

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

Part 1: Systems

The European Standard EN ISO/IEC 13818-1:1997 has the status of a British Standard

ICS 35.040

National foreword

This British Standard has been prepared by Technical Committee IST/37 and is the English language version of EN ISO/IEC 13818 *Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems* published by the European Committee for Standardization (CEN). It is identical with ISO/IEC 13818-1:1996 published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of picture, audio, multimedia and hypermedia information, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the EN ISO/IEC title page, page 2, the ISO/IEC title page, pages ii to x, pages 1 to 122 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

Amendments issued since publication

Amd. No.	Date	Comments
9625	September 1997	Indicated by a sideline in the margin

This British Standard having been prepared under the direction of the DISC Board, was published under the authority of the Standards Board and comes into effect on 15 April 1997

© BSI 05-2000

ISBN 0 580 27230 3

Contents

	Page
National foreword	Inside front cover
Foreword	2
Foreword	ix
Text of ISO/IEC 13818-1	1

ICS 35.040

Descriptors: Coding (data conversion), data conversion, data processing, definitions, encoding, figures, image processing, information technology, sound recording, system architecture, video recording, codification, data converting

English version

Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems

(ISO/IEC 13818-1:1996)

Technologies de l'information —
Codage générique des images animées et des
informations sonores associées
Partie 1: Systèmes
(ISO/IEC 13818-1:1996)

This European Standard was approved by CEN on 1997-01-19. CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CEN

European Committee for Standardization
Comité Européen de Normalisation
Europäisches Komitee für Normung

Central Secretariat: rue de Stassart 36, B-1050 Brussels

© 1997 Copyright reserved to CEN members

Ref. No. EN ISO/IEC 13818-1:1997 E
 **Documents**
Document Sharing Hub

Foreword

The text of the International Standard from Technical Committee ISO/IEC/JTC 1, Information technology, of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) has been taken over as a European Standard by the Technical Board of CEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by August 1997, and conflicting national standards shall be withdrawn at the latest by August 1997.

According to CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

INTERNATIONAL
STANDARD

ISO/IEC
13818-1

First edition
1996-04-15

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

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



Contents

	Page
Foreword	ix
Introduction	1
Intro. 1 Transport Stream	2
Intro. 2 Program Stream	4
Intro. 3 Conversion between Transport Stream and Program Stream	5
Intro. 4 Packetized Elementary Stream	5
Intro. 5 Timing model	6
Intro. 6 Conditional access	6
Intro. 7 Multiplex-wide operations	6
Intro. 8 Individual stream operations (PES Packet Layer)	7
Intro. 8.1 Demultiplexing	7
Intro. 8.2 Synchronization	7
Intro. 8.3 Relation to compression layer	7
Intro. 9 System reference decoder	8
Intro. 10 Applications	8
Section 1. General	
1.1 Scope	8
1.2 Normative references	9
1.3 Identical Recommendations International Standards	9
1.4 Additional references	9
Section 2. Technical elements	
2.1 Definitions	9
2.2 Symbols and abbreviations	14
2.2.1 Arithmetic operators	14
2.2.2 Logical operators	15
2.2.3 Relational operators	15
2.2.4 Bitwise operators	15
2.2.5 Assignment	15
2.2.6 Mnemonics	15
2.2.7 Constants	16
2.3 Method of describing bit stream syntax	16
2.4 Transport Stream bitstream requirements	17
2.4.1 Transport Stream coding structure and parameters	17
2.4.2 Transport Stream system target decoder	18
2.4.3 Specification of the Transport Stream syntax and semantics	26
2.4.3.2 Transport Stream packet layer	26
2.4.3.3 Semantic definition of fields in Transport Stream packet layer	26
2.4.3.4 Adaptation field	28
2.4.3.5 Semantic definition of fields in adaptation field	28
2.4.3.6 PES packet	36
2.4.3.7 Semantic definition of fields in PES packet	39
2.4.3.8 Carriage of Program Streams and ISO/IEC 11172-1 Systems streams in the Transport Stream	46
2.4.4 Program specific information	46
2.4.4.1 Pointer	48
2.4.4.2 Semantics definition of fields in pointer syntax	48
2.4.4.3 Program association Table	49
2.4.4.4 Table_id assignments	49
2.4.4.5 Semantic definition of fields in program association section	49
2.4.4.6 Conditional access Table	50

