BRAZILIAN
STANDARD

ABNT NBR 15603-2

First edition
2007.11.30

Valid from
2007.12.01

Corrected Version
2008.08.22

Digital terrestrial television — Multiplexing and service information (SI) Part 2: Data structure and definitions of basic information of SI

Descriptors: Digital terrestrial television. Service information. Program specific information. Descriptors.

ICS 33.160.01

ISBN 978-85-07-00611-4

ASSOCIAÇÃO
BRASILEIRA
DE NORMAS
TÉCNICAS

Reference number
ABNT NBR 15603-2:2007
129 páginas

ABNT office
Av.Treze de Maio, 13 - 28º andar
20031-901 - Rio de Janeiro - RJ
Tel.: + 55 21 3974-2300
Fax: + 55 21 2220-1762
abnt@abnt.org.br
www.abnt.org.br

Published in Brazil

Contents

Pages

Foreword

Associação Brasileira de Normas Técnicas (ABNT) is the Brazilian Standardization Forum. Brazilian Standards, which content is responsability of the Brazilian Committees (Comitês Brasileiros – ABNT/CB), Sectorial Standardization Bodies (Organismos de Normalização Setorial – ABNT/ONS) and Special Studies Committees (Comissões de Estudo Especiais – ABNT/CEE), are prepared by Study Committees (Comissões de Estudo – CE), made up of representants from the sectors involved including: producers, consumers and neutral entities (universities, laboratories and others).

Brazilian Standards are drafted in accordance with the rules given in the ABNT Directives (Diretivas), Part 2.

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

ABNT NBR 15603-2 was prepared within the purview of the Special Studies Committee of Digital Television (ABNT/CEE-00:001.85). The project was circulated for National Consultation in accordance with ABNT Notice (Edital) nº 07, from June 29, 2007 to August 28, 2007, with the number Draft 00:001.85-003/2.

Should any doubts arise regarding the interpretation of the English version, the provisions in the original text in Portuguese shall prevail at all time.

This standard is based on the work of the Brazilian Digital Television Forum as established by the Presidential Decree number 5.820 of June, 29th 2006.

ABNT NBR 15603 consists of the following parts, under the general title "*Digital terrestrial television – Multiplexing and service information (SI)*":

- Part 1: SI of the digital broadcasting systems;
- Part 2: Data structure and definitions of basic information of SI;
- Part 3: Syntaxes and definitions of extended information of SI.

This English version is equivalent to the corrected version 2 of ABNT NBR 15603-2:2007, from 2008.08.22.

This corrected version of ABNT NBR 15603-2:2007 includes the Technical Corrigendum 1 from 2008.08.22.

Digital terrestrial television — Multiplexing and service information (SI) Part 2: Data structure and definitions of basic information of SI

1 Scope

This part of ABNT NBR 15603 specifies the basic table of service information, known by SI tables, for broadcast signals which are part of data broadcasting of Brazilian terrestrial digital television system (SBTVD).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Directive of secretary of Justice nº 1220, July 11t, 2007, *Establishes the law no. 8.069, July 13,1990 (Teenager and Children Rights), of the law nº 10.359, from December 27, 2001, and of the decree nº 5834, of July 6, 2006, related to the parental rating process of audiovisual programs for television*

ABNT NBR 15601, *Digital terrestrial television – Transmission system*

ABNT NBR 15602-1, *Digital terrestrial television – Video coding, áudio coding and multiplexing – Part 1: Video coding*

ABNT NBR 15602-2, *Digital terrestrial television – Video coding, áudio coding and multiplexing – Part 2: Audio coding*

ABNT NBR 15603-1:2007, *Digital terrestrial television – Multiplexing and service information (SI) – Part 1: SI for digital broadcasting systems*

ISO 639-2, *Codes for the representation of names of languages – Part 2: Alpha-3 code*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

ISO/IEC 8859-15, *Information technology – 8-bit single-byte coded graphic character sets – Part 15: Latin alphabet Nº 9*

ISO/IEC 11172-2, *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, *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 13522-5, *Information technology – Coding of multimedia and hypermedia information – Part 5: Support for base-level interactive applications*

ISO/IEC 13818-1:2007, *Information technology – Generic coding of moving pictures and associated audio information: Systems*

ISO/IEC 13818-3, *Information technology – Generic coding of moving pictures and associated audio information -- Part 3: Audio*

ISO/IEC 13818-6, *Information technology – Generic coding of moving pictures and associated audio information -- Part 6: Extensions for DSM-CC*

ISO/IEC 13818-7, *Information technology – Generic coding of moving pictures and associated audio information -- Part 7: Advanced Audio Coding (AAC)*

ISO/IEC 13818-11, *Information technology – Generic coding of moving pictures and associated audio information -- Part 11: IPMP on MPEG-2 systems*

ISO/IEC 14496-1, *Information technology – Coding of audio-visual objects – Part 1: Systems*

ISO/IEC 14496-2, *Information technology – Coding of audio-visual objects – Part 2: Visual*

ISO/IEC 14496-3, *Information technology – Coding of audio visual objects – Part 3: Áudio*

ISO/IEC 14496-10, *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*

ITU Recommendation H.222.0:2002, *Information technology – Generic coding of moving pictures and associated audio information: Systems*

ITU Recommendation H.222.1, *Multimedia multiplex and synchronization for audiovisual communication in ATM environments*

ITU Recommendation H.262, *Information technology – Generic coding of moving pictures and associated audio information: Video*

ITU Recommendation H.264:2005, *Advanced video coding for generic audiovisual services*

ETSI EN 300 468:2007*, Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB*

ETSI EN 301 790, *Digital Video Broadcasting (DVB); Interaction channel for satellite distribution*

ETSI EN 50221, *Common interface specification for conditional access and other Digital Video Broadcasting decoder applications*

ARIB STD-B10, *Service information for digital broadcasting system*

ARIB STD-B21, *Receiver for digital broadcasting*

ARIB STD-B24, *Data coding and transmission specification for digital broadcasting*

RFC 1521, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for specifying and describing the format of internet message bodies*

RFC 1590, *Media type registration procedure*

3 Terms and definitions

For the purposes of this part of ABNT NBR 15603, the following terms and definitions apply.

3.1
entitlement management message
EMM
conditional access information which specify the authorization or services levels or the services allowed for each decoder

NOTE EMM can be addressed individually or for groups of decoders.

3.2
Modified Julian Date
MJD
indication of Brazilian official data

NOTE The conversion of MJD to Brazilian official data is described in Annex A.

3.3
original_network_id
unique identifier of a network

3.4
full seg receiver
device which decodes entirely audio, video, data and other type of information contained in the 13 segment transport stream layer designated to fix (indoor) and mobile

NOTE The classification of full-seg shall be applied to digital converters, also known by set top box and the 13 segments with exhibition screen but not limited to those. This kind of receiver shall be capable of receiving and decoding the high definition television signals and at manufactures criteria, also be capable of receiving and decoding the information on layer “A” of the transport stream, which is originally aimed to the portable receivers, here defined as one-seg.

