_id` in the NAL unit header syntax element and associated non-VCL NAL units, as defined in Annex F of Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.59 HEVC layer aggregation: Successive HEVC layer component aggregation of all HEVC layer components in an HEVC video sequence.

2.1.60 HEVC layer component: VCL NAL units and the associated non-VCL NAL units of an HEVC access unit which belong to an HEVC sub-partition.

2.1.61 HEVC layer component aggregation: Concatenation of all HEVC layer components with the same output time from all HEVC sub-partitions indicated in an HEVC layer list in the order indicated by the HEVC layer list, resulting in a valid HEVC access unit as defined in Annex F of Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.62 HEVC layer list: Ordered list of HEVC sub-partitions for a target HEVC operation point of which the HEVC layer aggregation results in a valid HEVC layered video stream.

NOTE – An HEVC layer list is signalled for each target HEVC operation point using the HEVC operation point descriptor.

2.1.63 HEVC layered video stream: HEVC video stream that contains all VCL NAL units and associated non-VCL NAL units conforming to one or more profiles defined in Annex G or Annex H of Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.64 HEVC operation point: Operation point based on a target highest TemporalId, and a target layer identifier list as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2.

NOTE – Rec. ITU-T H.265 | ISO/IEC 23008-2 specifies the sub-bitstream extraction process for an operation point according to which the operation point is a conforming bitstream. An operation point is associated with an HEVC layered video stream or HEVC base layer.

2.1.65 HEVC slice: An HEVC independent slice segment and zero or more subsequent HEVC dependent slice segments preceding the next HEVC independent slice segment (if any) within the same HEVC access unit.

2.1.66 HEVC slice segment: A byte_stream_nal_unit with nal_unit_type in the range of 0 to 9 and 16 to 23, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.67 HEVC still picture (system): An HEVC still picture consists of an HEVC access unit containing an IDR picture preceded by VPS, SPS and PPS NAL units, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2, that carry sufficient information to correctly decode this IDR picture. Preceding an HEVC still picture, there shall be another HEVC still picture or an end of sequence NAL unit terminating a preceding coded video sequence, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.68 HEVC sub-partition: Either an HEVC base sub-partition or an HEVC enhancement sub-partition.

NOTE – An HEVC sub-partition can either be an HEVC temporal video sub-bitstream if it includes VCL NAL units with the minimum value of TemporalId (i.e., including TemporalId equal to 0), or it can be an HEVC temporal video subset, if it complements an HEVC base sub-partition or HEVC enhancement sub-partition with the same target layer identifier.

2.1.69 HEVC temporal enhancement sub-partition: An HEVC temporal video subset of the same HEVC layer as another HEVC enhancement sub-partition of the same HEVC video stream which contains one or more complementary temporal sub-layers, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.70 HEVC temporal video sub-bitstream: An HEVC video sub-bitstream that contains all VCL NAL units and associated non-VCL NAL units of the temporal sub-layer of the same layer, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2, associated with TemporalId equal to 0 and which may additionally contain all VCL NAL units and associated non-VCL NAL units of all temporal sub-layers of the same layer associated with a contiguous range of TemporalId from 1 to a value equal to or smaller than sps_max_sub_layers_minus1 included in the active sequence parameter set, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.71 HEVC temporal video subset: An HEVC video sub-bitstream that contains all VCL NAL units and the associated non-VCL NAL units of one or more temporal sub-layers of the same layer, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2, with each temporal sub-layer not being present in the corresponding HEVC temporal video sub-bitstream and TemporalId associated with each temporal sub-layer forming a contiguous range of values that is equal to or smaller than sps_max_sub_layers_minus1 included in the active sequence parameter set, as specified in Rec. ITU-T H.265 | ISO/IEC 23008-2.

NOTE – According to the constraints for the transport of HEVC specified in 2.17.1, each temporal sub-layer of an *HEVC video stream* is present either in the HEVC temporal video sub-bitstream or in exactly one HEVC temporal video subset which is carried in a set of elementary streams that are associated by hierarchy descriptors or HEVC hierarchy extension descriptors. This prevents multiple inclusions of the same temporal sub-layer and allows aggregation of the HEVC temporal video sub-bitstream with associated HEVC temporal video subsets according to the hierarchy descriptors, as specified in 2.17.3 and according to the hierarchy descriptors or HEVC hierarchy extension descriptors, as specified in 2.17.4.

2.1.72 HEVC tile of slices: One or more consecutive HEVC slices which form the coded representation of a tile, as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.73 HEVC tile substream: Substream of a Rec. ITU-T H.265 | ISO/IEC 23008 2 video stream that contains a Motion Constrained Tile Set, parameter sets, slice headers or a combination thereof.

2.1.74 HEVC video sequence (system): A coded video sequence as defined in Rec. ITU-T H.265 | ISO/IEC 23008-2.

2.1.75 HEVC video stream: Byte stream as specified in Rec. ITU-T H. 265 | ISO/IEC 23008-2 Annex B.

NOTE – This term represents either a byte stream as specified in Annex B of the first version of Rec. ITU-T H.265 | ISO/IEC 23008-2 or an HEVC layered video sub-bitstream.

2.1.76 HEVC video sub-bitstream: A subset of the NAL units of an HEVC video stream in their original order.

2.1.77 JPEG 2000 (J2K) video access unit: The JPEG 2000 codestream or codestreams comprising a decodable and randomly accessible (portion of) image, preceded by all the parameters required to decode the access unit and display the decoded data.

2.1.78 J2K block: The JPEG 2000 codestream or codestreams corresponding to a rectangular portion of a video frame, as detailed in S.3.

NOTE – Usage of J2K blocks requires J2K block mode (defined in 2.1.79)

2.1.79 J2K block mode: Optional mode defined in S.3, dividing each frame of a J2K video stream in a certain amount of rectangular blocks, each encoded as an independent J2K block (defined in 2.1.78).

2.1.80 J2K still picture (system): J2K video access unit as defined in 2.1.87 with constraints as specified in S.2.

2.1.81 J2K stripe: The JPEG 2000 codestream or codestreams comprising a decodable horizontally divided portion of an image, as detailed in S.4.

NOTE – Usage of J2K stripes requires J2K stripe mode (defined in 2.1.82) to be enabled in the J2K video descriptor. Such usage enables transport of a J2K video stream with a low end-to-end latency.

2.1.82 J2K stripe mode: Optional mode defined in S.4, dividing the (portion of) image transported in a J2K video access unit in a succession of horizontal stripes, each encoded as an independent J2K stripe (defined in 2.1.81).

2.1.83 J2K video elementary stream: Video elementary stream consisting of a succession of J2K video access units.

2.1.84 J2K video sequence: J2K video elementary stream where all the access units have the same profile/level, J2K video access unit coding parameters and video parameters.

2.1.85 JPEG XS elementary stream header (jxes header): All parameters required to decode a JPEG XS video access unit and display the decoded data.

2.1.86 JPEG XS still picture (system): JPEG XS video access unit as defined in 2.1.87 with constraints as specified in W.2.

2.1.87 JPEG XS video access unit: The JPEG XS codestream or multiple JPEG XS codestreams, as defined in ISO/IEC 21122-1, comprising a decodable and randomly accessible image, preceded by a JPEG XS elementary stream header.

2.1.88 JPEG XS video elementary stream: Video elementary stream consisting of a succession of JPEG XS video access units.

2.1.89 JPEG XS video sequence: JPEG XS video elementary stream where all the access units have the same profile, level and sublevel (as defined in ISO/IEC 21122-2), JPEG XS video access unit coding parameters, and video parameters.

2.1.90 layer (video and systems): One of the levels in the data hierarchy of the video and system specifications defined in Parts 1 and 2 of this Recommendation | International Standard.

2.1.91 LCEVC access unit (system): An access unit as defined for byte streams in ISO/IEC 23094-2 with the constraints specified in 2.25.1.

2.1.92 LCEVC video sequence (system): Coded video sequence as defined in ISO/IEC 23094-2.

2.1.93 MCTS: Motion Constrained Tile Set according to Rec. ITU-T H.265 | ISO/IEC 23008-2.

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

2.1.95 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.

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

2.1.97 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.

2.1.98 metadata format: Identifies the coding format of metadata.

2.1.99 metadata service: A coherent set of metadata of the same format delivered to a receiver for a specific purpose.

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

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

2.1.102 (multiplexed) stream (system): A bit stream composed of 0 or more elementary streams combined in a manner that conforms to this Recommendation | International Standard.

2.1.103 MVC base view sub-bitstream: The MVC base view sub-bitstream is defined to contain the AVC video sub-bitstream of MVC conforming to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 and one additional MVC video sub-bitstream associated with an MVC view_id subset including the view order index that immediately follows the view order index associated with the base view.

NOTE – The MVC base view sub-bitstream is also an AVC video stream where no re-assembly is required before decoding.

2.1.104 MVC operation point: An MVC operation point is identified by a temporal_id value representing a target temporal level and a set of view_id values representing the target output views. One MVC operation point is associated with an AVC video stream which conforms to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10. The AVC video stream associated with an MVC operation point is re-assembled from a set consisting of one or more of the following items: AVC video sub-bitstream of MVC, MVC base view sub-bitstream, MVC video sub-bitstreams.

2.1.105 MVC slice (system): A byte_stream_nal_unit with nal_unit_type syntax element equal to 20 of an AVC video stream which conforms to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10.

NOTE – As specified in Rec. ITU-T H.264 | ISO/IEC 14496-10, the value of svc_extension_flag is set equal to 0 for coded video sequences conforming to one or more profiles specified in Annex H. MVC slices should not include NAL units for which nal_unit_type is equal to 20 with svc_extension_flag equal to 1.

2.1.106 MVC video sub-bitstream: The MVC video sub-bitstream is defined to be all VCL NAL units with nal_unit_type equal to 20 associated with the same multiview video coding (MVC) view_id subset of an advanced video coding (AVC) video stream and associated non-VCL NAL units which conform to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10.