	Page
2.4.4.7	Semantic definition of fields in conditional access section 51
2.4.4.8	Program Map Table 51
2.4.4.9	Semantic definition of fields in Transport Stream program map section 51
2.4.4.10	Syntax of the Private section 53
2.4.4.11	Semantic definition of fields in private section 54
2.5	Program Stream bitstream requirements 55
2.5.1	Program Stream coding structure and parameters 55
2.5.2	Program Stream system target decoder 55
2.5.2.1	System clock frequency 56
2.5.2.2	Input to the Program Stream system target decoder 56
2.5.2.3	Buffering 57
2.5.2.4	PES streams 58
2.5.2.5	Decoding and presentation 58
2.5.3	Specification of the Program Stream syntax and semantics 58
2.5.3.1	Program Stream 58
2.5.3.2	Semantic definition of fields in Program Stream 59
2.5.3.3	Pack layer of Program Stream 59
2.5.3.4	Semantic definition of fields in program stream pack 59
2.5.3.5	System header 60
2.5.3.6	Semantic definition of fields in system header 60
2.5.3.7	Packet layer of Program Stream 62
2.5.4	Program Stream map 62
2.5.4.1	Syntax of Program Stream map 62
2.5.4.2	Semantic definition of fields in Program Stream map 62
2.5.5	Program Stream directory 63
2.5.5.1	Syntax of Program Stream directory packet 64
2.5.5.2	Semantic definition of fields in Program Stream directory 64
2.6	Program and program element descriptors 66
2.6.1	Semantic definition of fields in program and program element descriptors 66
2.6.2	Video stream descriptor 66
2.6.3	Semantic definitions of fields in video stream descriptor 67
2.6.4	Audio stream descriptor 68
2.6.5	Semantic definition of fields in audio stream descriptor 68
2.6.6	Hierarchy descriptor 68
2.6.7	Semantic definition of fields in hierarchy descriptor 69
2.6.8	Registration descriptor 69
2.6.9	Semantic definition of fields in registration descriptor 69
2.6.10	Data stream alignment descriptor 69
2.6.11	Semantics of fields in data stream alignment descriptor 70
2.6.12	Target background grid descriptor 70
2.6.13	Semantics of fields in target background grid descriptor 71
2.6.14	Video window descriptor 71
2.6.15	Semantic definition of fields in video window descriptor 71
2.6.16	Conditional access descriptor 72
2.6.17	Semantic definition of fields in conditional access descriptor 72
2.6.18	ISO 639 language descriptor 72
2.6.19	Semantic definition of fields in ISO 639 language descriptor 73

		Page
2.6.20	System clock descriptor	73
2.6.21	Semantic definition of fields in system clock descriptor	74
2.6.22	Multiplex buffer utilization descriptor	74
2.6.23	Semantic definition of fields in multiplex buffer utilization descriptor	74
2.6.24	Copyright descriptor	74
2.6.25	Semantic definition of fields in copyright descriptor	74
2.6.26	Maximum bitrate descriptor	75
2.6.27	Semantic definition of fields in maximum bitrate descriptor	75
2.6.28	Private data indicator descriptor	75
2.6.29	Semantic definition of fields in Private data indicator descriptor	75
2.6.30	Smoothing buffer descriptor	75
2.6.31	Semantic definition of fields in smoothing buffer descriptor	76
2.6.32	STD descriptor	76
2.6.33	Semantic definition of fields in STD descriptor	76
2.6.34	IBP descriptor	76
2.6.35	Semantic definition of fields in IBP descriptor	77
2.7	Restrictions on the multiplexed stream semantics	77
2.7.1	Frequency of coding the system clock reference	77
2.7.2	Frequency of coding the program clock reference	77
2.7.3	Frequency of coding the elementary stream clock reference	77
2.7.4	Frequency of presentation timestamp coding	77
2.7.5	Conditional coding of timestamps	78
2.7.6	Timing constraints for scalable coding	78
2.7.7	Frequency of coding P-STD_buffer_size in PES packet headers	79
2.7.8	Coding of system header in the Program Stream	79
2.7.9	Constrained system parameter Program Stream	79
2.7.10	Transport Stream	79
2.8	Compatibility with ISO/IEC 11172	80
Annex A	CRC Decoder Model	81
A.0	CRC decoder model	81
Annex B	Digital Storage Medium Command and Control (DSM-CC)	82
B.0	Introduction	82
B.0.1	Purpose	82
B.0.2	Future applications	82
B.0.3	Benefits	82
B.0.4	Basic functions	82
B.0.4.1	Stream selection	82
B.0.4.2	Retrieval	83
B.0.4.3	Storage	83
B.1	General elements	83
B.1.1	Scope	83
B.1.2	Overview of the DSM-CC application	83
B.1.3	The transmission of DSM-CC commands and acknowledgments	84
B.2	Technical elements	85
B.2.1	Definitions	85
B.2.2	Specification of DSM-CC syntax	85
B.2.3	Semantics of fields in specification of DSM-CC syntax	86
B.2.4	Control layer	86

		Page
B.2.5	Semantics of fields in control layer	86
B.2.6	Acknowledgment layer	88
B.2.7	Semantics of fields in acknowledgment layer	88
B.2.8	Time code	89
B.2.9	Semantics of fields in time code	89
Annex C	Program Specific Information	90
C.0	Explanation of Program Specific Information in Transport Streams	90
C.1	Introduction	90
C.2	Functional Mechanism	90
C.3	The Mapping of Sections into Transport Stream Packets	91
C.4	Repetition Rates and Random Access	91
C.5	What is a Program?	91
C.6	Allocation of program_number	92
C.7	Usage of PSI in a Typical System	92
C.8	The Relationships of PSI Structures	93
C.8.1	Program Association Table	93
C.8.2	Program Map Table	93
C.8.3	Conditional Access Table	93
C.8.4	Network Information Table	94
C.8.5	Private_section()	95
C.8.6	Descriptors	95
C.9	Bandwidth Utilization and Signal Acquisition Time	95
Annex D	Systems Timing Model and Application Implications of this Recommendation International Standard	97
D.0	Introduction	97
D.0.1	Timing Model	98
D.0.2	Audio and Video Presentation Synchronization	99
D.0.3	System Time Clock recovery in the decoder	101
D.0.4	SCR and PCR Jitter	103
D.0.5	Clock Recovery in the Presence of Network Jitter	105
D.0.6	System clock used for chroma sub-carrier generation	105
D.0.7	Component video and audio reconstruction	107
D.0.8	Frame Slipping	107
D.0.9	Smoothing of network jitter	107
Annex E	Data Transmission Applications	108
E.0	General considerations	108
E.1	Suggestion	108
Annex F	Graphics of Syntax for this Recommendation International Standard	109
F.0	Introduction	109
F.0.1	Transport Stream syntax	109
F.0.2	PES packet	110
F.0.3	Program Association Section	111
F.0.4	CA section	111
F.0.5	TS program map section	112
F.0.6	Private section	112
F.0.7	Program Stream	113
F.0.8	Program Stream map	114