3.5
one-seg receiver
device which decodes entirely audio, video, data and other type of information contained in layer “A” located on the central segment of the 13 segments.

NOTE The classification of the one-seg shall be assigned to portable receivers, also known as handheld devices, with screen size bellow 7 inches. Within the products classified as one-seg are, but not limited to, the mobile phone integrated receivers, PDA, one-seg dongles, portable television, that are energized by an internal battery and, therefore, without the necessary use of an external power source, as well as those aimed vehicular reception. This kind of receiver shall be capable of receiving and decoding only digital terrestrial television signal in layer “A” of the transport stream, and therefore only the basic profile, aimed the portable devices.

3.6
reserved
term used in the definition of a data packet, indicating that the value can be used in the future by a ISO standard which defines their extensions

NOTE Unless otherwise specified within this Standard, all “reserved” bits are set to “1”

3.7
reserved_future_use
term that, when used in the clause defining a data packet, indicates that the value may be used in the future

NOTE Unless otherwise specified in this Standard, all “reserved_future_use” bits are set to “1”.

3.8
transport_stream_id
unique identifier of a TS inside a network

4 Abbreviations

For the purposes of this part of ABNT NBR 15603, the following abbreviations apply.

BAT	Bouquet Association Table
BCD	Binary Coded Decimal
BIT	Broadcaster Information Table
CA	Conditional Access
CAT	Conditional Access Table
CRC	Cyclic Redundancy Check
EIT	Event Information Table
EMM	Entitlement Management Message
EPG	Electronic Program Guide
LDT	Linked Description Table
LSB	Least Significant Bit
MJD	Modified Julian Date
MPEG	Moving Pictures Expert Group
MSB	Most Significant Bit
NBIT	Network Board Information Table
NIT	Network Information Table
NVOD	Near Video On Demand
PAT	Program Association Table
PCAT	Partial Content Announcement Table
PID	Packet Identifier
PMT	Program Map Table
PSI	Program Specific Information
RS	Reed Solomon
RST	Running Status Table
SDT	Service Description Table
SI	Service Information
ST	Stuffing Table
TDT	Time and Date Table
TOT	Time Offset Table
bslbf	Bit String Left Bit First
rpchof	Remainder Polynominal Coefficients, Highest Order First
uimsbf	Unsigned Integer Most Significant Bit First

5 Structure for construction of the basic information

The tables for the construction of the basic information related to SI shall be according to the Tables 1 to 4.

Table 1 — Tables PSI/MPEG-2

Table name	Function
Program Association Table (PAT)	For each service in the multiplexer, PAT shall designate PID values of transport streams (TS). PAT shall generate the link among the “transport_stream_id”, “program_number” and “program_map_id” fields
Program Map Table (PMT)	PMT shall identify and indicate broadcasting location for each service and the program clock reference (PCR) location for each service
Conditional Access Table (CAT)	CAT shall provide information about the conditional access systems used in the multiplexer and the association with EMM broadcasts

Table 2 — SI Tables

Table name	Functions
Bouquet association table (BAT)	It shall provide information about the existing bouquets and services included on each bouquet
Network information table (NIT)	It shall be responsible for informing the physical organization of the multiplexers/transport streams (TS) grouping in a same network, as well as all relevant data on the existing services tunning
Service description table (SDT)	Table that shall inform the existing services in a *transport stream* (TS)
Event information table (EIT)	It shall provide information in chronological order about the existing events by service
Time date table (TDT)	It shall be used as reference to inform the date and time of the system
Time offset table (TOT)	It shall be responsible to inform present date time, and time zone to the receiver This table is mandatory.
Running status table (RST)	It shall allow fast and accurate update of one or more events status, like “pausing” or “running”. It shall be necessary when programming time modifications happens, modifying the event status for pausing or running, and so forth.
Local event information table (LIT)	It shall inform the instructions related to local events such as discrimination per hour, name and explanation about the event (type of scenery etc.)
Event relation table (ERT)	It shall indicate relationship between programs or local events, such as groups and attributes of programs and local events
Index transmission table (ITT)	It shall describes information related to programs indexes when broadcasting the programs
Partial content announcement table (PCAT)	It shall announce a partial content included in data broadcasting
Stuffing table (ST)	It shall be used to make the other tables invalid
Broadcaster information table (BIT)	It shall designate the broadcaster units and SI parameters for each broadcaster unit
Network board information table (NBIT)	It shall be convey network board information and reference information to get the network board information
Linked description table (LDT)	It shall be conveys information about reference to other tables

Table 3 — Tables used in digital broadcasting which are not part of SI

Table name	Function
Selection information table (SIT) [a]	It shall convey information related to programs transmitted by a partial transport stream
Discontinuity information table (DIT) [a]	It shall convey instructions about switching points of possibly discontinued information services transmitted by a partial transport stream (TS)
Download control table (DCT)	It shall convey various information to separate and extract DLTs (Download tables)
Download table (DLT)	It shall convey the receivers update systems
Software download trigger table (SDTT) [a]	It shall convey instructions about receivers' updates notification, such as "service_id" used for the update, schedule and receiver types which shall be covered by the update
Common data table (CDT) [a]	It shall conveys data such as company's logo marks which are commonly required for receivers and shall be stored in non-volatile memory
Application information table (AIT)	It shall convey dynamic control information concerning ARIB-J applications and additional information for the execution
[a] Tables specified in standard ARIB STD-B21.	

Table 4 — Functions of tables used in digital broadcasting excluding SI

Function name	Description
ECM	Conveys information consisting of program information (information related to programs, signal descrambling keys etc.) and control information (turn on/turn off, descrambling signal receiver)
EMM	Conveys individual information which shall include contract information for each subscriber and work key to decrypt common information
DSM-CC section [a]	Conveys information notification such as the update service ID, schedule and receivers types which shall be covered by the update
[a] Tables specified in ARIB STD-B24.	

6 Basic service information description

The basic service information description shall be according to the EN 300 468.

For each service in the multiplexer, PAT shall indicate the location (PID value of transport stream packets) for the related PMT (program map table), and PAT shall provide the location of NIT (Network Information table) too. ST (stuffing table) shall be used for make invalid existing sections.

CAT shall indicate information for conditional access system used in the multiplexer. The information shall be understood as private (not defined in this standard) and relies on the CA system, however, when necessary, includes the location of EMM stream.

PMT shall identify and indicates the location of stream related to any transmitted services, and the location of program clock reference field (PCR) for a service.

The NIT is defined to provide information related to the physical network.

NOTE The NIT location is defined in this Standard according to ISO/IEC 13818-1, however the data format is different of the forecast in ISO/IEC 13818-1.

Besides PSI, it shall be necessary provide event and service identification data for the user. The data coding shall be according to specified in this Standard.