NOTE – In contrast to a sub-bitstream as specified in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10, an MVC video sub-bitstream according to this Specification is not necessarily a decodable MVC video sub-bitstream. The one exception is when an MVC video sub-bitstream is also an MVC base view sub-bitstream. Re-assembling MVC video sub-bitstreams in an increasing order of view order index, starting from the lowest value of view order index up to any value of view order index, results in a decodable AVC video stream.

2.1.107 MVC view_id subset: A set of one or more view_id values, as defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 in the NAL unit header syntax element, associated with one set of consecutive view order index values.

NOTE – An MVC video sub-bitstream or MVC base view sub-bitstream based on a specific MVC view_id subset may not include view components for all view_id values included in that MVC view_id subset. One or more view order index values may be skipped if the view associated with a missing view order index value is not required for decoding the transmitted views.

2.1.108 MVCD base view sub-bitstream: The MVCD base view sub-bitstream is defined to contain the AVC video sub-bitstream of MVCD conforming to one or more profiles defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10 and one additional MVCD video sub-bitstream associated with an MVCD view_id subset including the view order index that immediately follows the view order index associated with the base view.

NOTE – The MVCD base view sub-bitstream is also an AVC video stream where no re-assembly is required before decoding.

2.1.109 MVCD slice (system): A byte_stream_nal_unit with nal_unit_type syntax element equal to 21 of an AVC video stream which conforms to one or more profiles defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.110 MVCD video sub-bitstream: The MVCD video sub-bitstream is defined to be all VCL NAL units with nal_unit_type equal to 21 associated with the same MVCD view_id subset of an AVC video stream and associated non-VCL NAL units which conform to one or more profiles defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10.

NOTE – In contrast to a sub-bitstream as specified in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10, an MVCD video sub-bitstream according to this Specification is not necessarily a decodable MVCD video sub-bitstream. The one exception is when an MVCD video sub-bitstream is also an MVCD base view sub-bitstream. Re-assembling MVCD video sub-bitstreams in an increasing order of view order index, starting from the lowest value of view order index up to any value of view order index, results in a decodable AVC video stream.

2.1.111 MVC view-component subset: The VCL NAL units of an AVC access unit associated with the same MVC view_id subset and associated non-VCL NAL units.

NOTE – Re-assembling MVC view-component subsets ordered according to the view order index, starting from the minimum view order index up to the highest view order index present in the access unit, while reordering the non-VCL NAL units conforming to the order of NAL units within an access unit, as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10, results in an AVC access unit.

2.1.112 MVCD view-component subset: The VCL NAL units of an AVC access unit associated with the same MVCD view_id subset and associated non-VCL NAL units.

NOTE – Re-assembling MVCD view-component subsets ordered according to the view order index, starting from the minimum view order index up to the highest view order index present in the access unit, while reordering the non-VCL NAL units conforming to the order of NAL units within an access unit, as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10, results in an AVC access unit.

2.1.113 MVCD view_id subset: A set of one or more view_id values, as defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10 in the NAL unit header syntax element, associated with one set of consecutive view order index values.

NOTE – An MVCD video sub-bitstream or MVCD base view sub-bitstream based on a specific MVCD view_id subset may not include view components for all view_id values included in that MVCD view_id subset. One or more view order index values may be skipped if the view associated with a missing view order index value is not required for decoding the transmitted views.

2.1.114 pack (system): A pack consists of a pack header followed by zero or more packets. It is a layer in the system coding syntax described in 2.5.3.3.

2.1.115 packet data (system): Contiguous bytes of data from an elementary stream present in a packet.

2.1.116 packet identifier (PID) (system): A unique integer value used to identify elementary streams of a program in a single or multi-program transport stream as described in 2.4.3.

2.1.117 padding (audio): A method to adjust the average length of an audio frame in time to the duration of the corresponding PCM samples, by conditionally adding a slot to the audio frame.

2.1.118 payload: Payload refers to the bytes which follow the header bytes in a packet. For example, the payload of some transport stream packets includes a PES_packet_header and its PES_packet_data_bytes, or pointer_field and PSI sections, or private data; but a PES_packet_payload consists of only PES_packet_data_bytes. The transport stream packet header and adaptation fields are not payload.

2.1.119 PES (system): An abbreviation for a Packetized Elementary Stream.

2.1.120 PES packet (system): The data structure used to carry elementary stream data. A PES packet consists of a PES packet header followed by a number of contiguous bytes from an elementary data stream. It is a layer in the system coding syntax described in 2.4.3.6.

2.1.121 PES packet header (system): The leading fields in a PES packet up to and not including the PES_packet_data_byte fields, where the stream is not a padding stream. In the case of a padding stream the PES packet header is similarly defined as the leading fields in a PES packet up to and not including padding_byte fields.

2.1.122 packetized elementary stream (PES) (system): A PES system consists of PES packets, all of whose payloads consist of data from a single elementary stream, and all of which have the same stream_id. Specific semantic constraints apply. Refer to Intro. 4.

2.1.123 presentation time-stamp (PTS) (system): A field that may be present in a PES packet header that indicates the time that a presentation unit is presented in the system target decoder.

2.1.124 presentation unit (PU) (system): A decoded audio access unit or a decoded picture.

2.1.125 program (system): A program is a collection of program elements. Program elements may be elementary streams. Program elements need not have any defined time base; those that do, have a common time base and are intended for synchronized presentation.

2.1.126 program clock reference (PCR) (system): A time stamp in the transport stream from which decoder timing is derived.

2.1.127 program element (system): A generic term for one of the elementary streams or other data streams that may be included in a program.

2.1.128 program-specific information (PSI) (system): PSI consists of normative data which is necessary for the demultiplexing of transport streams and the successful regeneration of programs and is described in 2.4.4.1. An example of privately defined PSI data is the non-mandatory network information table.

2.1.129 quality access unit: An access unit that contains dynamic quality metadata as defined in ISO/IEC 23001-10.

2.1.130 random access: The process of beginning to read and decode the coded bit stream at an arbitrary point.

2.1.131 reserved: The term "reserved", when used in the clauses defining the coded bit stream, indicates that the value may be used in the future for ISO defined extensions. Unless otherwise specified within this Recommendation | International Standard, all reserved bits shall be set to '1'.

2.1.132 sample variant: An assembled media sample replacing an original sample as defined in ISO/IEC 23001-12:2018.

NOTE – In ISO/IEC 23001-12:2018, the term “sample” is used commonly for samples as defined in ISO Base Media File Format (ISO/IEC 14496-12) and for access units as defined in 2.1.1.

2.1.133 scrambling (system): The alteration of the characteristics of a video, audio or coded data stream in order to prevent unauthorized reception of the information in a clear form. This alteration is a specified process under the control of a conditional access system.

2.1.134 service-compatible: This is defined as 'simulcast' of two stereoscopic views which do not include scalable or temporal coding. The two views are independently compressed using MPEG-2 video or AVC or both and can be decoded independently.

2.1.135 source stream: A single non-multiplexed stream of samples before compression coding.

2.1.136 splicing (system): The concatenation, performed on the system level, of two different elementary streams. The resulting system stream conforms totally to this Recommendation | International Standard. The splice may result in discontinuities in time base, continuity counter, PSI, and decoding.

2.1.137 start codes (system): 32-bit codes embedded in the coded bit stream. They are used for several purposes including identifying some of the layers in the coding syntax. Start codes consist of a 24-bit prefix (0x000001) and an 8-bit stream_id as shown in Table 2-22.

2.1.138 STD input buffer (system): A first-in first-out buffer at the input of a system target decoder for storage of compressed data from elementary streams before decoding.

2.1.139 still picture: A still picture consists of a video sequence, coded as defined in Rec. ITU-T H.262 | ISO/IEC 13818-2, ISO/IEC 11172-2 or ISO/IEC 14496-2, that contains exactly one coded picture which is intra-coded. This picture has an associated PTS and in case of coding according to ISO/IEC 11172-2, Rec. ITU-T H.262 | ISO/IEC 13818-2 or ISO/IEC 14496-2, the presentation time of succeeding pictures, if any, is later than that of the still picture by at least two picture periods.

2.1.140 SVC dependency representation: The VCL NAL units of an AVC access unit associated with the same value of dependency_id which is provided as part of the NAL unit header or the associated prefix NAL unit header, and the associated non-VCL NAL units. Re-assembling SVC dependency representations in a consecutive order of dependency_id starting from the lowest value of dependency_id present in the access unit up to any value of dependency_id present in the access unit, while reordering the non-VCL NAL units conforming to the order of NAL units within an access unit as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10, results in an AVC access unit.

2.1.141 SVC slice (system): A byte_stream_nal_unit as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10 with nal_unit_type equal to 20 of an AVC video stream which conforms to one or more profiles defined in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10.

NOTE – As specified in Rec. ITU-T H.264 | ISO/IEC 14496-10, the value of svc_extension_flag is set equal to 1 for coded video sequences conforming to one or more profiles specified in Annex G. SVC slices should not include NAL units for which nal_unit_type is equal to 20 with svc_extension_flag equal to 0.

2.1.142 SVC video sub-bitstream: The video sub-bitstream that contains VCL NAL units with nal_unit_type equal to 20 with the same NAL unit header syntax element dependency_id not equal to 0.

2.1.143 system header (system): The system header is a data structure defined in 2.5.3.5 that carries information summarizing the system characteristics of Rec. ITU-T H.222.0 | ISO/IEC 13818-1 program stream.

2.1.144 system clock reference (SCR) (system): A time stamp in the program stream from which decoder timing is derived.

2.1.145 system target decoder (STD) (system): A hypothetical reference model of a decoding process used to define the semantics of a Rec. ITU-T H.222.0 | ISO/IEC 13818-1 multiplexed bit stream.

2.1.146 time-stamp (system): A term that indicates the time of a specific action such as the arrival of a byte or the presentation of a Presentation Unit.

2.1.147 transport stream packet header (system): The leading fields in a transport stream packet, up to and including the continuity_counter field.

2.1.148 variable bitrate: An attribute of transport streams or program streams wherein the rate of arrival of bytes at the input to a decoder varies with time.