	Page
Annex G General Information	114
G.0 General Information	114
G.0.1 Sync Byte Emulation	114
G.0.2 Skipped picture status and decoding process	114
G.0.3 Selection of PID Values	114
G.0.4 PES start_code emulation	115
Annex H Private Data	115
H.0 Private Data	115
Annex I Systems conformance and real-time interface	116
I.0 Systems conformance and real-time interface	116
Annex J Interfacing Jitter-Inducing Networks to MPEG-2 Decoders	116
J.0 Introduction	116
J.1 Network compliance models	117
J.2 Network specification for jitter smoothing	118
J.3 Example decoder implementations	119
J.3.1 Network adapter followed by an MPEG-2 decoder	119
J.3.2 Integrated decoder	119
Annex K Splicing Transport Streams	119
K.0 Introduction	119
K.1 The different types of splicing point	120
K.1.1 Ordinary splicing points	120
K.1.2 Seamless splicing points	120
K.2 Decoder behaviour on splices	121
K.2.1 On non-seamless splices	121
K.2.2 On seamless splices	121
K.2.3 Buffer Overflow	121
Figure Intro. 1 — Simplified overview the scope of this Recommendation International Standard	1
Figure Intro. 2 — Prototypical transport demultiplexing and decoding example	3
Figure Intro. 3 — Prototypical transport multiplexing example	3
Figure Intro. 4 — Prototypical Transport Stream to Program Stream conversion	4
Figure Intro. 5 — Prototypical decoder for Program Streams	4
Figure 2-1 — Transport Stream system target decoder notation	18
Figure 2-2 — Program Stream system target decoder notation	55
Figure 2-3 — Target background grid descriptor display area	71
Figure A.1 — 32-bit CRC decoder model	81
Figure B.1 — Configuration of DSM-CC application	83
Figure B.2 — DSM-CC bitstream decoded as a standalone bitstream	84
Figure B.3 — DSM-CC bitstream decoded as part of the system bitstream	84
Figure C.1 — Program and network mapping relationships	94
Figure D.1 — Constant delay model	98
Figure D.2 — STC recovery using PLL	102
Figure F.1 — Transport Stream syntax diagram	109
Figure F.2 — PES packet syntax diagram	110
Figure F.3 — Program association section diagram	111
Figure F.4 — Conditional access section diagram	111
Figure F.5 — TS program map section diagram	112

	Page
Figure F.6 — Private section diagram	112
Figure F.7 — Program Stream diagram	113
Figure F.8 — Program Stream map diagram	114
Figure J.1 — Sending system streams over a jitter-inducing network	117
Figure J.2 — Jitter-smoothing using network-layer timestamps	118
Figure J.3 — Integrated dejittering and MPEG-2 decoding	119
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	29
Table 2-7 — Splice parameters Table 1	34
Table 2-8 — Splice parameters Table 2	34
Table 2-9 — Splice parameters Table 3	34
Table 2-10 — Splice parameters Table 4	35
Table 2-11 — Splice parameters Table 5	35
Table 2-12 — Splice parameters Table 6	35
Table 2-13 — Splice parameters Table 7	35
Table 2-14 — Splice parameters Table 8	36
Table 2-15 — Splice parameters Table 9	36
Table 2-16 — Splice parameters Table 10	36
Table 2-17 — PES packet	37
Table 2-18 — Stream_id assignments	40
Table 2-19 — PES scrambling control values	40
Table 2-20 — Trick mode control values	42
Table 2-21 — Field_id field control values	44
Table 2-22 — Coefficient selection values	44
Table 2-23 — Program specific information	47
Table 2-24 — Program specific information pointer	48
Table 2-25 — Program association section	49
Table 2-26 — table_id assignment values	49
Table 2-27 — Conditional access section	50
Table 2-28 — Transport Stream program map section	52
Table 2-29 — Stream type assignments	53
Table 2-30 — Private section	54
Table 2-31 — Program Stream	58
Table 2-32 — Program Stream pack	59
Table 2-33 — Program Stream pack header	59
Table 2-34 — Program Stream system header	60
Table 2-35 — Program Stream map	63
Table 2-36 — Program Stream directory packet	65
Table 2-37 — Intra_coded indicator	65
Table 2-38 — Coding_parameters indicator	66
Table 2-39 — Program and program element descriptors	66
Table 2-40 — Video stream descriptor	67
Table 2-41 — Frame rate code	67