Unlike PAT, CAT, and PMT of the PSI, which provide information only for the multiplexer in which they are contained (the actual multiplexer), the additional information defined in this standard can also provide information on services and events carried by different multiplexers, and even on other networks. These data is structured in eleven tables as defined below:

a) bouquet association table (BAT): provide information related to bouquets;

b) service description table (SDT): it has information which describe the service in a system such as service name and provider name;

c) event information table (EIT): it has a lot of information related to a program or a event, such as: event name, event start, durarion etc. By using these descriptors, it is possible describe different types of events in a same transmission;

d) running status table (RST): provides the status for each event (running/not running). RST updates this information and allows the automatic switching between events;

e) time and date table (TDT): provides information related to current data and time. This information is provided in other table due to the update frequency of this information;

f) time offset table (TOT): provides the information related to the current data and time and also the time offset information.

EXAMPLE the official time to be used in Brazil shall be UTC-3 and the adjustments for daylight saving time (DST) and time zone shall be defined in the local_time_offset_descriptor in accordance with the operational guidelines

g) partial content announcement table (PCAT): describes the start time and current time of the partial data content;

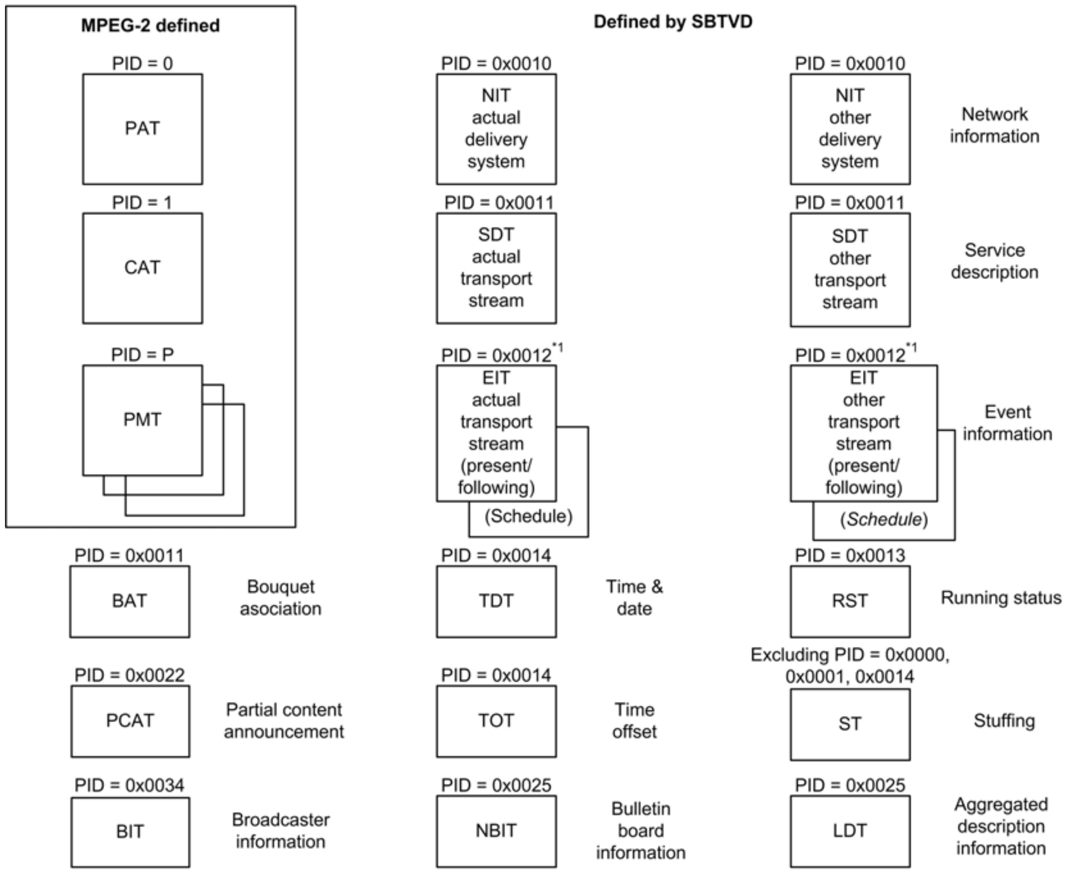
h) stuffing table (ST): it shall be according to EN 300 468;

i) broadcast information table (BIT): describes the network information or SI broadcasting parameter information for each broadcaster;

j) network board information table (NBIT): describes the network board information and the reference information about how to get the network group information;

k) link description table (LDT): describes data to be used as reference in other tables.

The descriptors usage shall allow a flexibilization of table structure and shall allow future compatible extensions (see Figure 1).



*1: Using PID values 0x0012, 0x0026, 0x0027 for digital terrestrial transmission

Figure 1 — Structure of transmission control signals

7 SI Tables

7.1 SI tables mechanisms

7.1.1 General

The SI tables specified in this Standard and the tables MPEG2-PSI shall be divided in one or more sections before to be inserted in the transport stream packets. The tables listed in Clause 6 shall be generated in a way that they never need to be regenerate in the STB. The tables, when transmitted, shall never be scrambled, except EIT, which can be scrambled if necessary. A section is a structure which shall be always according to the syntax used for the mapping of all MPEG-2 and SI tables specified in this standard, into the transport stream packets. The mechanisms of SI tables shall meet the ISO/IEC 13818-1.

7.1.2 Explanation

The sections in each table may be variable in length and shall be limited to 1024 bytes, except for sections in the EIT, which shall be limited to 4096 bytes. Each section is uniquely identified by the combination of the following elements:

a) table_id: shall identify to which table the section belongs. The list of values of existent table_id shall be according to Table 6;

NOTA Some table_ids may be according to ISO standards or others documents presented by the present document. Other values of table_id can be allocated by the user for private purposes.

b) table_id_extension: shall be used to identificate a subtable. The interpretation of each subtable is showed in 7.2;

c) section_number: shall allow that the sections of a subtable to be able to reassembled in their original order by the decoder. Sections should be transmitted in numerical order, unless it is desired to transmit some sections of a subtable more frequently than others. For the SI tables specified in this document, the section numbering applies to the other subtables;

d) version_number: when the characteristics of the TS described in the SI change (such as new events start, different composition of elementary stream for given service), then new SI data shall be sent containing the updated information. A new version of the SI data shall be signaled by sending a sub_table with the same identifiers as the previous sub_table containing the relevant data, but with the next value of version_number. For the SI tables specified in this document, the version_number applies to all sections of a sub_table;

e) current_next_indicator: each section shall be numbered as valid “now” (current), or as valid in the immediate future (next). This indicator shall allow the transmission of a future version of the SI in advance of the change, providing to the decoder with the opportunity to prepare for the change. There is however, no requirement to transmit the next version of a section in advance, but if it is transmitted, then it shall be informed as the next correct version of that section.

7.1.3 Section mapping of a transport stream (TS) packet

The sections shall be mapped directly in the TS packets and can start in the beginning of the payload of the TS packet, but it’s not mandatory, because the beginning of the first section in the payload of TS packet is pointed by *pointer_field*. Never there is more than one *pointer_field* in a TS packet, so the beginning of any other section can be identified through the length of the first section or any sections which come after, since there isn’t any gap between sections in the TS packet.