2.1.149 video sub-bitstream: A video sub-bitstream is defined to be all VCL NAL units associated with the same value of dependency_id of an AVC video stream which conforms to one or more profiles defined in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10 and all associated non-VCL NAL units in decoding order as defined in Rec. ITU-T H.264 | ISO/IEC 14496-10. Re-assembling video sub-bitstreams in a consecutive order of dependency_id, starting from the dependency_id equal to 0 up to any value of dependency_id, results in an AVC video stream. A video sub-bitstream shall have the AVC byte stream format as defined in Annex B of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.150 view order index: An index that indicates the decoding order of MVC view components in an AVC access unit as defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 or MVCD view components in an AVC access unit as defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10. The association of view order index values to the NAL unit header syntax element view_id is indicated for an AVC video sequence in the sequence parameter set MVC extension as

defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 or in the sequence parameter set MVCD extension as defined in Annex I of Rec. ITU-T H.264 | ISO/IEC 14496-10.

2.1.151 VVC video stream: A byte stream as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3 Annex B.

2.1.152 VVC access unit: An access unit as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3 with the constraints specified in 2.23.1.

2.1.153 VVC 24-hour picture (system): A VVC access unit with a presentation time that is more than 24 hours in the future. For the purpose of this definition, VVC access unit n has a presentation time that is more than 24 hours in the future if the difference between the initial arrival time $t_{ai}(n)$ and the DPB output time $t_{o,dpb}(n)$ is more than 24 hours.

2.1.154 VVC slice: A slice as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.155 VVC subpicture: A subpicture as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.156 VVC tile of slices: One or more consecutive VVC slices which form the coded representation of a tile as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.157 VVC still picture (system): A VVC still picture consists of a VVC access unit containing an IDR picture preceded by VPS, SPS and PPS NAL units, as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3, that carry sufficient information to correctly decode this IDR picture. Preceding a VVC still picture, there shall be another VVC still picture or an End of Sequence NAL unit terminating a preceding coded video sequence as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.158 VVC video sequence (system): coded video sequence as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.159 VVC video sub-bitstream: A subset of the NAL units of a VVC video stream in their original order.

2.1.160 VVC temporal video sub-bitstream: A VVC video sub-bitstream that contains all VCL NAL units and associated non-VCL NAL units of the temporal sublayer, as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3, associated to TemporalId equal to 0 and which may additionally contain all VCL NAL units and associated non-VCL NAL units of all temporal sublayers associated to a contiguous range of TemporalId from 1 to a value equal to or smaller than $sps_max_sublayers_minus1$ included in the referred sequence parameter set, as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.1.161 VVC temporal video subset: A VVC video sub-bitstream that contains all VCL NAL units and the associated non-VCL NAL units of one or more temporal sublayers, as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3, with each temporal sublayer not being present in the corresponding VVC temporal video sub-bitstream and TemporalId associated with each temporal sublayer forming a contiguous range of values.

NOTE – According to the constraints for the transport of VVC specified in 2.23.1, each temporal sublayer of a VVC video stream is present either in the VVC temporal video sub-bitstream or in exactly one VVC temporal video subset which are carried in a set of elementary streams that are associated by hierarchy descriptors. This prevents multiple inclusion of the same temporal sublayer and allows aggregation of the VVC temporal video sub-bitstream with associated VVC temporal video subsets according to the hierarchy descriptors as specified in 2.23.3.

2.1.162 VVC highest temporal sublayer representation: The sublayer representation of the temporal sublayer with the highest value of TemporalId, as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3, in the associated VVC temporal video sub-bitstream or VVC temporal video subset.

2.1.163 VVC complete temporal representation: A sublayer representation as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3 that contains all temporal sublayers up to the temporal sublayer with TemporalId equal to $sps_max_sublayers_minus1$ as included in the referred sequence parameter set, as specified in Rec. ITU-T H.266 | ISO/IEC 23090-3.

2.2 Symbols and abbreviations

The mathematical operators used to describe this Recommendation | International Standard are similar to those used in the C-programming language. However, integer division with truncation and rounding are specifically defined. The bitwise operators are defined assuming two's-complement representation of integers. Numbering and counting loops generally begin from 0.

2.2.1 Arithmetic operators

+	Addition
–	Subtraction (as a binary operator) or negation (as a unary operator)
++	Increment
--	Decrement

* or ×	Multiplication
^	Power
/	Integer division with truncation of the result toward 0. For example, 7/4 and −7/−4 are truncated to 1 and −7/4 and 7/−4 are truncated to −1.
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from 0 unless otherwise specified. For example 3//2 is rounded to 2, and −3//2 is rounded to −2.
DIV	Integer division with truncation of the result towards −∞.
%	Modulus operator. Defined only for positive numbers.
Sign()	$\text{Sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$
NINT()	Nearest integer operator. Returns the nearest integer value to the real-valued argument. Half-integer values are rounded away from 0.
sin	Sine
cos	Cosine
exp	Exponential
√	Square root
log ₁₀	Logarithm to base ten
log _e	Logarithm to base e

2.2.2 Assignment

= Assignment operator

2.2.3 Bitwise operators

& AND
 | OR
 >> Shift right with sign extension
 << Shift left with 0 fill

2.2.4 Constants

π 3.14159265359
 e 2.71828182845

2.2.5 Logical operators

|| Logical OR
 && Logical AND
 ! Logical NOT

2.2.6 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bit-stream.

bslbf Bit string, left bit first, where "left" is the order in which bit strings are written in this Recommendation | International Standard. Bit strings are written as a string of 1s and 0s within single quote marks, e.g., '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.

NOTE – In some tables of this specification, two bit strings are used to define a range of values. In other tables, bit strings are used to specify integer values. In these cases, each bit string is equivalent to a binary number of which the least significant bit (unity position) equals the rightmost bit of the bit string and the significance of bits increases from right to left.

ch Channel
 gr Granule of 3 * 32 sub-band samples in audio Layer II, 18 * 32 sub-band samples in audio Layer III.
 main_data The main_data portion of the bit stream contains the scale factors, Huffman encoded data, and ancillary information.

main_data_beg	This gives the location in the bit stream of the beginning of the main_data for the frame. The location is equal to the ending location of the previous frame's main_data plus 1 bit. It is calculated from the main_data_end value of the previous frame.
part2_length	This value contains the number of main_data bits used for scale factors.
rpchof	Remainder polynomial coefficients, highest order first
sb	Sub-band
scfsi	Scalefactor selector information
switch_point_l	Number of scalefactor band (long block scalefactor band) from which point on window switching is used
switch_point_s	Number of scalefactor band (short block scalefactor band) from which point on window switching is used
tcimbsf	Two's complement integer, msb (sign) bit first
uimbsf	Unsigned integer, most significant bit first
vlclbf	Variable length code, left bit first, where "left" refers to the order in which the variable length codes are written
window	Number of actual time slot in case of block_type = 2, $0 \leq \text{window} \leq 2$.

The byte order of multi-byte words is most significant byte first.

2.2.7 Range operator

.. Range operator. n .. m defines the inclusive range of numbers from n up to m

2.2.8 Relational operators

> Greater than

≥ Greater than or equal to

< Less than

≤ Less than or equal to

= = Equal to

!= Not equal to

max [,...] The maximum value in the argument list

min [,...] The minimum value in the argument list

2.3 Method of describing bit stream syntax

The bit streams retrieved by the decoder are described in 2.4.1 and 2.5.1. Each data item in the bit stream is in bold type. It is described by its name, its length in bits, and a mnemonic for its type and order of transmission.

The action caused by a decoded data element in a bit stream depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and definition of the state variables used in their decoding are described in the clauses containing the semantic description of the syntax. The following constructs are used to express the conditions when data elements are present, and are in normal type.

Note this syntax uses the "C"-code convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true:

```

while ( condition ) {           If the condition is true, then the group of data elements occurs next in the data stream. This
    data_element               repeats until the condition is not true.
    ...
}

do {                            The data element always occurs at least once. The data element is repeated until the condition
    data_element               is not true.
    ...
}
while ( condition )

```

if (condition) { data_element ... }	If the condition is true, then the first group of data elements occurs next in the data stream.
else { data_element ... }	If the condition is not true, then the second group of data elements occurs next in the data stream.
for (i = 0; i < n; i++) { data_element ... }	The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to 1 for the second occurrence, and so forth.

As noted, the group of data elements may contain nested conditional constructs. For compactness, the {} are omitted when only one data element follows:

data_element []	<code>data_element []</code> is an array of data. The number of data elements is indicated by the context.
data_element [n]	<code>data_element [n]</code> is the n+1th element of an array of data.
data_element [m][n]	<code>data_element [m][n]</code> is the m+1,n+1th element of a two-dimensional array of data.
data_element [l][m][n]	<code>data_element [l][m][n]</code> is the l+1,m+1,n+1th element of a three-dimensional array of data.
data_element [m..n]	is the inclusive range of bits between bit m and bit n in the <code>data_element</code> .

While the syntax is expressed in procedural terms, it should not be assumed that either Figure 2-1 or Figure 2-2 implements a satisfactory decoding procedure. In particular, they define a correct and error-free input bitstream. Actual decoders must include a means to look for start codes and sync bytes (transport stream) in order to begin decoding correctly, and to identify errors, erasures or insertions while decoding. The methods to identify these situations, and the actions to be taken, are not standardized.

2.4 Transport stream bitstream requirements

2.4.1 Transport stream coding structure and parameters

The Rec. ITU-T H.222.0 | ISO/IEC 13818-1 transport stream coding layer allows one or more programs to be combined into a single stream. Data from each elementary stream are multiplexed together with information that allows synchronized presentation of the elementary streams within a program.

A transport stream consists of one or more programs. Audio and video elementary streams consist of access units.

Elementary stream data is carried in PES packets. A PES packet consists of a PES packet header followed by packet data. PES packets are inserted into transport stream packets. The first byte of each PES packet header is located at the first available payload location of a transport stream packet.

The PES packet header begins with a 32-bit start-code that also identifies the stream or stream type to which the packet data belongs. The PES packet header may contain decoding and presentation time stamps (DTS and PTS). The PES packet header also contains other optional fields. The PES packet data field contains a variable number of contiguous bytes from one elementary stream.

Transport stream packets begin with a 4-byte prefix, which contains a 13-bit packet ID (PID), defined in Table 2-2. The PID identifies, via the program-specific information (PSI) tables, the contents of the data contained in the transport stream packet. Transport stream packets of one PID value carry data of one and only one elementary stream.