	Page
Table 2-42 — Audio stream descriptor	68
Table 2-43 — Hierarchy descriptor	68
Table 2-44 — Hierarchy_type field values	69
Table 2-45 — Registration descriptor	69
Table 2-46 — Data stream alignment descriptor	70
Table 2-47 — Video stream alignment values	70
Table 2-48 — Audio stream alignment values	70
Table 2-49 — Target background grid descriptor	71
Table 2-50 — Video window descriptor	71
Table 2-51 — Conditional access descriptor	72
Table 2-52 — ISO 639 language descriptor	73
Table 2-53 — Audio type values	73
Table 2-54 — System clock descriptor	73
Table 2-55 — Multiplex buffer utilization descriptor	74
Table 2-56 — Copyright descriptor	74
Table 2-57 — Maximum bitrate descriptor	75
Table 2-58 — Private data indicator descriptor	75
Table 2-59 — Smoothing buffer descriptor	76
Table 2-60 — STD descriptor	76
Table 2-61 — IBP descriptor	77
Table B.1 — DSM-CC syntax	85
Table B.2 — Command_id assigned values	86
Table B.3 — DSM-CC control	87
Table B.4 — Select mode assigned values	87
Table B.5 — DSM-CC Acknowledgment	88
Table B.6 — Time code	89
Table C.1 — Composite_descriptor	95
Table C.2 — Sub-descriptor	95
Table C.3 — Program association table bandwidth usage (bit/s)	96
Table C.4 — Program map table bandwidth usage (bit/s)	96
Table D.1 — Re-multiplexing strategy	104
Table E.1 — PES packet header example	108

Descriptors: Data processing, moving pictures, image processing, video recording, sound recording, video data, audio data, data converting, coding (data conversion), system architecture.

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 13818-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation H.222.0.

ISO/IEC 13818 consists of the following parts, under the general title *Information technology — Generic coding of moving pictures and associated audio information*:

- *Part 1: Systems*;
- *Part 2: Video*;
- *Part 3: Audio*;
- *Part 4: Compliance testing*;
- *Part 6: Extensions for DSM-CC*;
- *Part 9: Extension for real time interface for systems decoders*.

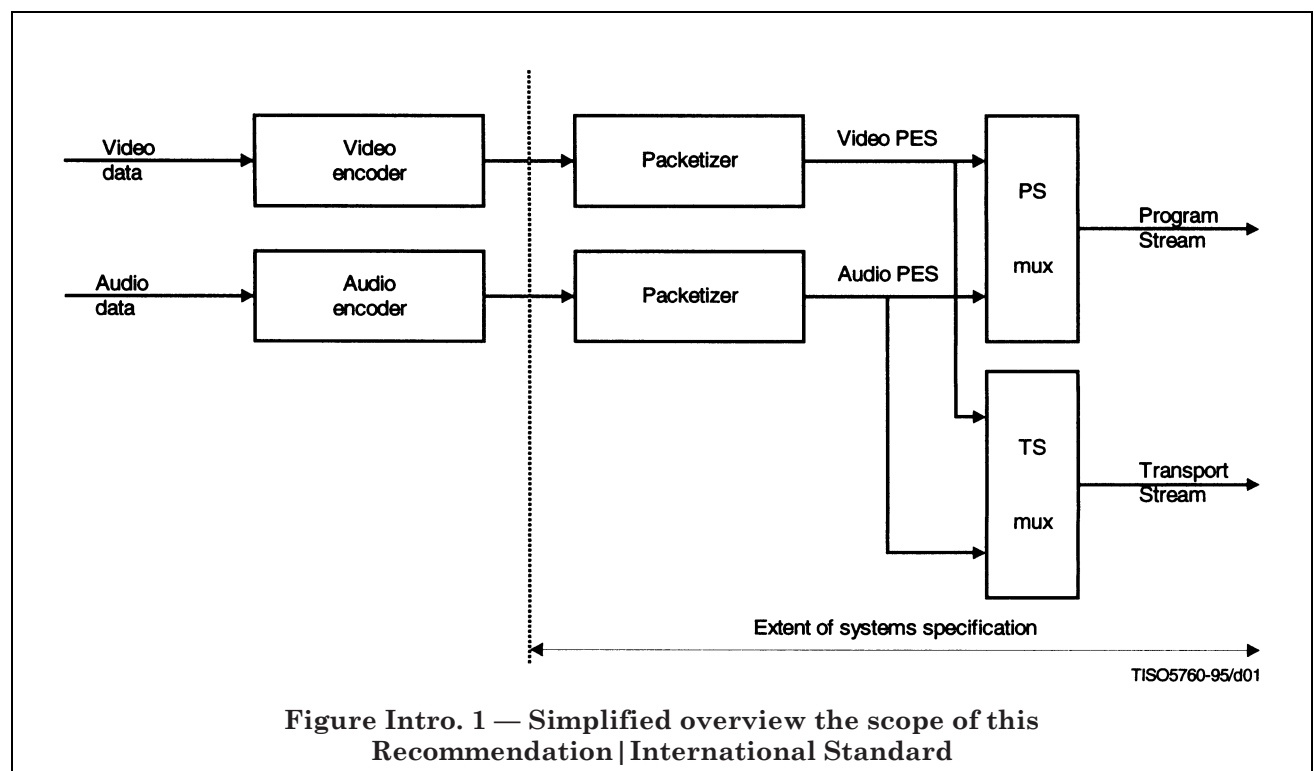
Annex A forms an integral part of this part of ISO/IEC 13818. Annex B to Annex K are for information only.

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 ITU-T Rec. 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.



The **Program Stream** is analogous and similar to 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 which comprise a single program to be in separate streams which 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 timebase. 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 differ: 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 ITU-T Rec. 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 2.4.2.2.

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.