In the TS packets of any value of a unique PID, a section shall finish before be allowed the beginning of the next one, or won’t be possible identify to which section header the data belongs. The section finishes before the end of the TS packet, and if is not convenient to another open section, a stuffing mechanism can be used to fill the space.

The stuffing can be done through the stuffing of each byte remaining of the TS packet with the value ‘’0xFF’’. So the value ‘’0xFF’’ shall not be used for the table_id. If the byte immediatelly next to the last byte of the section takes the value ‘’0xFF’’, and the rest of the packet shall be filled with ‘’0xFF’’ bytes. These bytes can be discarded by a decoder. The stuffing can be done using the adaptation_field mechanism.

7.1.4 PID and table_ID allocation

Table 5 lists the PID values which shall be used for the TS packets which carry SI sections.

Table 5 — PID allocation for SI

Table	PID
PAT [a]	0x0000
PMT [a]	Indirect designation by PAT
CAT [a]	0x0001
NIT [a]	0x0010
SDT	0x0011
BAT	0x0011
EIT	0x0012
EIT (digital terrestrial television broadcasting)	0x0012, 0x0026, 0x0027
RST	0x0013
TDT	0x0014
TOT	0x0014
PCAT	0x0022
BIT	0x0024
NBIT	0x0025
LDT	0x0025
ST	Exclude 0x0000, 0x0001, 0x0014
Null packets [a]	0x1FFF

[a] According to ARIB STD-B10.

Table 6 lists the values, which shall be used for table_id and transmission level for the SI, defined in this Standard.

The values specified as sending frequency in Table 6 are only criteria of operation and not the standard values.

Table 6 — Transmission levels and table_id value allocation

Table_id	Table	Transmission level	Transmission frequency
0x00	PAT	Mandatory	Once or more/100 ms
0x01	CAT	Mandatory	Once or more/1 s
0x02	PMT	Mandatory	Once or more/100 ms
0x40	NIT (actual network)	Mandatory	Once or more/10 s
0x41	NIT (other network)	Optional	Once or more/10 s
0x42	SDT (actual stream)	Mandatory	Once or more/2 s
0x46	SDT (other stream)	Optional	Once or more/10 s
0x4A	BAT	Optional	Once or more/10 s
0x4E	EIT (present and following program of the actual stream)	Mandatory	Once or more/2 s
0x4F	EIT (present and following program of the actual stream)	Optional	Once or more/10 s
0x50 – 0x5F	EIT (program within 8 days of the actual stream)	Optional	Once or more/10 s
	EIT (Program after 8 days of the actual stream)	Optional	Once or more/30 s
0x60 – 0x6F	EIT (program within 8 days of the other stream)	Optional	Once or more/10 s
	EIT (program after 8 days of the other stream)	Optional	Once or more/30 s
0x70	TDT	Optional	Once or more/30 s
0x71	RST	Optional	Optional
0x72	ST	Optional	Optional
0x73	TOT	Mandatory	Once or more/30 s
0xC2	PCAT	Optional	Optional
0xC4	BIT	Optional	Once or more/20 s
0xC5	NBIT (board information body)	Optional	Once or more/20 s
0xC6	NBIT (reference information to gain board information)	Optional	1 s or more 10 s
0xC7	LDT	Optional	1 s or more 10 s
0x90 – 0xBF	Selectable range as table_id values set by companies		

7.1.5 Table repetition rates and ramdom access

In systems where random access is considered, it is recommended to retransmit SI sections specified within this document several times, even when changes do not occur in the configuration. For the SI specified within this document, multi-sectional availability in the same sub_table section shall not exceed 4KB. (Multi-sectional availability means continuous allocation to TS packets).

Moreover, TS packets of the same PID shall be transmitted within the range of 4 KB + 100 % in 32 ms each. The rule of “4 KB in 32 ms” is a detailed specification of 1 Mbit/s for every PID. This limit applies for TSs with a total band of up to 100 Mbit/s.

7.1.6 Data scrambling

With the exception of the EIT carrying schedule information, no other tables specified in this Standard shall be scrambled. One method for scrambling the EIT schedule table is informed in the appendix of this document. If a scrambling method operating over a TS packet is used, it may be necessary to use a stuffing mechanism to fill from the end of a section to the end of a packet, so that any transitions between scrambled and unscrambled data occur at packet boundaries.

In order to identify the CA systems which control the descrambling of the EIT data, a scrambled EIT schedule table shall be identified in the PSI. Service_id value 0xFFF shall be allocated to identifying a scrambled EIT, and the program map section for this service shall be described in the EIT as a private stream and shall include one or more CA_descriptors, according to ISO/IEC 13818-1, which defines PID values and optionally, other private data to identify the associated CA streams. Service_id value 0xFFF shall not be used for any other service.

7.2 Tables and data structures

7.2.1 Program association tables (PAT)

7.2.1.1 General information

PAT shall describe the relation between the program_number and the PID value of transport stream packet carrying the program definitions. This program_number is the numeric tag associated to a program. This table shall be according to ISO/IEC 13818-1.

7.2.1.2 Program association section

The total PAT shall be compounded by one or more sections with the following syntax below (see Table 7). It can also be set to use multiple sections.

Table 7 — PAT

Syntax	Number of bits	Identifier
program_association_section() {		
table_id	8	uimbsf
section_syntax_indicator	1	bslbf
‘0’	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<N;i++){		
program_number	16	uimsbf
Reserved	3	bslbf
if(program_number == ‘0’){		
network_PID	13	uimsbf
}		
else{		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchof
}		

7.2.1.3 Field semantics

The field semantics of the program association table (PAT) shall be the follow one:

- table_id: 8 bits field , which shall be according to the Table 6;
- section_syntax_indicator: 1 bit field which always shall be set to “1”;
- section_length: 12 bits field, the first two bits of which shall be ‘00’. The remaining 10 bits inform the number of bytes of the section, starting immediately following the section_length field and including the CRC. The value in this field shall not exceed to 1021 (0x3FD);
- transport_stream_id: 16 bits field whose function shall be the identification flag of a TS of any other multiplexer present in the network. This value shall be set by the user;

- version_number: 5 bits field which corresponds to the version number of the entire program association table. The version number shall be increased by 1 until 32, every time a PAT definition changes. When the current_next_indicator field is set to ‘1’, then the version_number shall be that of the next applicable PAT to become valid. When the field current_next_indicator is set to ‘0’, so the version_number field shall be the one of the next applicable PAT to become valid;

- current_next_indicator: 1 bit indicator, when set to ‘1’ shall indicate that the program association table sent is currently valid and applicable. When this bit is set to ‘0’, it shall indicate that the table sent is not applicable and the system shall wait the next valid table;

- section_number: 8 bits field shall inform the number of the section. The section_number field of the first PAT section shall be 0x00. It shall be incremented by 1 each additional section in the PAT;

- last_section_number: 8 bits field shall specify the number of the last section, that is, the section with the highest section_number of a PAT;

- program_number: 16 bit field which specifies the program to which the program_map_PID is applicable. When having 0x000 value, then the next reference PID shall be the network PID. For all the other cases this field value may be set by the user. This field value shall not be repeated within a same PAT;

- network_PID: 13 bits field used only in conjunction with when the program_number is 0x0000, specifying the TS packets PID which shall contain the network information table (NIT). The network_PID value is set by the user, but it shall only have values in accordance with the specified in Table 8;

- program_map_PID: 13 bitS field which specifies the TS packets PID which shall contain the applicable program_map_section for the specified program by program_number. None of program_number shall have more than one program_map_PID allocation. The program_map_PID value is set by the user, but shall have values in accordance with the specified in Table 7;

- CRC_32: 32 bits field contains the CRC value that provides a zero output of the registers in the dedoder, according to Annex B, after processing the enire program association section.