The PSI tables are carried in the transport stream. There are six PSI tables:

- Program association table (PAT);
- Program map table (PMT);
- Conditional access table (CAT);
- Network information table (NIT);
- Transport stream description table (TSDT);
- IPMP control information table.

These tables contain the necessary and sufficient information to demultiplex and present programs. The program map table in Table 2-33 specifies, among other information, which PIDs, and therefore which elementary streams, are associated to form each program. This table also indicates the PID of the transport stream packets which carry the PCR for each program. The conditional access table shall be present if scrambling is employed. The network information table is optional and its contents are not specified by this Recommendation | International Standard. The IPMP control information table shall be present if IPMP as described in ISO/IEC 13818-11 is used by any of the components in the Rec. ITU-T H.222.0 | ISO/IEC 13818-1 stream.

Transport stream packets may be null packets. Null packets are intended for padding of transport streams. They may be inserted or deleted by re-multiplexing processes and, therefore, the delivery of the payload of null packets to the decoder cannot be assumed.

This Recommendation | International Standard does not specify the coded data which may be used as part of conditional access systems. This Specification does, however, provide mechanisms for program service providers to transport and identify this data for decoder processing, and to reference correctly data which are specified by this Specification. This type of support is provided both through transport stream packet structures and in the conditional access table (refer to Table 2-32).

2.4.2 Transport stream system target decoder

2.4.2.1 General

The semantics of the transport stream specified in 2.4.3 and the constraints on these semantics specified in 2.7 require exact definitions of byte arrival and decoding events and the times at which these occur. The definitions needed are set out in this Recommendation | International Standard using a hypothetical decoder known as the transport stream system target decoder (T-STD). Informative Annex D contains further explanation of the T-STD.

The T-STD is a conceptual model used to define these terms precisely and to model the decoding process during the construction or verification of transport streams. The T-STD is defined only for this purpose. There are three types of decoders in the T-STD: video, audio, and systems. Figure 2-1 illustrates an example. Neither the architecture of the T-STD nor the timing described precludes uninterrupted, synchronized play-back of transport streams from a variety of decoders with different architectures or timing schedules.

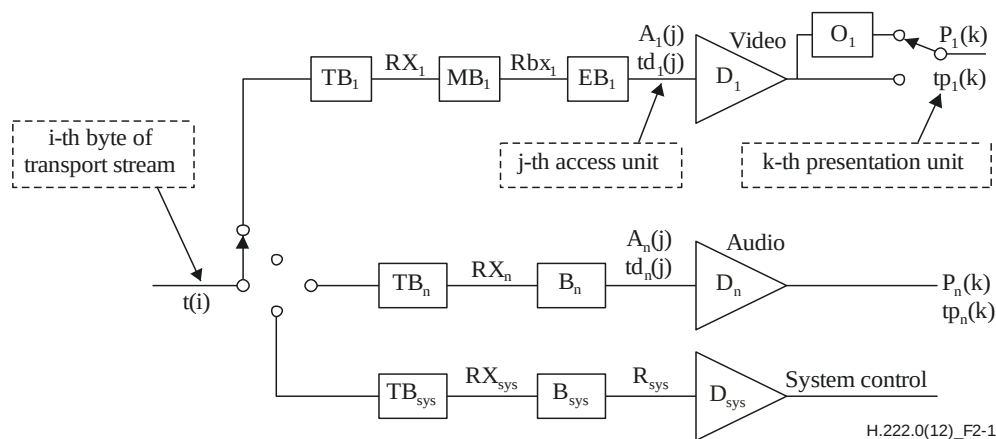


Figure 2-1 – Transport stream system target decoder notation

The following notation is used to describe the transport stream system target decoder and is partially illustrated in Figure 2-1.

- i, i', i'' are indices to bytes in the transport stream. The first byte has index 0.
- j is an index to access units in the elementary streams.
- k, k', k'' are indices to presentation units in the elementary streams.
- n is an index to the elementary streams.
- p is an index to transport stream packets in the transport stream.
- $t(i)$ indicates the time in seconds at which the i -th byte of the transport stream enters the system target decoder. The value $t(0)$ is an arbitrary constant.
- PCR(i) is the time encoded in the PCR field measured in units of the period of the 27-MHz system clock where i is the byte index of the final byte of the program_clock_reference_base field.

$A_n(j)$	is the j -th access unit in elementary stream n . $A_n(j)$ is indexed in decoding order.
$td_n(j)$	is the decoding time, measured in seconds, in the system target decoder of the j -th access unit in elementary stream n .
$P_n(k)$	is the k -th presentation unit in elementary stream n . $P_n(k)$ results from decoding $A_n(j)$. $P_n(k)$ is indexed in presentation order.
$tp_n(k)$	is the presentation time, measured in seconds, in the system target decoder of the k -th presentation unit in elementary stream n .
t	is time measured in seconds.
$F_n(t)$	is the fullness, measured in bytes, of the system target decoder input buffer for elementary stream n at time t .
B_n	is the main buffer for elementary stream n . It is present only for audio elementary streams.
BS_n	is the size of buffer, B_n , measured in bytes.
B_{sys}	is the main buffer in the system target decoder for system information for the program that is in the process of being decoded.
BS_{sys}	is the size of B_{sys} , measured in bytes.
MB_n	is the multiplexing buffer, for elementary stream n . It is present only for video elementary streams.
MBS_n	is the size of MB_n , measured in bytes.
EB_n	is the elementary stream buffer for elementary stream n . It is present only for video elementary streams.
EBS_n	is the size of the elementary stream buffer EB_n , measured in bytes.
TB_{sys}	is the transport buffer for system information for the program that is in the process of being decoded.
TBS_{sys}	is the size of TB_{sys} , measured in bytes.
TB_n	is the transport buffer for elementary stream n .
TBS_n	is the size of TB_n , measured in bytes.
D_{sys}	is the decoder for system information in program stream n .
D_n	is the decoder for elementary stream n .
O_n	is the re-order buffer for video elementary stream n .
R_{sys}	is the rate at which data are removed from B_{sys} .
Rx_n	is the rate at which data are removed from TB_n .
Rbx_n	is the rate at which PES packet payload data are removed from MB_n when the leak method is used. Defined only for video elementary streams.
$Rbx_n(j)$	is the rate at which PES packet payload data are removed from MB_n when the <code>vbv_delay</code> method is used. Defined only for video elementary streams.
Rx_{sys}	is the rate at which data are removed from TB_{sys} .
R_{es}	is the video elementary stream rate coded in a sequence header.

2.4.2.2 System clock frequency

Timing information referenced in the T-STD is carried by several data fields defined in this Specification. Refer to 2.4.3.4 and 2.4.3.6. In PCR fields this information is coded as the sampled value of a program's system clock. The PCR fields are carried in the adaptation field of the transport stream packets with a PID value equal to the PCR_PID defined in the TS_program_map_section of the program being decoded.

Practical decoders may reconstruct this clock from these values and their respective arrival times. The following are minimum constraints which apply to the program's system clock frequency as represented by the values of the PCR fields when they are received by a decoder.

The value of the system clock frequency is measured in Hz and shall meet the following constraints:

$$27\,000\,000 - 810 \leq \text{system_clock_frequency} \leq 27\,000\,000 + 810$$

$$\text{rate of change of system_clock_frequency with time} \leq 75 \times 10^{-3} \text{ Hz/s}$$

NOTE – Sources of coded data should follow a tighter tolerance in order to facilitate compliant operation of consumer recorders and playback equipment.

A program's `system_clock_frequency` may be more accurate than required. Such improved accuracy may be transmitted to the decoder via the system clock descriptor described in 2.6.20.

Bit rates defined in this Specification are measured in terms of `system_clock_frequency`. For example, a bit rate of 27 000 000 bits per second in the T-STD would indicate that one byte of data is transferred every eight (8) cycles of the system clock.

The notation "`system_clock_frequency`" is used in several places in this Specification to refer to the frequency of a clock meeting these requirements. For notational convenience, equations in which PCR, PTS, or DTS appear, lead to values of time which are accurate to some integral multiple of $(300 \times 2^{33}/\text{system_clock_frequency})$ seconds. This is due to the encoding of PCR timing information as 33 bits of $1/300$ of the system clock frequency plus 9 bits for the remainder, and encoding as 33 bits of the system clock frequency divided by 300 for PTS and DTS.

2.4.2.3 Input to the transport stream system target decoder

Input to the transport stream system target decoder (T-STD) is a transport stream. A transport stream may contain multiple programs with independent time bases. However, the T-STD decodes only one program at a time. In the T-STD model all timing indications refer to the time base of that program.

Data from the transport stream enters the T-STD at a piecewise constant rate. The time $t(i)$ at which the i -th byte enters the T-STD is defined by decoding the program clock reference (PCR) fields in the input stream, encoded in the transport stream packet adaptation field of the program to be decoded and by counting the bytes in the complete transport stream between successive PCRs of that program. The PCR field (see equation 2-1) is encoded in two parts: one, in units of the period of $1/300$ times the system clock frequency, called `program_clock_reference_base` (see equation 2-2), and one in units of the system clock frequency called `program_clock_reference_extension` (see equation 2-3). The values encoded in these are computed by $\text{PCR_base}(i)$ (see equation 2-2) and $\text{PCR_ext}(i)$ (see equation 2-3) respectively. The value encoded in the PCR field indicates the time $t(i)$, where i is the index of the byte containing the last bit of the `program_clock_reference_base` field.

Specifically:

$$\text{PCR}(i) = \text{PCR_base}(i) \times 300 + \text{PCR_ext}(i) \quad (2-1)$$

where:

$$\text{PCR_base}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 300) \% 2^{33} \quad (2-2)$$

$$\text{PCR_ext}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 1) \% 300 \quad (2-3)$$

For all other bytes the input arrival time, $t(i)$ shown in equation 2-4 below, is computed from $\text{PCR}(i'')$ and the transport rate at which data arrive, where the transport rate is determined as the number of bytes in the transport stream between the bytes containing the last bit of two successive `program_clock_reference_base` fields of the same program divided by the difference between the time values encoded in these same two PCR fields.

$$t(i) = \frac{\text{PCR}(i'')}{\text{system_clock_frequency}} + \frac{i - i''}{\text{transport_rate}(i)} \quad (2-4)$$

where:

i is the index of any byte in the transport stream for $i'' < i < i'$.

i'' is the index of the byte containing the last bit of the most recent `program_clock_reference_base` field applicable to the program being decoded.