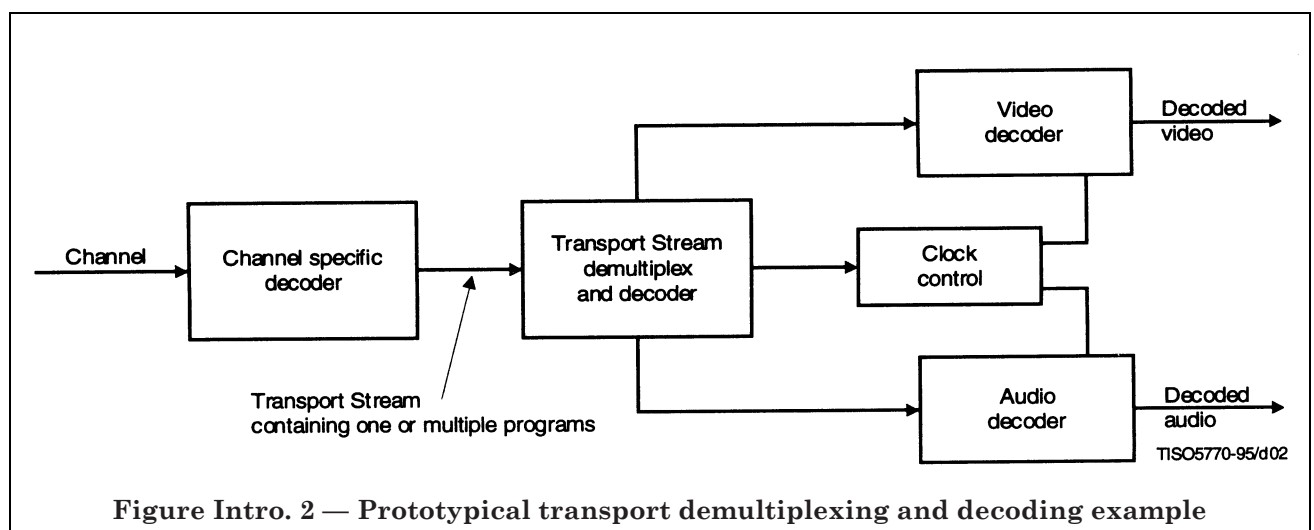


Figure Intro. 2 — Prototypical transport demultiplexing and decoding example

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.

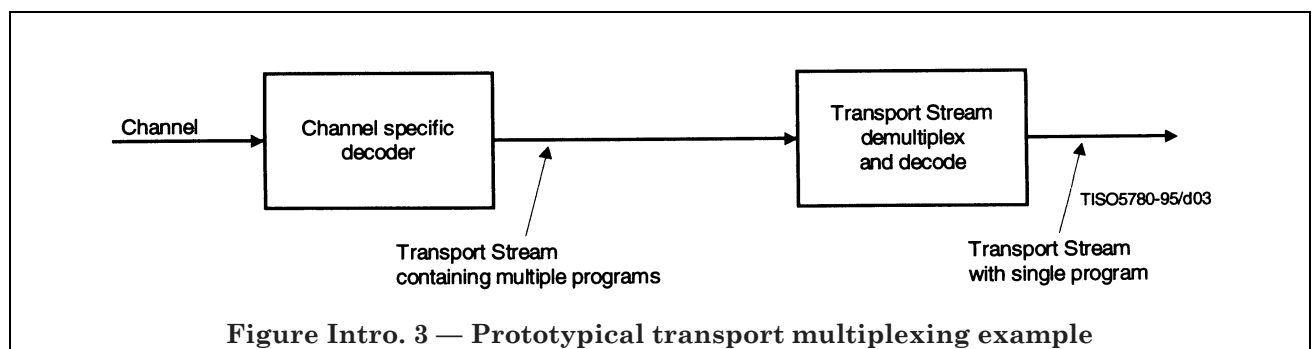
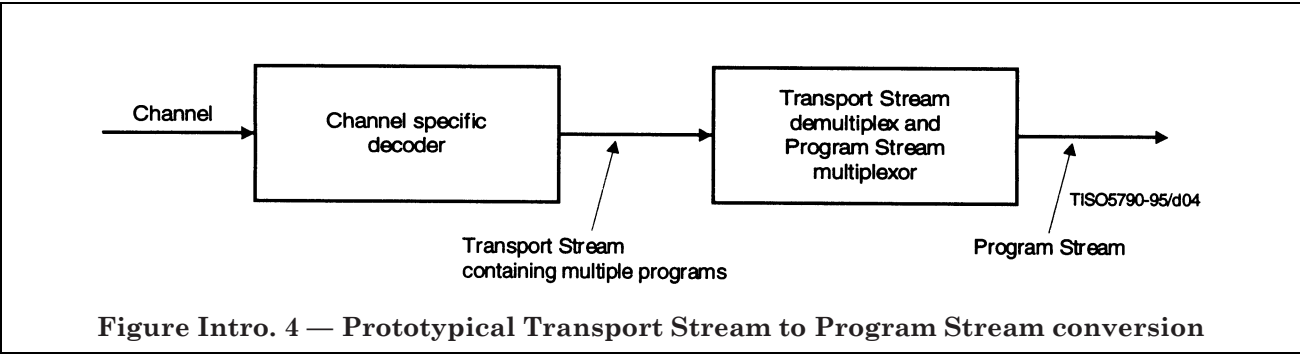


Figure Intro. 3 — Prototypical transport multiplexing example

Figure Intro. 4 illustrates a case in which an multi-program Transport Stream is first demultiplexed and then converted into a Program Stream.

Figure Intro. 3 and Figure 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.