Table 8 — PID Table

Value	Description
0x0000	Program association table
0x0001	Conditional access table
0x0002 - 0x000F	Reserved
0x0010 - 0x1FFE	It can be used as network_PID, program_map_PID, elementary_PID or other purposes
0x1FFF	Null packets
NOTE TS packets with PID values 0x0000, 0x0010 - 0x1FFFE can be used to carry PCR.	

7.2.2 Conditional access table (CAT)

7.2.2.1 General information

CAT informs the CA systems which shall be used and associate them to the EMM (entitlement management messages (EMM), describing any other special parameters associated to them. CAT shall be in accordance with ISO/IEC 13818-1.

7.2.2.2 Conditional access section

CAT table shall be composed by one or more sections with the syntax of Table 9.

Table 9 — CAT sections

Syntax	Number of bits	Identifier
CA_section() {		
table_id	8	uimbsf
section_syntax_indicator	1	bslbf
‘0’	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
Reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

7.2.2.3 Field semantics

Definition of field semantics in the conditional access section:

— table_id: 8 bits field which shall be set to 0x01 in accordance with Table 6;

— section_syntax_indicator: 1 bit field which shall be set to “1”;

— section_length: 12 bitS field, the first two bits of which shall be ‘00’. The remaining 10 bits inform the number of bytes of the section, starting immediately following the section_length field and including the CRC. The value in this field shall not exceed to 1021 (0x3FD);

- version_number: 5 bits field corresponds to the version number of the entire program association table. The version number shall be incremented by 1 until 32 every time a PAT definition changes. When the current_next_indicator field is set to ‘1’, then the version_number shall be that of the next applicable PAT to become valid;
- current_next_indicator: 1 bit indicator, when set to ‘1’ indicates that the program association table sent is currently valid and applicable. When this bit is set to ‘0’, it indicates that the table sent is not applicable and the system shall wait the next valid table;
- section_number: 8 bits field informs the number of the section. The section_number field of the first CAT section shall be 0x00. It shall be incremented by 1 each additional section in the CAT;
- last_section_number: 8 bits field specifies the number of the last section, that is, the section with the highest section_number of a CAT;
- CRC_32: 32 bits field contains the CRC value that provides a zero output of the registers in the dedocder, according to Annex B, after processing the entire program association section.

7.2.2.4 Possible descriptors in CAT

The descriptors that may appear in CAT are:

- conditional access descriptor;
- conditional playback descriptor;
- CA service descriptor.

7.2.3 Program map table (PMT)

7.2.3.1 General information

The program map table designates mapping between program numbers and the program elements which compound them. A simple instance of this mapping type is called “program definition”. The program map table shall be the entire collection of all the program definitions in a transport stream. This table shall be transmitted in packets using PID values selected by the encoder. More than one PID may be used if desired. PMT specification shall be in accordance with ISO/IEC 13818-1.

7.2.3.2 Program map section

PMT shall be compounded by one or more sections with the following syntax given on Table 10. it can also be segmented to use multiple sections. Each section number shall always be set to zero. Sections shall be identified by program_number field.

Table 10 — PMT sections

Syntax	Number of bits	Identifier
TS_program_map_section() {		
table_id	8	uimbsf
section_syntax_indicator	1	bslbf
‘0’	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
PCR_PID	13	uimsbf
Reserved	4	bslbf
program_info_length	12	uimsbf
for(i=0,i<N,i++){		uimsbf
descriptor()		
}		
for(i=0,i<N1,i++){		
stream_type	8	uimsbf
Reserved	3	bslbf
elementary_PID	13	uimsbf
Reserved	4	bslbf
ES_info_length	12	uimsbf
for(i=0,i<N2,i++){		
Descriptor()		
}		
}		
CRC_32	32	rpchof
}		

7.2.3.3 Possible descriptors in the PMT

The descriptors that may appear in the PMT are:

- conditional access descriptor;
- copyright descriptor;
- country availability descriptor;
- linkage descriptor;
- component descriptor;
- mosaic descriptor;
- stream identifier descriptor;
- parental rating descriptor;
- hierarchical transmission descriptor;
- digital copy descriptor;
- emergency information descriptor;
- data component descriptor;
- system management descriptor;
- target area descriptor;
- video decode control descriptor;
- content availability descriptor;
- carousel compatible composite descriptor;
- conditional playback descriptor;
- AVC video descriptor;
- AVC and HRD timing descriptor;
- AAC audio descriptor.

7.2.4 Network information table (NIT)

7.2.4.1 General information

The NIT conveys information relating to the physical organization of the TS grouping existing in a same network, and its characteristics, as well as all relevant data on the combination of the existing services.

NIT (see Table 11) also loads information related to the physical organization of multiplexers/TS carried by a given network and the characteristics from the proper network. Combination of original_network_id and transport_stream_id allows to each TS to be identified only by all area of application of the present document. Individual values of network_id are assigned to each network, which serve as unique identification codes for the networks. The standardization organization shall be the responsible one for specifying the allocation of these codes. In an event in which NIT is transmitted in a network of which the TS has been originated, network_id and original_network_id shall have value the same. NIT usage is mandatory.

Table 11 — NIT sections

Syntax	Number of bits	Identifier
network_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
network_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

7.2.4.2 Network information section

Network information section shall be in accordance with Table 11.