$\text{PCR}(i'')$ is the time encoded in the program clock reference base and extension fields in units of the system clock.

The transport rate for any byte i between byte i'' and byte i' is given by:

$$\text{transport_rate}(i) = \frac{((i' - i'') \times \text{system_clock_frequency})}{\text{PCR}(i') - \text{PCR}(i'')} \quad (2-5)$$

where:

i' is the index of the byte containing the last bit of the immediately following `program_clock_reference_base` field applicable to the program being decoded.

In the case of a time-base discontinuity, indicated by the `discontinuity_indicator` in the transport packet adaptation field, the definition given in equation 2-4 and equation 2-5 for the time of arrival of bytes at the input to the T-STD is not applicable between the last PCR of the old time base and the first PCR of the new time base. In this case the time of arrival of these bytes is determined according to equation 2-4 with the modification that the transport rate used is that applicable between the last and next to last PCR of the old time base.

A tolerance is specified for the PCR values. The PCR tolerance is defined as the maximum inaccuracy allowed in received PCRs. This inaccuracy may be due to imprecision in the PCR values or to PCR modification during re-multiplexing. It does not include errors in packet arrival time due to network jitter or other causes. The PCR tolerance is ± 500 ns.

In the T-STD model, the inaccuracy will be reflected as an inaccuracy in the calculated transport rate using equation 2-5.

Transport streams with multiple programs and variable rate

Transport streams may contain multiple programs which have independent time bases. Separate sets of PCRs, as indicated by the respective `PCR_PID` values, are required for each such independent program, and therefore the PCRs cannot be co-located. The transport stream rate is piecewise constant for the program entering the T-STD. Therefore, if the transport stream rate is variable it can only vary at the PCRs of the program under consideration. Since the PCRs, and therefore the points in the transport Stream where the rate varies, are not co-located, the rate at which the transport stream enters the T-STD would have to differ depending on which program is entering the T-STD. Therefore, it is not possible to construct a consistent T-STD delivery schedule for an entire transport stream when that transport stream contains multiple programs with independent time bases and the rate of the transport stream is variable. It is straightforward, however, to construct constant bit rate transport streams with multiple variable rate programs.

2.4.2.4 Buffering

Complete transport stream packets containing system information, for the program selected for decoding, enter the system transport buffer, TB_{sys} , at the transport stream rate. These include transport stream packets whose PID values are 0, 1, 2 or 3, and all transport stream packets identified via the program association table (see Table 2-30) as having the `program_map_PID` value for the selected program. Network information table (NIT) data as specified by the NIT PID is not transferred to TB_{sys} .

NOTE 1 – Size of IPMP control information table could be large, and the repetition rate of this table should be adjusted to meet the buffer requirement.

All bytes that enter the buffer TB_n are removed at the rate R_{x_n} specified below. Bytes which are part of the PES packet header or its contents are delivered to the main buffer B_n for audio elementary streams and system data, and to the multiplexing buffer MB_n for video elementary streams. Other bytes are not, and may be used to control the system. Duplicate transport stream packets are not delivered to B_n , MB_n , or B_{sys} .

The buffer TB_n is emptied as follows:

- When there is no data in TB_n , R_{x_n} is equal to zero.
- Otherwise for video:

$$R_{x_n} = 1.2 \times R_{\max}[\text{profile, level}]$$

where:

$R_{\max}[\text{profile, level}]$ is specified according to the profile and level which can be found in Table 8-13 of Rec. ITU-T H.262 | ISO/IEC 13818-2. This table specifies the upper bound of the rate of each elementary video stream within a specific profile and level.

R_{x_n} is equal to $1.2 \times R_{\max}$ for ISO/IEC 11172-2 constrained parameter video streams, where $R_{\max}$ refers to the maximum bitrate for a constrained parameters bitstream in ISO/IEC 11172-2.

For ISO/IEC 13818-7 ADTS audio:

Number of Channels	R_{x_n} [bit/s]
1 .. 2	2 000 000
3 .. 8	5 529 600
9 .. 12	8 294 400
13 .. 48	33 177 600

Channels: The number of full-bandwidth audio output channels plus the number of independently switched coupling channel elements within the same elementary audio stream. For example, in the typical case that there are no independently switched coupling channel elements, mono is 1 channel, stereo is 2 channels and 5.1 channel surround is five channels (the low-frequency effects (LFE) channel is not counted).

For other audio,

$$R_{x_n} = 2 \times 10^6 \text{ bits per second}$$

For systems data:

$$R_{x_n} = 1 \times 10^6 \text{ bits per second}$$

R_{x_n} is measured with respect to the system clock frequency.

Complete transport stream packets containing system information, for the program selected for decoding, enter the system transport buffer, TB_{sys} , at the transport stream rate. These include transport stream packets whose PID values are 0, 1, 2 and 3 (if present), and all transport stream packets identified via the program association table (PAT) (see Table 2-30) as having the program_map_PID value for the selected program. Network information table (NIT) data as specified by the NIT PID is not transferred to TB_{sys} .

Bytes are removed from TB_{sys} at the rate $R_{x_{sys}}$ and delivered to B_{sys} . Each byte is transferred instantaneously.

Duplicate transport stream packets are not delivered to B_{sys} .

Transport packets which do not enter any TB_n or TB_{sys} are discarded.

The transport buffer size is fixed at 512 bytes.

The elementary stream buffer sizes EBS_1 through EBS_n are defined for video as equal to the `vbv_buffer_size` as it is carried in the sequence header. Refer to the summary of constrained parameters in ISO/IEC 11172-2 and Table 8-14 of Rec. ITU-T H.262 | ISO/IEC 13818-2.

NOTE 2 – In the following equations, unit conversion should be implicitly performed as appropriate. Values expressed in bits are implicitly converted into values expressed in bytes by: $number_of_bytes = (number_of_bits + 7) / 8$.

The multiplexing buffer size MBS_1 through MBS_n are defined for video as follows:

For Low and Main level:

$$MBS_n = BS_{mux} + BS_{oh} + VB_{V_{max}}[profile, level] - vbv_buffer_size$$

where BS_{oh} , PES packet overhead buffering is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{max}[profile, level]$$

and BS_{mux} , additional multiplex buffering is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max}[profile, level]$$

and where $VB_{V_{max}}[profile, level]$ is defined in Table 8-14 of Rec. ITU-T H.262 | ISO/IEC 13818-2 and $R_{max}[profile, level]$ is defined in Table 8-13 of Rec. ITU-T H.262 | ISO/IEC 13818-2, and `vbv buffer size` is carried in the sequence header described in 6.2.2 of Rec. ITU-T H.262 | ISO/IEC 13818-2.

For High 1440 and High level:

$$MBS_n = BS_{mux} + BS_{oh}$$

where BS_{oh} is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{max}[profile, level]$$

and BS_{mux} is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{max}[profile, level]$$

and where $R_{\max}[\text{profile, level}]$ is defined in Table 8-13 of Rec. ITU-T H.262 | ISO/IEC 13818-2.

For Constrained Parameters ISO/IEC 11172-2 bitstreams:

$$MBS_n = BS_{mux} + BS_{oh} + vbv_max - vbv_buffer_size$$

where BS_{oh} is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times R_{\max}$$

and BS_{mux} is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times R_{\max}$$

and where $R_{\max}$ and vbv_max refer to the maximum bitrate and the maximum vbv_buffer_size for a Constrained Parameters bitstream in ISO/IEC 11172-2 respectively.

A portion $BS_{mux} = 4 \text{ ms} \times R_{\max}[\text{profile, level}]$ of the MBS_n is allocated for buffering to allow multiplexing. The remainder is available for BS_{oh} and may also be available for initial multiplexing.

NOTE 3 – Buffer occupancy by PES packet overhead is directly bounded in PES streams by the PES-STD which is defined in 2.5.2.4. It is possible, but not necessary, to utilize PES streams to construct transport streams.

Buffer BS_n

The main buffer sizes BS_1 through BS_n are defined as follows.

Audio

For ISO/IEC 13818-7 ADTS audio:

Number of Channels	BS_n [bytes]
1 .. 2	3 584
3 .. 8	8 976
9 .. 12	12 804
13 .. 48	51 216

Channels: The number of full-bandwidth audio output channels plus the number of independently switched coupling channel elements within the same elementary audio stream. For example, in the typical case that there are no independently switched coupling channel elements, mono is 1 channel, stereo is 2 channels, and 5.1 channel surround is 5 channels (the LFE channel is not counted).

For other audio:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ bytes}$$

The size of the access unit decoding buffer BS_{dec} , and the PES packet overhead buffer BS_{oh} are constrained by:

$$BS_{dec} + BS_{oh} \leq 2848 \text{ bytes}$$

A portion (736 bytes) of the 3584 byte buffer is allocated for buffering to allow multiplexing. The rest, 2848 bytes, are shared for access unit buffering BS_{dec} , BS_{oh} and additional multiplexing.

Systems

The main buffer B_{sys} for system data is of size $BS_{sys} = 1536$ bytes.

Video

For video elementary streams, data is transferred from MB_n to EB_n using one of two methods: the leak method or the video buffering verifier (VBV) delay method.

Leak method

The leak method transfers data from MB_n to EB_n using a leak rate R_{bx} . The leak method is used whenever any of the following is true:

- the STD descriptor (refer to 2.6.32) for the elementary stream is not present in the transport stream;
- the STD descriptor is present and the leak_valid flag has a value of '1';
- the STD descriptor is present, the leak_valid has a value of '0', and the vbv_delay fields coded in the video stream have the value 0xFFFF; or
- trick mode status is true (refer to 2.4.3.7).