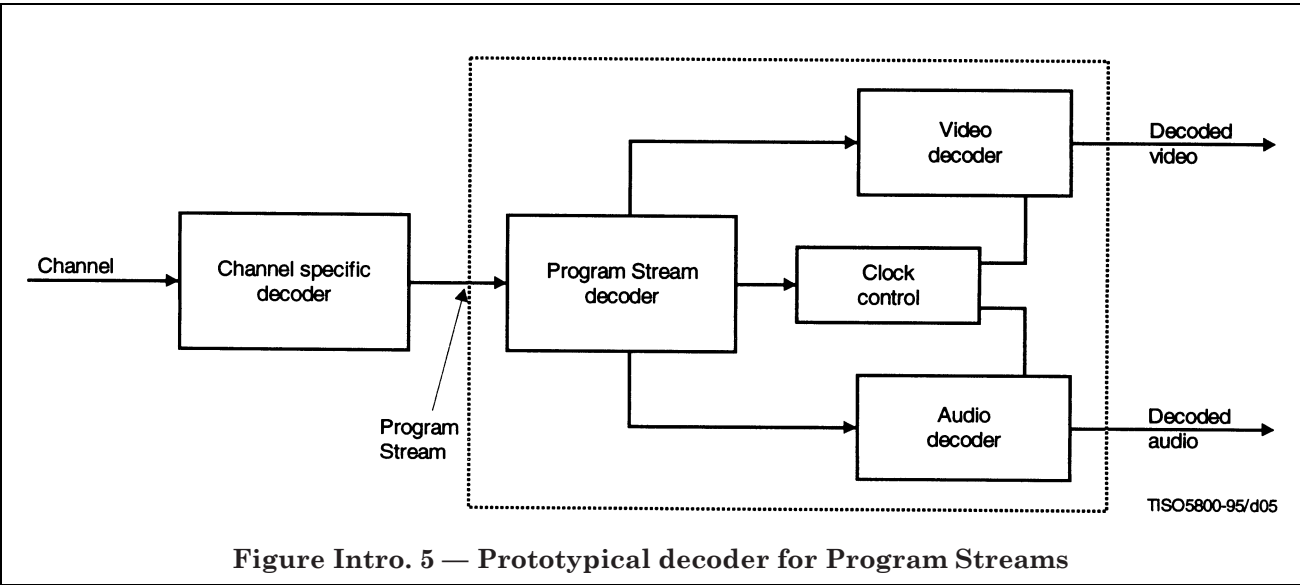


Intro. 2 Program Stream

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 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 an Program Stream decoder. Indeed non-audio/video data is also allowed, but not shown.



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. 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 (PTS) 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 ITU-T Rec. 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 ITU-T Rec. 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.

Modern data communications networks may be capable of supporting ITU-T Rec. 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 ITU-T Rec. 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.

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 five 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; and
- 5) time identification.

An ITU-T Rec. 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.

An ITU-T Rec. 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 indicated 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.

1.3 Identical Recommendations | International Standards

— ITU-T Recommendation H.262 (1995) | ISO/IEC 13818-2: ..., *Information technology — Generic coding of moving pictures and associated audio information: Video*¹⁾.

1.4 Additional references

- ISO 639-2: ..., *Codes for the representation of names of languages — Part 2: Alpha-3 code*²⁾.
- ISO 8859-1:1987, *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*
- 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 13522-1: ..., *Information technology — Coding of Multimedia and Hypermedia information — Part 1: MHEG object representation — Base notation (ASN.1)*²⁾.
- ISO/IEC 13818-3:1995, *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio.*
- Recommendation ITU-R BT.601.3, *Encoding parameters of digital television for studios.*
- Recommendation ITU-R BT.470-2, *Television systems.*
- Recommendation ITU-R BR.648, *Digital recording of audio signals.*
- Report ITU-R BO.955.2, *Satellite sound broadcasting of vehicular, portable, and fixed receivers in the range 500–3 000 MHz.*
- CCITT Recommendation J.17 (1988), *Pre-emphasis used on sound-programme circuits.*
- IEEE Standard 1180:1990, *Standard Specification for the Implementations of 8 by 8 Inverse Discrete Cosine Transform.*
- IEC Publication 908:1987, *Compact disc digital audio system.*

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

¹⁾ To be published.

²⁾ To be published.

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

2.1.2

bitrate

the rate at which the compressed bit stream is delivered from the channel to the input of a decoder

2.1.3

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

2.1.4

channel

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

2.1.5

coded representation

a data element as represented in its encoded form

2.1.6

compression

reduction in the number of bits used to represent an item of data

2.1.7

constant bitrate

operation where the bitrate is constant from start to finish of the compressed bit stream

2.1.8

constrained system parameter stream; CSPS (system)

a Program Stream for which the constraints defined in 2.7.9 of ITU-T Rec. H.222.0|ISO/IEC 13818-1 apply

2.1.9

CRC

the Cyclic Redundancy Check to verify the correctness of data

2.1.10

data element

an item of data as represented before encoding and after decoding

2.1.11

decoded stream

the decoded reconstruction of a compressed bit stream

2.1.12

decoder

an embodiment of a decoding process

2.1.13

decoding (process)

the process defined in ITU-T Rec. H.222.0|ISO/IEC 13818-1 that reads an input coded bit stream and outputs decoded pictures or audio samples

2.1.14

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

2.1.15**digital storage media; DSM**

a digital storage or transmission device or system

2.1.16**DSM-CC**

digital storage media command and control

2.1.17**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

2.1.18**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

2.1.19**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

2.1.20**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

2.1.21**elementary stream clock reference; ESCR (system)**

a time stamp in the PES Stream from which decoders of PES streams may derive timing