NIT semantics shall be the following:

- table_id: 8-bit field which identifies table. It shall be set to 0x40 for actual network NIT and 0x41 for other network NIT, in accordance with Table 6;

- section_syntax_indicator: the section_syntax_indicator is a 1 bit field which shall be set to “1”;

- section_length: 12 bits field, the first two bits of which shall be ‘00’. It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The value in this field shall not exceed to 1 021 so that the entire section has a maximum length of 1 024 bytes;

- network_id: 16 bits fiel with unique value of network identification;

- version_number: 5 bits field that shall represent the subtable version number. The version_number shall be incremented by 1 every time the information carried inside the subtable changes. When the value is 31, it shall return to 0. When the current_next_indicator is set to ‘1’, then the version_number shall be that of the current subtable defined by table_id and network_id. When the current_next_indicator is set to ‘0’, then the version_number shall be that of the next applicable subtable defined by the table_id and network_id;

- current_next_indicator: 1 bit indicator which, when set to ‘1’, indicates that the subtable is applicable in the moment. When this bit is set to ‘0’, it indicates that the table sent is not yet applicable and the system shall wait the next valid table;

- section_number: 8 bits field informs the number of the section. The section_number field of the first subtable section shall be 0x00. It shall be incremented each additional section keeping unchanged the values of fields: table_id, event_id, service_id, transport_stream_id, and original_network_id;

- last_section_number: 8 bits field specifies the number of the last section of the subtable which this section belongs to;

- network_descriptors_length: 12 bits field which provides the network descriptor total length in bytes;

- transport_stream_loop_length: 12 bits field which specifies total length in bytes of TS loop, which ends immediately before the first CRC-32 byte;

- transport_stream_id: 16 bits field which serves as a label for identification of this TS, making it different from any other multiplex within the delivery system;

- original_network_id: 16 bits field provides the label identifying the delivery system origin and shall have the same value of the network_id;

- transport_descriptors_length: 12 bits field specifying the total length in bytes of TS descriptor;

- CRC_32: 32 bits field contains the CRC value as specified in Annex B.

7.2.4.3 NIT descriptors

The descriptors that may appear in the NIT are:

- network name descriptor;
- system management descriptor;
- service list descriptor;
- stuffing descriptor;
- linkage descriptor;
- terrestrial delivery system descriptor;
- emergency information descriptor;
- partial reception descriptor;
- TS information descriptor.

7.2.5 Bouquet association table (BAT)

7.2.5.1 Bouquet association section

BAT shall provide information related to bouquets. A bouquet shall be a collection of services permeating a network.

BAT shall be segmented in bouquet_association_sections, according Table 12 syntax. Any section that is part of BAT shall be broadcasted in TS packets with PID equal to 0x0011. A BAT sub-table sections describing a given bouquet shall have a "bouquet_id" field receiving the value designated to bouquet described on other location.

All BAT sections shall be identified with a table_id value equal to 0x4A.

Table 12 — Bouquet association selection

Syntax	Number of bits	Identifier
bouquet_association_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
bouquet_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
bouquet_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
Transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
Descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics of bouquet association section information shall be:

- *table_id:* see Table 6;
- *bouquet_id:* 16 bits field using used as bouquet identification label. Value determination for this field is specified in the future and shall be in accordance with EN 300 468.

7.2.5.2 BAT descriptor

The descriptors that may appear in BAT may be:

- service list descriptor;
- stuffing descriptor;
- bouquet name descriptor;
- country availability descriptor;
- linkage descriptor;
- CA identifier descriptor.

7.2.6 Service description table (SDT)

7.2.6.1 Service description section

Each SDT sub-table (see Table 13) shall describe services contained into specific TS. Services can make part of current TS or can make part of other TS, being differentiated by table_id (see Table 6).

SDT shall be segmented in service_description_sections (service description section) that uses syntax of Table 13. Any section forming part of a SDT shall have to be transmitted in TS packets with a PID value of 0x0011. Any SDT section describing current TS (that is, TS containing SDT) shall have table_id specified with a value of 0x42 and with the same table_id_extension (transport_stream_id) and with the same original_network_id. Any SDT section belonging to TS other than actual TS shall take a table_id value of 0x46.

Table 13 — Service description section

Syntax	Number of bits	Identifier
service_description_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
reserved_future_use	8	bslbf
for(i=0;i<N;i++){		
service_id	16	uimsbf
reserved_future_use	6	bslbf
EIT_schedule_flag	1	bslbf
EIT_present_following_flag	1	bslbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for service description section information shall be:

- table_id: see Table 6;
- EIT_user_defined_flags: each broadcaster can define this 3 bits field individually as the extension to indicate what type of EIT will be transmitted. If it is set to "111", it means that it will not be used;
- EIT_schedule_flag: 1 bit field which, when set to "1", indicates that event information service (EIT) is present in the current TS. If the flag is set to 0 the event information service (EIT) may not be present in TS;
- EIT_present_following_flag: 1 bit field which, when set to "1", indicates that EIT_present_following information is present in current TS (see guideline for information on maximum interval between a sub-table occurrences). If the flag is set to 0, the present/following information for EIT service should not be present in TS;
- running_status: 3 bits field indicating status of the service, as defined in Table 14.

Table 14 — Service description section

Value	Meaning
0	Undefined
1	Not running
2	Starts in a few minutes
3	Pausing
4	Running
5-7	Reserved for future use

7.2.6.2 Possible SDT descriptors

The descriptors that may appear in the SDT are:

- service descriptor;
- stuffing descriptor;
- bouquet name descriptor;
- country availability descriptor;
- linkage descriptor;
- NVOD reference service descriptor;
- service and time shifting descriptor;
- mosaic descriptor;
- CA identifier descriptor;
- digital copy control descriptor;
- logo transmission descriptor;
- content availability descriptor.

7.2.7 Event information table (EIT)

7.2.7.1 Event information section

EIT (see Table 15) shall provide information in chronological order related to events contained into each service.

There were identified four EIT classifications, distinguishable by using different table_id (see Table 6) as below:

a) actual TS, present/following event information = table_id = "0x4E";

b) other TS, present/following event information = table_id = "0x4F";

c) actual TS, event schedule information = table_id = "0x50" for "0x5F";

d) other TS, event schedule information = table_id = "0x60" to "0x6F".

The present/following table shall contain only information pertaining to the present event while the following event may be transmitted in the actual TS or other TS, except in the case of a “near video on demand” service (NVOD), where it may have more than two event descriptions. Event schedule tables for either the actual TS or other TS contain a list of events, including events taking place at some time beyond the next event. Event tables are optional. Event information shall be chronologically ordered.

EIT shall be segmented into *event_information_sections* using the syntax of Table 15. Any sections forming part of an EIT shall be transmitted in TS packets with a PID value of 0x0012.

Table 15 — Event information section

Syntax	Number of bits	Identifier
event_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
segment_last_section_number	8	uimsbf
last_table_id	8	uimsbf
for(i=0;i<N;i++){		
event_id	16	uimsbf
start_time	40	bslbf
Duration	24	uimsbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<N;j++){		
Descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics of information for event information section shall be according to EN 300 468:2007, subclause 6.2.4; except for the fields below:

- table_id: see Table 6;

- *start_time*: 40-bit field that shall contain the start time of the event in the UTC -3 and the date in “modified julian date” (MJD) (see Annex A). This field shall be coded as 16 bits giving the 16 LSBs of MJD proceeded by 24 bits coded as 6 digits in 4-bit Binary Coded Decimal (BCD). If the start time is undefined (for example, for reference of a NVOD event), all bits of the field shall be set to "1";

EXAMPLE 93/10/13 12:45:00 → 0xC079124500, where C079 is the MJD and 124500 is the UTC-3

- duration**:** 24 bits field that shall contain the event duration in hours, minutes and seconds. When duration is not defined (as emergency news), all bits in this field shall be set to "1";

- format**:** 6 digits, 4 bits BCD = 24 bits;

EXAMPLE 01:45:30 is coded as "0 x 014530".