For Low and Main level:

$$R_{bx_n} = R_{\max} [profile, level]$$

For High-1440 and High level:

$$R_{bx_n} = \text{Min}\{1.05 \times R_{es}, R_{\max}[profile, level]\}$$

For Constrained Parameters bitstream in ISO/IEC 11172-2:

$$R_{bx_n} = 1.2 \times R_{\max}$$

where $R_{\max}$ is the maximum bit rate for a Constrained Parameters bitstream in ISO/IEC 11172-2.

If there is PES packet payload data in MB_n , and buffer EB_n is not full, the PES packet payload is transferred from MB_n to EB_n at a rate equal to R_{bx_n} . If EB_n is full, data are not removed from MB_n . When a byte of data is transferred from MB_n to EB_n , all PES packet header bytes that are in MB_n and immediately precede that byte, are instantaneously removed and discarded. When there is no PES packet payload data present in MB_n , no data is removed from MB_n . All data that enters MB_n leaves it. All PES packet payload data bytes enter EB_n instantaneously upon leaving MB_n .

Vbv_delay method

The vbv_delay method specifies precisely the time at which each byte of coded video data is transferred from MB_n to EB_n , using the vbv_delay values coded in the video elementary stream. The vbv_delay method is used whenever the STD descriptor (refer to 2.6.32) for this elementary stream is present in the transport stream, the leak_valid flag in the descriptor has the value '0', and vbv_delay fields coded in the video stream are not equal to 0xFFFF. If any vbv_delay values in a video sequence are not equal to 0xFFFF, none of the vbv_delay fields in that sequence shall be equal to 0xFFFF (refer to ISO/IEC 11172-2 and Rec. ITU-T H.262 | ISO/IEC 13818-2).

When the vbv_delay method is used, the final byte of the video picture start code for picture j is transferred from MB_n to the EB_n at the time $td_n(j) - vbv_delay(j)$, where $td_n(j)$ is the decoding time of picture j , as defined above, and $vbv_delay(j)$ is the delay time, in seconds, indicated by the vbv_delay field of picture j . The transfer of bytes between the final bytes of successive picture start codes (including the final byte of the second start code), into the buffer EB_n , is at a piecewise constant rate, $R_{bx}(j)$, which is specified for each picture j . Specifically, the rate, $R_{bx}(j)$, of transfer into this buffer is given by:

$$R_{bx}(j) = NB(j) / (vbv_delay(j) - vbv_delay(j+1) + td_n(j+1) - td_n(j)) \quad (2-6)$$

where $NB(j)$ is the number of bytes between the final bytes of the picture start codes (including the final byte of the second start code) of pictures j and $j+1$, excluding PES packet header bytes.

NOTE 4 – $vbv_delay(j+1)$ and $td_n(j+1)$ may have values that differ from those normally expected for periodic video display if the low_delay flag in the video sequence extension is set to '1'. It may not be possible to determine the correct values by examination of the bit stream.

The $R_{bx}(j)$ derived from equation 2-6 shall be less than or equal to $R_{\max}[profile, level]$ for elementary streams of stream type 0x02 (refer to Table 2-34), where $R_{\max}[profile, level]$ is defined in Rec. ITU-T H.262 | ISO/IEC 13818-2, and shall be less than or equal to the maximum bit rate allowed for constrained parameter video elementary streams of stream type 0x01, refer to ISO/IEC 11172-2.

When a byte of data is transferred from MB_n to EB_n , all PES packet header bytes that are in MB_n and immediately precede that byte are instantaneously removed and discarded. All data that enters MB_n leaves it. All PES packet payload data bytes enter EB_n instantaneously upon leaving MB_n .

Removal of access units

For each elementary stream buffer EB_n and main buffer B_n all data for the access unit that has been in the buffer longest, $A_n(j)$, and any stuffing bytes that immediately precede it that are present in the buffer at the time $td_n(j)$ are removed instantaneously at time $td_n(j)$. The decoding time $td_n(j)$ is specified in the DTS or PTS fields (refer to 2.4.3.6). Decoding times $td_n(j+1)$, $td_n(j+2)$, ... of access units without encoded DTS or PTS fields which directly follow access unit j may be derived from information in the elementary stream. Refer to Annex C of Rec. ITU-T H.262 | ISO/IEC 13818-2, ISO/IEC 13818-3, or ISO/IEC 11172. Also refer to 2.7.5. In the case of audio, all PES packet headers that are stored immediately before the access unit or that are embedded within the data of the access unit are removed simultaneously with the removal of the access unit. As the access unit is removed it is instantaneously decoded to a presentation unit.

System data

In the case of system data, data is removed from the main buffer B_{sys} at a rate of R_{sys} whenever there is at least 1 byte available in buffer B_{sys} .

$$R_{sys} = \max(80\,000 \text{ bits/s}, \text{transport_rate}(i) \times 8 \text{ bits/byte} / 500) \quad (2-7)$$

NOTE 5 – The intention of increasing R_{sys} in the case of high transport rates is to allow an increased data rate for the Program-specific information.

Low delay

When the `low_delay` flag in the video sequence extension is set to '1' (see 6.2.2.3 of Rec. ITU-T H.262 | ISO/IEC 13818-2) the EB_n buffer may underflow. In this case, when the T-STD elementary stream buffer EB_n is examined at the time specified by $td_n(j)$, the complete data for the access unit may not be present in the buffer EB_n . When this case arises, the buffer shall be re-examined at intervals of two field-periods until the data for the complete access unit is present in the buffer. At this time the entire access unit shall be removed from buffer EB_n instantaneously. Overflow of buffer EB_n shall not occur.

When the `low_delay_mode` flag is set to '1', EB_n underflow is allowed to occur continuously without limit. The T-STD decoder shall remove access unit data from buffer EB_n at the earliest time consistent with the paragraph above and any DTS or PTS values encoded in the bit stream. Note that the decoder may be unable to re-establish correct decoding and display times as indicated by DTS and PTS until the EB_n buffer underflow situation ceases and a PTS or DTS is found in the bit stream.

Trick mode

When the `DSM_trick_mode` flag (2.4.3.6) is set to '1' in the PES Packet header of a packet containing the start of a B-type video access unit and the `trick_mode_control` field is set to '001' (slow motion) or '010' (freeze frame), or '100' (slow reverse) the B-picture access unit is not removed from the video data buffer EB_n until the last time of possibly multiple times that any field of the picture is decoded and presented. Repetition of the presentation of fields and pictures is defined in 2.4.3.8 under slow motion, slow reverse, and `field_id_cntrl`. The access unit is removed instantaneously from EB_n at the indicated time, which is dependent on the value of `rep_cntrl`.

When the `DSM_trick_mode` flag is set to '1' in the PES packet header of a packet containing the first byte of a picture start code, `trick_mode` status becomes true when that picture start code in the PES packet is removed from buffer EB_n . Trick mode status remains true until a PES packet header is received by the T-STD in which the `DSM_trick_mode` flag is set to '0' and the first byte of the picture start code after that PES packet header is removed from buffer EB_n . When trick mode status is true, the buffer EB_n may underflow. All other constraints from normal streams are retained when trick mode status is true.

2.4.2.5 Decoding

Elementary streams buffered in B_1 through B_n and EB_1 through EB_n are decoded instantaneously by decoders D_1 through D_n and may be delayed in re-order buffers O_1 through O_n before being presented at the output of the T-STD. Re-order buffers are used only in the case of a video elementary stream when some access units are not carried in presentation order. These access units will need to be re-ordered before presentation. In particular, if $P_n(k)$ is an I-picture or a P-picture carried before one or more B-pictures, then it must be delayed in the re-order buffer, O_n , of the T-STD before being presented. Any picture previously stored in O_n is presented before the current picture can be stored. $P_n(k)$ should be delayed until the next I-picture or P-picture is decoded. While it is stored in the re-order buffer, the subsequent B-pictures are decoded and presented.

The time at which a presentation unit $P_n(k)$ is presented is $tp_n(k)$. For presentation units that do not require re-ordering delay, $tp_n(k)$ is equal to $td_n(j)$ since the access units are decoded instantaneously; this is the case, for example, for B-frames. For presentation units that are delayed, $tp_n(k)$ and $td_n(j)$ differ by the time that $P_n(k)$ is delayed in the re-order buffer, which is a multiple of the nominal picture period. Care should be taken to use adequate re-ordering delay from the

beginning of video elementary streams to meet the requirements of the entire stream. For example, a stream which initially has only I- and P-pictures but later includes B-pictures should include re-ordering delay starting at the beginning of the stream.

Rec. ITU-T H.262 | ISO/IEC 13818-2 explains re-ordering of video pictures in greater detail.

2.4.2.6 Presentation

The function of a decoding system is to reconstruct presentation units from compressed data and to present them in a synchronized sequence at the correct presentation times. Although real audio and visual presentation devices generally have finite and different delays and may have additional delays imposed by post-processing or output functions, the system target decoder models these delays as zero.

In the T-STD in Figure 2-1 the display of a video presentation unit (a picture) occurs instantaneously at its presentation time, $tp_n(k)$.

In the T-STD the output of an audio presentation unit starts at its presentation time, $tp_n(k)$, when the decoder instantaneously presents the first sample. Subsequent samples in the presentation unit are presented in sequence at the audio sampling rate.

2.4.2.7 Buffer management

Transport streams shall be constructed so that conditions defined in this subclause are satisfied. This subclause makes use of the notation defined for the system target decoder.

TB_n and TB_{sys} shall not overflow. TB_n and TB_{sys} shall empty at least once every second. B_n shall not overflow nor underflow. B_{sys} shall not overflow.

EB_n shall not underflow except when the low delay flag in the video sequence extension is set to '1' (refer to 6.2.2.3 in Rec. ITU-T H.262 | ISO/IEC 13818-2) or `trick_mode` status is true.

When the leak method for specifying transfers is in effect, MB_n shall not overflow, and shall empty at least once every second. EB_n shall not overflow.

When the `vbv_delay` method for specifying transfers is in effect, MB_n shall not overflow nor underflow, and EB_n shall not overflow.