2.1.22**encoder**

an embodiment of an encoding process

2.1.23**encoding (process)**

a process, not specified in ITU-T Rec. H.222.0|ISO/IEC 13818-1, that reads a stream of input pictures or audio samples and produces a coded bit stream conforming to ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.24**entropy coding**

variable length lossless coding of the digital representation of a signal to reduce redundancy

2.1.25**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

2.1.26**fast forward playback (video)**

the process of displaying a sequence, or parts of a sequence, of pictures in display-order faster than real-time

2.1.27**forbidden**

the term “forbidden”, when used in the clauses of ITU-T Rec. H.222.0|ISO/IEC 13818-1 defining the coded bit stream, indicates that the value specified shall never be used

2.1.28

(multiplexed) stream (system)

a bit stream composed of 0 or more elementary streams combined in a manner that conforms to ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.29

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 ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.30

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 of ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.31

packet data (system)

contiguous bytes of data from an elementary stream present in a packet

2.1.32

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 of ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.33

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

2.1.34

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

2.1.35

PES (system)

an abbreviation for Packetized Elementary Stream

2.1.36

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 of ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.37

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

2.1.38

PES stream (system)

a PES stream 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 of ITU-T Rec. H.222.0|ISO/IEC 13818-1

2.1.39**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

2.1.40**presentation unit; PU (system)**

a decoded Audio Access Unit or a decoded picture

2.1.41**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

2.1.42**program clock reference; PCR (system)**

a time stamp in the Transport Stream from which decoder timing is derived

2.1.43**program element (system)**

a generic term for one of the elementary streams or other data streams that may be included in a program

2.1.44**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 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1. An example of privately defined PSI data is the non-mandatory network information table

2.1.45**random access**

the process of beginning to read and decode the coded bit stream at an arbitrary point

2.1.46**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 ITU-T Rec. H.222.0 | ISO/IEC 13818-1, all reserved bits shall be set to “1”

2.1.47**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

2.1.48**source stream**

a single non-multiplexed stream of samples before compression coding

2.1.49**splicing (system)**

the concatenation, performed on the system level, of two different elementary streams. The resulting system stream conforms totally to ITU-T Rec. H.222.0 | ISO/IEC 13818-1. The splice may result in discontinuities in timebase, continuity counter, PSI, and decoding

2.1.50**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 2-18 of ITU-T Rec. H.222.0 | ISO/IEC 13818-1

2.1.51

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

2.1.52

still picture

a coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded. This picture has an associated PTS and the presentation time of succeeding pictures, if any, is later than that of the still picture by at least two picture periods

2.1.53

system header (system)

the system header is a data structure defined in **2.5.3.5** of ITU-T Rec. H.222.0|ISO/IEC 13818-1 that carries information summarizing the system characteristics of ITU-T Rec. H.222.0|ISO/IEC 13818-1 Program Stream

2.1.54

system clock reference; SCR (system)

a time stamp in the Program Stream from which decoder timing is derived

2.1.55

system target decoder; STD (system)

a hypothetical reference model of a decoding process used to define the semantics of an ITU-T Rec. H.222.0|ISO/IEC 13818-1 multiplexed bit stream

2.1.56

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

2.1.57

transport stream packet header (system)

the leading fields in a Transport Stream packet, up to and including the continuity_counter field

2.1.58

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

2.2 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 $-\infty$
%	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
$\sqrt{\quad}$	Square root
$\log_{10}$	Logarithm to base ten
$\log_e$	Logarithm to base e

2.2.2 Logical operators

	Logical OR
&&	Logical AND
!	Logical NOT

2.2.3 Relational operators

>	Greater than
$\geq$	Greater than or equal to
<	Less than
$\leq$	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

2.2.4 Bitwise operators

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

2.2.5 Assignment

=	Assignment operator
---	---------------------

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.
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 Constants

π	3.14159265359
e	2.71828182845

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) { data_element ... }	If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.
do { data_element ... } while (condition)	The data element always occurs at least once. The data element is repeated until the condition is not true.
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 [] data_element [] is an array of data. The number of data elements is indicated by the context.

data_element [n] data_element [n] is the n + 1th element of an array of data.

data_element [m][n] data_element [m][n] is the m + 1, n + 1th element of a two-dimensional array of data.

data_element [l][m][n] data_element [l][m][n] 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 data_element.

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 ITU-T Rec. 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 four PSI tables:

- Program Association Table;
- Program Map Table;
- Conditional Access Table;
- Network Information Table.

These tables contain the necessary and sufficient information to demultiplex and present programs. The Program Map Table, in Table 2-28, 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.