- running_status: 3 bits field that shall indicate the status of an event. Its definitions can be seen in Table 14.

The specification of table EIT types (H, M or L) are in Annex I.

7.2.7.2 Possible EIT descriptors

The desriptors that may appear in EIT are:

- stuffing descriptor;

- linkage descriptor;

- short event descriptor;

- extended event descriptor;

- event shifting descriptor;

- component descriptor;

- CA identifier descriptor;

- content descriptor;

- parental rating descriptor;

- digital copy control descriptor;

- audio component descriptor;

- hyperlink descriptor;

- data content descriptor;

- series descriptor;

- event group descriptor;

- component group descriptor;
- LDT linkage descriptor;
- content availability descriptor;
- carousel compatible composite descriptor.

7.2.8 Time and date table (TDT)

The TDT (see Table 16) shall carry only the time and date information.

TDT shall consist of a single section using the syntax of Table 16. This TDT section shall be transmitted in data stream packets with a PID value of 0x0014, and the table_id shall have a value of 0x70.

Table 16 — Time and date section

Syntax	Number of bits	Identifier
time_date_section(){		
table_id	8	Uimsbf
section_syntax_indicator	1	Bslbf
reserved_future_use	1	Bslbf
reserved	2	Bslbf
section_length	12	Uimsbf
UTC_time	40	Bslbf
}		

Semantics for the time and date section shall be:

- table_id: see Table 6;
- section_length**:** This is a 12-bit field, which shall be set to "0x005". It specifies the number of bytes of the section, starting immediately following the section-length and up to the end of the section;
- UTC_time: (Current time and date): This 40-bit field contains the current time and date in Brazil (UTC-3) and MJD (see Annex A). This field is coded as 16-bit giving the 16 LSBs of MJB followed by 24 bits coded as 6 digits in a 4-bit BCD.

EXAMPLE 93/10/13 12:45:00 is coded as "0xC079124500".

NOTE As the field of MJD has 16 bits, the current date may be indicated until April 22nd, 2038.

7.2.9 Time offset table (TOT)

7.2.9.1 Date and time offset sections

The TOT (see Table 17) shall carry the time and date information and local time offset. The TOT shall consist of a single section using the syntax of Table 17. This TOT section shall be transmitted in TS packets with a PID value of 0x0014, and the table_id shall take the value 0x73.

Table 17 — Date and time offset sections

Syntax	Number of bits	Identifier
time_offset_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
UTC_time	40	bslbf
reserved	4	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

Semantics for date and time offset section shall be:

— table_id**:** see Table 6;

— UTC_time: (current time and date): 40 bits field shall contain the current time and date in Brazil (UTC-3) and MJD (see Annex A). This field is coded as 16 bits giving the 16 LSB of MJB followed by 24 bits coded as 6 digits for time in BCD. This section shall be in according with EN 300 468:2007, subclause 6.2.6.

EXAMPLE 93/10/13 12:45:00 is coded as “0xC079124500”

NOTE As the field of MJD has 16 bits, the current date may be indicated until April 22nd, 2038.

7.2.9.2 Possible TOT descriptors

The descriptor of TOT used in TOT, to submit information of change in the time is local_time_offset_descriptor.

7.2.10 Running status table (RST)

The RST (see Table 18) allows accurate and fast updating of the timing status of one or more events. This may be necessary when an event starts early or late due to scheduling changes. The use of a separate table shall be necessary in order to enable a faster updating.

RST table shall be segmented in running_status_sections using syntax of Table 18. Any sections forming part of an RST shall be transmitted in TS packets with a PID value of 0x0013, and table_id field shall be set to 0x71.

Table 18 — Running status section

Syntax	Number of bits	Identifier
running_status_section(){		
Table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
reserved_future_use	5	bslbf
running_status	3	uimsbf
}		
}		

Semantics for running status section shall be:

- table_id: see Table 6;
- running_status: 3 bits field indicating status of the event, as defined in Table 14.

7.2.11 Stuffing table (ST)

The purpose of this section (see Table 19) is to invalidate sections at a delivery system boundary. When one section of a sub-table is overwritten, then all sections of that sub-table shall also be overwritten (stuffed) in order to retain the integrity of the section_number field.

Table 19 — Stuffing section

Syntax	Number of bits	Identifier
stuffing_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
for(i=0;i<N;i++){		
data_byte	8	uimsbf
}		
}		

Semantics for the stuffing section shall be:

— table_id: this field shall take the value "0x72" according to Table 6.

7.2.12 Partial content announcement table (PCAT)

PCAT (see Table 20) shall be used to inform the access terminal start time of a data transmission by broadcaster, to partially update the accumulated content in access terminal, that is, performs a content update scheduling.

Table 20 — Partial content announcement table

Syntax	Number of bits	Identifier
partial_content_announcement_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
content_id	32	uimsbf
num_of_content_version	8	uimsbf
for(i=0;i<num_of_content_version;i++){		
content_version	16	uimsbf
content_minor_version	16	uimsbf
version_indicator	2	bslbf
reserved_future_use	2	bslbf
content_descriptor_length	12	uimsbf
reserved_future_use	4	bslbf
schedule_description_length	12	uimsbf
for(j=0;j<N;j++){		
start_time	40	bslbf
duration	24	uimsbf
}		
for(j=0;j<N2;j++){		
descriptors()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for partial content announcement section shall be:

- table_id**:** see Table 6;

- section_syntax_indicator**:** 1 bit field which shall be set to "1";

- section_length**:** 12 bits fields that shall specifie the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes;

- service_id**:** 16 bits field that shall indicate the event service_id that announces partial original data broadcasting program and partial data. The service_id shall be the same as the program_number in the corresponding program_map section;

- version_number**:** 5 bits field that shall be the version number of the next subtable**.** The version_number shall be incremented by 1 when a change in the information carried within the sub-table occurs. When the value reaches 31, it shall wrap around to 0;

- current_next_indicator: 1 bit indicator that, when set to "1", shall indicate that the sub-table shall be the currently applicable sub-table;

- section_number: 8 bits field that shall indicate the number of the section;

- last_section_number: 8 bits field that shall specify the number of last section of the sub-table of which this section is part;

- transport_stream_id: 16 bits field which shall serve as a label to identify the TS, from any other multiplexer within delivery system;

- original_network_id: 16 bits field that shall store the label which identifies the originating delivery system;

- ontent_id: 32 bits field that shall serve as a label identifying which partial content the partial data belongs to. The content_id shall be attibuted to the original transmitted data of the partial contents, so that it shall serve as a label to identify the contents in the service uniformly;

- num_of_content_version: 8 bits field that shall indicate the number of contents version announced in the table;

- content_version: 16 bits field that shall indicate the total contents version of the partial contents announced in the table;

- content_minor_version: 16 bits field that shall indicate partial contents version announced in the table;

- version_indicator: 2 bits field that shall indicate the meaning related to content version and contents minor version:

  00: whole version shall be the target (designation of contents version shall be invalid);

  01: target shall be the version after the designated version;

  02: target shall be the version before the designated version;

  03: target shall be only the designated version;

- content_descriptor_length: 12 bits field that shall indicate the total length in bytes of the following schedule loop and descriptor loop;

- schedule_description_length: 12 bits field that shall indicate the total length in bytes of the following schedule loop;

- start_time: 40 bits field that shall indicate the start time of partial contents announcement by UTC-3 and MJD;

- duration: 24 bits field that shall indicate the duration of partial contents announcement by hours, minutes, and seconds;

- descriptor: shall store data contents descriptor in case of partial contents.