The delay of any data through the system target decoder buffers shall be less than or equal to one second except for still picture video data, ISO/IEC 14496, ISO/IEC 23008-2, ISO/IEC 23090-3, ISO/IEC 23094-1 and ISO/IEC 23094-2 streams. Specifically: $td_n(j) - t(i) \leq 1$ second for all j , and all bytes i in access unit $A_n(j)$.

For still picture video data, the delay is constrained by $td_n(j) - t(i) \leq 60$ seconds for all j , and all bytes i in access unit $A_n(j)$.

For ISO/IEC 14496, ISO/IEC 23008-2, ISO/IEC 23090-3, ISO/IEC 23094-1 and ISO/IEC 23094-2 streams, the delay is constrained by $td_n(j) - t(i) \leq 10$ seconds for all j , and all bytes i in access unit $A_n(j)$.

Definition of overflow and underflow

Let $F_n(t)$ be the instantaneous fullness of T-STD buffer B_n .

$F_n(t) = 0$ instantaneously before $t = t(0)$

Overflow does not occur if:

$$F_n(t) \leq BS_n$$

for all t and n .

Underflow does not occur if:

$$0 \leq F_n(t)$$

for all t and n .

2.4.2.8 T-STD extensions for carriage of ISO/IEC 14496 data

For decoding of ISO/IEC 14496 data carried in a transport stream the T-STD model is extended. T-STD parameters for decoding of individual ISO/IEC 14496 elementary streams are defined in 2.11.2, while 2.11.3 defines T-STD extensions and parameters for decoding of ISO/IEC 14496 scenes and associated streams.

2.4.2.9 T-STD extensions for carriage of Rec. ITU-T H.264 | ISO/IEC 14496-10 video

To define the decoding in the T-STD of Rec. ITU-T H.264 | ISO/IEC 14496-10 video streams carried in a transport stream, the T-STD model needs to be extended. The T-STD extension and T-STD parameters for decoding of AVC video streams conforming to one or more profiles defined in Annex A of Rec. ITU-T H.264 | ISO/IEC 14496-10 video streams are defined in 2.14.3.1, T-STD extension and T-STD parameters for decoding of AVC video streams conforming to one or more profiles defined in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10 are defined in 2.14.3.5, and T-STD extension and T-STD parameters for decoding of AVC video streams conforming to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 are defined in 2.14.3.7.

2.4.2.10 T-STD extensions for carriage of ISO/IEC 14496-17 text streams

To define the decoding in the T-STD of ISO/IEC 14496-17 text streams carried in a transport stream, the T-STD model needs to be extended. The T-STD extension and T-STD parameters for decoding of ISO/IEC 14496-17 text streams are defined in 2.15.3.1.

2.4.2.11 T-STD extensions for carriage of J2K video elementary streams

The interpretation, extensions, use and constraints for syntax elements in the adaptation header (2.4.3.4 and 2.4.3.5) for JPEG 2000 part 1 video are defined in S.5.

The interpretation, extensions, use and constraints for syntax elements in the PES header (2.4.3.6 and 2.4.3.7) for JPEG 2000 Part 1 video are defined in S.5.

To define the decoding of J2K video elementary streams carried in a transport stream, the T-STD model needs to be extended. The T-STD extensions and T-STD parameters for decoding of J2K video elementary streams conforming to one or more profiles defined in Rec. ITU-T T.800 (2002) | ISO/IEC 15444-1:2004 are defined in S.6.

NOTE – No extensions are specified for P-STD model, as carriage of J2K video elementary streams in program streams is not supported.

2.4.2.12 T-STD extensions for carriage of HEVC

T-STD extensions and T-STD parameters for the decoding of HEVC video streams are defined in 2.17.2 and 2.17.3. Program stream support including P-STD extensions and P-STD parameters are not specified for HEVC video streams.

2.4.2.13 T-STD extensions for carriage of MVCD video sub-bitstream

T-STD extensions and T-STD parameters for decoding of MVCD video sub-bitstreams are defined in 2.14.1 and 2.14.3.7.

NOTE – Program stream extensions are not specified for MVCD video sub-bitstreams.

2.4.2.14 T-STD extensions for carriage of MV HEVC and SHVC

T-STD extensions and T-STD parameters for decoding of HEVC layered video streams are defined in 2.17.4. Program stream support including P-STD extensions and P-STD parameters are not specified for HEVC extension video streams.

2.4.2.15 T-STD extensions for carriage of JPEG XS video elementary streams

The interpretation, extensions, use and constraints for syntax elements in the adaptation header (2.4.3.4 and 2.4.3.5) for JPEG XS Part-1 video are defined in W.5.

The interpretation, extensions, use and constraints for syntax elements in the PES header (2.4.3.6 and 2.4.3.7) for JPEG XS Part-1 video are defined in W.5.

To define the decoding of JPEG XS video elementary streams carried in a Transport Stream, the T-STD model needs to be extended. The T-STD extensions and T-STD parameters for decoding of JPEG XS video elementary streams conforming to one or more profiles defined in ISO/IEC 21122-2 are defined in W.6.

NOTE – No extensions are specified for P-STD model, as carriage of JPEG XS video elementary streams in program streams is not supported.

2.4.2.16 T-STD extensions for carriage of VVC

T-STD extensions and T-STD parameters for decoding of VVC video streams are defined in 2.23.2 and 2.23.3. Program stream support including P-STD extensions and P-STD parameters are not specified for VVC video streams.

2.4.2.17 T-STD extensions for carriage of EVC

T-STD extensions and T-STD parameters for decoding of EVC video streams are defined in 2.24.2. Program stream support including P-STD extensions and P-STD parameters are not specified for EVC video streams.

2.4.3 Specification of the transport stream syntax and semantics

The following syntax describes a stream of bytes. Transport stream packets shall be 188 bytes long.

2.4.3.1 Transport stream

See Table 2-1.

Table 2-1 – Transport stream

Syntax	No. of bits	Mnemonic
<pre> MPEG_transport_stream() { do { transport_packet() } while (nextbits() == sync_byte) } </pre>		

2.4.3.2 Transport stream packet layer

See Table 2-2.

Table 2-2 – Transport packet of this Recommendation | International Standard

Syntax	No. of bits	Mnemonic
transport_packet(){		
sync_byte	8	bslbf
transport_error_indicator	1	bslbf
payload_unit_start_indicator	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
transport_scrambling_control	2	bslbf
adaptation_field_control	2	bslbf
continuity_counter	4	uimsbf
if(adaptation_field_control == '10' adaptation_field_control == '11'){		
adaptation_field()		
}		
if(adaptation_field_control == '01' adaptation_field_control == '11') {		
for (i = 0; i < N; i++){		
data_byte	8	bslbf
}		
}		
}		

2.4.3.3 Semantic definition of fields in transport stream packet layer

sync_byte – The sync_byte is a fixed 8-bit field whose value is '0100 0111' (0x47). Sync_byte emulation in the choice of values for other regularly occurring fields, such as PID, should be avoided.

transport_error_indicator – The transport_error_indicator is a 1-bit flag. When set to '1' it indicates that at least 1 uncorrectable bit error exists in the associated transport stream packet. This bit may be set to '1' by entities external to the transport layer. When set to '1' this bit shall not be reset to '0' unless the bit value(s) in error have been corrected.

payload_unit_start_indicator – The payload_unit_start_indicator is a 1-bit flag which has normative meaning for transport stream packets that carry PES packets (refer to 2.4.3.6) or transport stream section data (refer to Table 2-31 in 2.4.4.5).

When the payload of the transport stream packet contains PES packet data, the payload_unit_start_indicator has the following significance: a '1' indicates that the payload of this transport stream packet will commence with the first byte of a PES packet and a '0' indicates no PES packet shall start in this transport stream packet. If the

payload_unit_start_indicator is set to '1', then one and only one PES packet starts in this transport stream packet. This also applies to private streams of stream_type 6 (refer to Table 2-34).

When the payload of the transport stream packet contains transport stream section data, the payload_unit_start_indicator has the following significance: if the transport stream packet carries the first byte of a section, the payload_unit_start_indicator value shall be '1', indicating that the first byte of the payload of this transport stream packet carries the pointer_field. If the transport stream packet does not carry the first byte of a section, the payload_unit_start_indicator value shall be '0', indicating that there is no pointer_field in the payload. Refer to 2.4.4.2 and 2.4.4.3. This also applies to private streams of stream_type 5 (refer to Table 2-34).

For null packets the payload_unit_start_indicator shall be set to '0'.

The meaning of this bit for transport stream packets carrying only private data is not defined in this Specification.

transport_priority – The transport_priority is a 1-bit indicator. When set to '1' it indicates that the associated packet is of greater priority than other packets having the same PID which do not have the bit set to '1'. The transport mechanism can use this to prioritize its data within an elementary stream. Depending on the application the transport_priority field may be coded regardless of the PID or within one PID only. This field may be changed by channel-specific encoders or decoders.

PID – The PID is a 13-bit field, indicating the type of the data stored in the packet payload. PID value 0x0000 is reserved for the program association table (see Table 2-30). PID value 0x0001 is reserved for the conditional access table (see Table 2-32). PID value 0x0002 is reserved for the transport stream description table (see Table 2-36), PID value 0x0003 is reserved for IPMP control information table (see ISO/IEC 13818-11) and PID values 0x0004-0x000F are reserved. PID value 0x1FFF is reserved for null packets (see Table 2-3).

Table 2-3 – PID table

Value	Description
0x0000	Program association table
0x0001	Conditional access table
0x0002	Transport stream description table
0x0003	IPMP control information table
0x0004	Adaptive streaming information (see Note 2)
0x0005 .. 0x000F	Reserved
0x0010 .. 0x1FFE	May be assigned as network_PID, Program_map_PID, elementary_PID, or for other purposes
0x1FFF	Null packet
NOTE 1 – The transport packets with PID values 0x0000, 0x0001, and 0x0010-0x1FFE are allowed to carry a PCR.	
NOTE 2 – Payload syntax is defined in 5.10.3.3.5 of ISO/IEC 23009-1.	

transport_scrambling_control – This 2-bit field indicates the scrambling mode of the transport stream packet payload. The transport stream packet header, and the adaptation field when present, shall not be scrambled. In the case of a null packet the value of the transport_scrambling_control field shall be set to '00' (see Table 2-4).