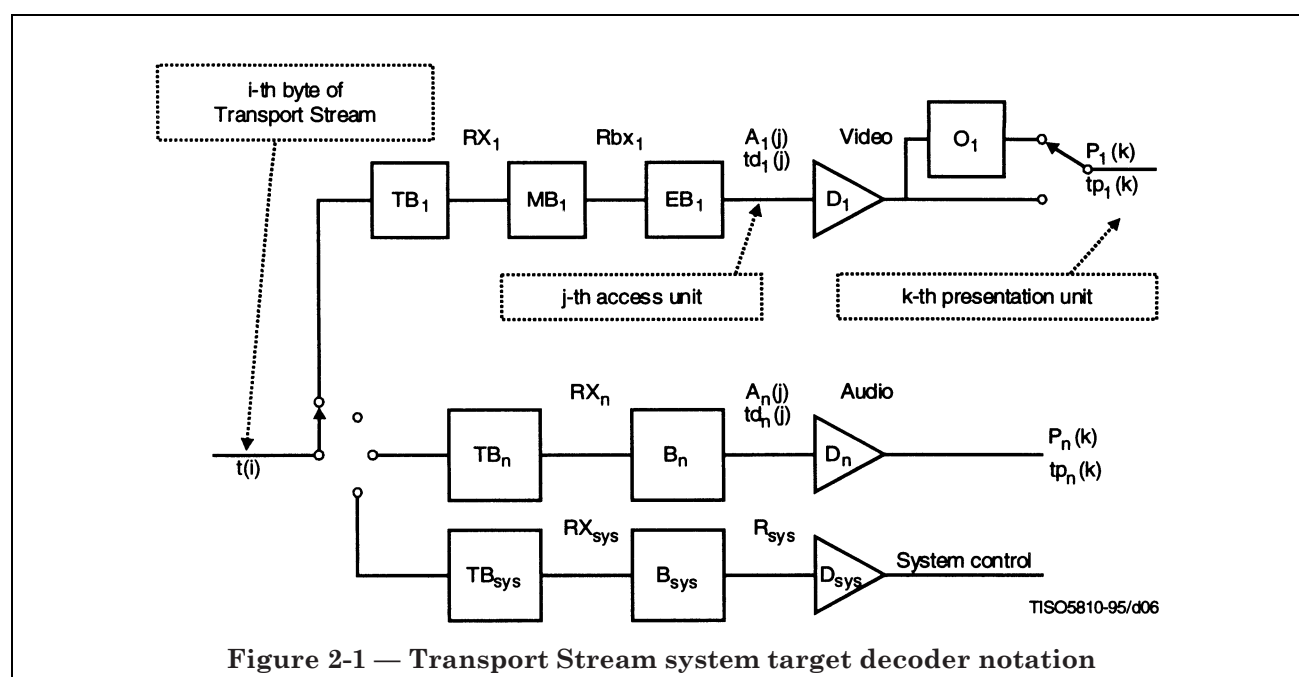
Transport Stream packets may be null packets. Null packets are intended for padding of Transport Streams. They may be inserted or deleted by re-multiplexing processes and, therefore, the delivery of the payload of null packets to the decoder cannot be assumed.

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 2-27 of the PSI).

2.4.2 Transport Stream system target decoder

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.



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

- 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 vbv_delay method is used. Defined only for video elementary streams.
Rx_{sys}	The rate at which data are removed from TB_{sys} .
R_{es}	The video elementary stream rate coded in a sequence header.

2.4.2.1 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:

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

rate of change of system_clock_frequency with time $\leq 75 \times 10^{-3}$ 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.

2.4.2.2 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 PCR_base(i) (see equation 2-2) and 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:

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

where:

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

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

For all other bytes the input arrival time, $t(i)$ shown in equation 2-4 below, is computed from $PCR(i'')$ 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.

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

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

The transport rate 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.

NOTE $i'' < i \leq i'$

In the case of a timebase 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 timebase and the first PCR of the new timebase. 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 timebase.

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.

2.4.2.3 Buffering

Complete Transport Stream packets containing data from elementary stream n , as indicated by its PID, are passed to the transport buffer for stream n , TB_n . This includes duplicate Transport Stream packets and packets with no payload. Transfer of the i -th byte from the system target decoder input to TB_n is instantaneous, so that the i -th byte enters the buffer for stream n , of size TBS_n , at time $t(i)$.

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 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 ITU-T Rec. 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 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 or 1, and all Transport Stream packets identified via the Program Association Table (see Table 2-25) 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 Summary of Constrained Parameters in ISO/IEC 11172-2 and Table 8-14 of ITU-T Rec. H.262 | ISO/IEC 13818-2.

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

For Low and Main level

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

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 $VB_{max}[profile, level]$ is defined in Table 8-14 in ITU-T Rec. H.262 | ISO/IEC 13812-2 and $R_{max}[profile, level]$ is defined in Table 8-13 in ITU-T Rec. H.262 | ISO/IEC 13818-2, and `vbv_buffer_size` is carried in the sequence header described in 6.2.2 of ITU-T Rec. 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 $R_{max}[profile, level]$ is defined in Table 8-13 in ITU-T Rec. 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 $BS_{mux} = 4 \text{ ms} \times R_{max}[profile, level]$ 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 1 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

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

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

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

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 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 **2.4.3.6**).

For Low and Main level:

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

For High-1440 and High level:

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

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} . 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 ITU-T Rec. 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:

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

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 2 $vbv_delay(j+1)$ 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 2-29), where $R_{max}[profile, level]$ is defined in ITU-T Rec. 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 ITU-T Rec. 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} .

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

NOTE 3 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 ITU-T Rec. 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.

2.4.2.4 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.

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

2.4.2.5 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.

2.4.2.6 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 ITU-T Rec. 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 Decoders buffers shall be less than or equal to one second except for still picture video data. 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 $A_n(j)$.

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 .

$$0 \leq F_n(t)$$

2.4.3 Specification of the Transport Stream syntax and semantics

2.4.3.1 Transport Stream

Table 2-1 — Transport Stream

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

See Table 2-2.

Syntax	No. of bits	Mnemonic
transport_packet() { sync_byte transport_error_indicator payload_unit_start_indicator transport_priority PID transport_scrambling_control adaptation_field_control continuity_counter 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 1 1 1 1 13 2 2 4 8	bslbf bslbf bslbf bslbf uimbsbf bslbf bslbf uimbsbf bslbf

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 PSI data (refer to 2.4.4).