Table 2-4 – Scrambling control values

Value	Description
'00'	Not scrambled
'01'	User-defined
'10'	User-defined
'11'	User-defined

adaptation_field_control – This 2-bit field indicates whether this transport stream packet header is followed by an adaptation field and/or payload (see Table 2-5).

Table 2-5 – Adaptation field control values

Value	Description
'00'	Reserved for future use by ISO/IEC
'01'	No adaptation_field, payload only
'10'	Adaptation_field only, no payload
'11'	Adaptation_field followed by payload

Rec. ITU-T H.222.0 | ISO/IEC 13818-1 decoders shall discard transport stream packets with the adaptation_field_control field set to a value of '00'. In the case of a null packet the value of the adaptation_field_control shall be set to '01'.

continuity_counter – The continuity_counter is a 4-bit field incrementing with each transport stream packet with the same PID. The continuity_counter wraps around to 0 after its maximum value. The continuity_counter shall not be incremented when the adaptation_field_control of the packet equals '00' or '10'.

In transport streams, duplicate packets may be sent as two, and only two, consecutive transport stream packets of the same PID. The duplicate packets shall have the same continuity_counter value as the original packet and the adaptation_field_control field shall be equal to '01' or '11'. In duplicate packets each byte of the original packet shall be duplicated, with the exception that in the program clock reference fields, if present, a valid value shall be encoded.

The continuity_counter in a particular transport stream packet is continuous when it differs by a positive value of one from the continuity_counter value in the previous transport stream packet of the same PID, or when either of the non-incrementing conditions (adaptation_field_control set to '00' or '10', or duplicate packets as described above) are met. The continuity counter may be discontinuous when the discontinuity_indicator is set to '1' (refer to 2.4.3.4). In the case of a null packet the value of the continuity_counter is undefined.

data_byte – Data bytes shall be contiguous bytes of data from the PES packets (refer to 2.4.3.6), transport stream sections (refer to 2.4.4), packet stuffing bytes after transport stream sections, or private data not in these structures as indicated by the PID. In the case of null packets with PID value 0x1FFF, data_bytes may be assigned any value. The number of data_bytes, N, is specified by 184 minus the number of bytes in the adaptation_field(), as described in 2.4.3.4.

2.4.3.4 Adaptation field

See Table 2-6.

Table 2-6 – Transport stream adaptation field

Syntax	No. of bits	Mnemonic
adaptation_field() {		
adaptation_field_length	8	uimsbf
if (adaptation_field_length > 0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if (PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf
reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	33	uimsbf
reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if (splicing_point_flag == '1') {		
splice_countdown	8	tcimsbf
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	8	uimsbf

Table 2-6 – Transport stream adaptation field

Syntax	No. of bits	Mnemonic
for (i = 0; i < transport_private_data_length; i++) { private_data_byte }	8	bslbf
}		
if (adaptation_field_extension_flag == '1') { adaptation_field_extension_length ltw_flag piecewise_rate_flag seamless_splice_flag af_descriptor_not_present_flag reserved if (ltw_flag == '1') { ltw_valid_flag ltw_offset } if (piecewise_rate_flag == '1') { reserved piecewise_rate } if (seamless_splice_flag == '1') { Splice_type DTS_next_AU[32..30] marker_bit DTS_next_AU[29..15] marker_bit DTS_next_AU[14..0] marker_bit } if (af_descriptor_not_present_flag == '0') { for (i = 0; i < N1; i++) { af_descriptor() } } else { for (i = 0; i < N2; i++) { reserved } } }	8 1 1 1 1 4 1 15 2 22	uimsbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf uimsbf bslbf uimsbf
if (seamless_splice_flag == '1') { Splice_type DTS_next_AU[32..30] marker_bit DTS_next_AU[29..15] marker_bit DTS_next_AU[14..0] marker_bit }	4 3 1 15 1 15 1	bslbf bslbf bslbf bslbf bslbf bslbf bslbf
if (af_descriptor_not_present_flag == '0') { for (i = 0; i < N1; i++) { af_descriptor() } }		
else { for (i = 0; i < N2; i++) { reserved } }	8	bslbf
for (i = 0; i < N3; i++) { stuffing_byte }	8	bslbf
}		

2.4.3.5 Semantic definition of fields in adaptation field

adaptation_field_length – The adaptation_field_length is an 8-bit field specifying the number of bytes in the adaptation_field immediately following the adaptation_field_length. The value '0' is for inserting a single stuffing byte in the adaptation field of a transport stream packet. When the adaptation_field_control value is '11', the value of the adaptation_field_length shall be in the range 0 to 182. When the adaptation_field_control value is '10', the value of the adaptation_field_length shall be 183. For transport stream packets carrying PES packets, stuffing is needed when there is insufficient PES packet data to completely fill the transport stream packet payload bytes. Stuffing is accomplished by defining an adaptation field longer than the sum of the lengths of the data elements in it, so that the payload bytes remaining after the adaptation field exactly accommodates the available PES packet data. The extra space in the adaptation field is filled with stuffing bytes.

This is the only method of stuffing allowed for transport stream packets carrying PES packets. For transport stream packets carrying sections, an alternative stuffing method is described in 2.4.4.1.

discontinuity_indicator – This is a 1-bit field which when set to '1' indicates that the discontinuity state is true for the current transport stream packet. When the discontinuity_indicator is set to '0' or is not present, the discontinuity state is false. The discontinuity indicator is used to indicate two types of discontinuities, system time-base discontinuities and continuity_counter discontinuities.

A system time-base discontinuity is indicated by the use of the `discontinuity_indicator` in transport stream packets of a PID designated as a PCR_PID (refer to 2.4.4.10). When the discontinuity state is true for a transport stream packet of a

PID designated as a PCR_PID, the next PCR in a transport stream packet with that same PID represents a sample of a new system time clock for the associated program. The system time-base discontinuity point is defined to be the instant in time when the first byte of a packet containing a PCR of a new system time-base arrives at the input of the T-STD. The discontinuity_indicator shall be set to '1' in the packet in which the system time-base discontinuity occurs. The discontinuity_indicator bit may also be set to '1' in transport stream packets of the same PCR_PID prior to the packet which contains the new system time-base PCR. In this case, once the discontinuity_indicator has been set to '1', it shall continue to be set to '1' in all transport stream packets of the same PCR_PID up to and including the transport stream packet which contains the first PCR of the new system time-base. After the occurrence of a system time-base discontinuity, no fewer than two PCRs for the new system time-base shall be received before another system time-base discontinuity can occur. Further, except when trick mode status is true, data from no more than two system time-bases shall be present in the set of T-STD buffers for one program at any time.

Prior to the occurrence of a system time-base discontinuity, the first byte of a transport stream packet which contains a PTS or DTS which refers to the new system time-base shall not arrive at the input of the T-STD. After the occurrence of a system time-base discontinuity, the first byte of a transport stream packet which contains a PTS or DTS which refers to the previous system time-base shall not arrive at the input of the T-STD.

A continuity_counter discontinuity is indicated by the use of the discontinuity_indicator in any transport stream packet. When the discontinuity state is true in any transport stream packet of a PID not designated as a PCR_PID, the continuity_counter in that packet may be discontinuous with respect to the previous transport stream packet of the same PID. When the discontinuity state is true in a transport stream packet of a PID that is designated as a PCR_PID, the continuity_counter may only be discontinuous in the packet in which a system time-base discontinuity occurs. A continuity counter discontinuity point occurs when the discontinuity state is true in a transport stream packet and the continuity_counter in the same packet is discontinuous with respect to the previous transport stream packet of the same PID. A continuity counter discontinuity point shall occur at most one time from the initiation of the discontinuity state until the conclusion of the discontinuity state. Furthermore, for all PIDs that are not designated as PCR_PIDs, when the discontinuity_indicator is set to '1' in a packet of a specific PID, the discontinuity_indicator may be set to '1' in the next transport stream packet of that same PID, but shall not be set to '1' in three consecutive transport stream packets of that same PID.

For the purpose of this clause, an elementary stream access point is defined as follows:

- ISO/IEC 11172-2 video and Rec. ITU-T H.262 | ISO/IEC 13818-2 video – The first byte of a video sequence header.
 - ISO/IEC 14496-2 visual – The first byte of the visual object sequence header.
 - AVC video streams conforming to one or more profiles defined in Annex A of Rec. ITU-T H.264 | ISO/IEC 14496-10 – The first byte of an AVC access unit. The SPS and PPS parameter sets referenced in this and all subsequent AVC access units in the coded video stream shall be provided after this access point in the byte stream and prior to their activation.
 - Video sub-bitstreams of AVC video streams conforming to one or more profiles defined in Annex G of Rec. ITU-T H.264 | ISO/IEC 14496-10 – The first byte of an SVC dependency representation is an elementary stream access point if the following conditions are met:
 - The subset sequence parameter sets and picture parameter sets referenced in this and all subsequent SVC dependency representation in the video sub-bitstream shall be provided after this access point in the byte stream and prior to their activation.
 - If this SVC video sub-bitstream access point requires the elementary stream access point of the same AVC access unit, if any, contained in the corresponding elementary stream that needs to be present in decoding order before decoding the elementary stream associated with this elementary stream access point, then the corresponding elementary stream shall also include an elementary stream access point.
- NOTE 1 – If the hierarchy descriptor is present for this SVC video sub-bitstream then the video sub-bitstream of which the hierarchy_layer_index equals the hierarchy_embedded_layer_index of this SVC sub-bitstream should have an elementary stream access point in the same access unit.
- MVC video sub-bitstreams of AVC video streams conforming to one or more profiles defined in Annex H of Rec. ITU-T H.264 | ISO/IEC 14496-10 – The first byte of an MVC view-component subset is an elementary stream access point if the following two conditions are met:
 - The subset sequence parameter sets and picture parameter sets referenced in this and all subsequent MVC view-component subsets in the MVC video sub-bitstream shall be provided after this access point in the byte stream and prior to their activation.
 - If this MVC video sub-bitstream access point requires the elementary stream access point of the same AVC access unit, if any, contained in the corresponding elementary stream that needs to be present in decoding order before decoding the elementary stream associated with this elementary stream