7.2.13 Broadcaster information table (BIT)

7.2.13.1 General information

The BIT table (see Table 21) designates broadcaster units, sending information on these units over all network and (SI) parameters to each existing broadcaster unit.

Table 21 — Broadcaster information section

Syntax	Number of bits	Identifier
broadcaster_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
original_network_id	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	3	bslbf
broadcast_view_propriety	1	bslbf
first_descriptors_length	12	uimsbf
for(i = 0;i< N1;i++){		
descriptor()		
}		
for(i = 0;i< N2;i++){		
broadcaster_id	8	uimsbf
reserved_future_use	4	bslbf
broadcaster_descriptors_length	12	uimsbf
for(k=0;k<N3;k++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for the broadcaster information section shall be:

- table_id: see Table 6;
- section_syntax_indicator: 1 bit field which shall be set to "1";
- section_length: 12 bits field that shall specify the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4093 so that the entire section has a maximum length of 4096 bytes;
- original_network_id: 16 bits field that shall store the label which identifies the network_id originating delivery system;
- version_number: 5 bits field that shall be the version number of the sub-table. The version_number shall be incremented by 1 when a change in the information carried within the sub-table occurs. When the value reaches 31, it shall wrap around to 0;
- current_next_indicator: 1 bit indicator that, when set to "1", shall indicate that the sub-table is the currently applicable sub-table.
- section_number: 8 bits field that shall indicate the number of the section;
- last_section_number: 8 bits field that shall specify the number of last section of the sub-table (that is, the section with the highest section_number) of which this section is part;
- broadcast_view_property: 1 bit field that shall indicate whether user indication with a unit of broadcaster name is appropriate (value "1") or not (value "0"), set according to broadcaster_id;
- first_descriptors_length: 12 bits field that shall indicate the total length in bytes of the following descriptor;
- broadcaster_id: 8 bits field that shall indicate broadcaster denoted with this loop;
- broadcaster_descriptors_length: 12 bits field that shall indicate the total length in bytes of the following descriptor;
- CRC_32: 32 bits field that shall contain CRC value, as specified in Annex B.

7.2.13.2 Possible BIT descriptors

The descriptors that may appear in BIT are:

- service list descriptor (the only one which is mandatory);
- SI parameters descriptor;
- broadcaster name descriptor;
- SI Prime TS descriptor;
- extended broadcaster descriptor.

7.2.14 Network board information table (NBIT)

7.2.14.1 Network board information sections

There are two types of NBIT as discriminated in the Table 6 (table_id) One of them describes the network board information and other gives the reference information to reach the network board. The network board information shall be transmitted to access terminal as an announcement to users, such as: service and/or genre type. With this information, the access terminal can include icons related to service and genre in the start of the message. The information title and contents are provided by board_information_descriptor in text format.

The syntax of the table NBIT is giving in Table 22.

Table 22 — Network board information section

Syntax	Number of bits	Identifier
network_board_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
original_network_id	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0;i<n;i++){		
information_id	16	uimsbf
information_type	4	uimsbf
description_body_location	2	uimsbf
reserved_future_use	2	bslbf
user_defined	8	bslbf
Number_of_keys	8	uimsbf
for(j=0;j<number_of_keys;j++){		
Key_id	16	uimsbf
}		
reserved_future_use	4	bslbf
descriptors_loop_length	12	uimsbf
For(j=0;j<m;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics for network board information section shall be:

- table_id: see Table 6;

- section_syntax_indicator**:** 1 bit field which shall be set to "1";

- section_length: 12 bits field that shall specify the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4 093 bytes so that the entire section has a maximum length of 4096 bytes;

- original_network_id: 16 bits field that shall give the label which identifies the network_id originating delivery system;

- version_number: 5 bits field that shall be the version number of the sub-table. The version_number shall be incremented by 1 when a change in the information carried within the sub-table occurs. When the value reaches 31, it shall wrap around to 0. When current_next_indicator is set to "1", then version_number shall be the one defined by current sub-table defined by table_id and network_id. When current_next_indicator is set to "0", then version_number shall be the one of following sub-table defined by table_id and network_id;

- current_next_indicator: 1 bit indicator that, when set to "1", shall indicate that the sub-table is the currently applicable sub-table. When the bit is set to "0", it shall indicate that the sub_table sent shall not be applicable yet and shall wait the next sub-table valid;

- section_number: 8 bits field that shall give the number of the section. The section number of the first section of sub-table shall be set to 0x00. Section number shall be incremented by 1 witch each additional section with the same table_id and network_id;

- last_section_number**:** 8 bits field that shall specify the number of last section of the sub-table (that is, the section with the highest section_number) of which this section is part;

- information_id: 16 bits field that shall indicate the ID number (allocated uniformly) of the submitted information;

- information_type: 4 bits field that shall indicate the submitted information type according to Table 23;

- description_body_location: 2 bits field that shall indicate the location of the table where contents of the information are described according to Table 24;

- user_defined: each broadcaster can define this 8-bit field individually;

- number_of_keys: 8 bit field that shall indicate the number of next *key_id*;

- key_id: 16 bits field that shall describe key_id according to Table 23;

- descriptors_loop_length: 12 bits field that shall give the total length in bytes of the following descriptor;

- CRC_32: 32 bits field that shall contain CRC value, as specified in Annex B.

Table 23 — Information type

Value	Description	Key_id
0x0	Undefined	-
0x1	Information	None
0x2	Information with service_id	Service_id
0x3	Information with genre	Content_nibble, user_nibble
0x4-0xF	Reserved for future use	-

Table 24 — Description body location

Value	Description
00	Undefined
01	Detail information is described in the actual TS table
10	Detail information is described in SI prime TS table
11	Reserved for future use

7.2.14.2 Possible NBIT descriptors

The NBIT descriptors shall be:

- stuffing descriptor;
- board information descriptor.

7.2.15 Linked description table (LDT)

The LDT (see Table 25) shall be used to link various descriptions to refer from other tables.

Table 25 — Linked description section

Syntax	Number of bits	Identifier
linked_description_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
original_service_id	16	uimsbf
Reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
for(i=0;i<n;i++){		
description_id	16	uimsbf
reserved_future_use	12	bslbf
descriptors_loop_length	12	uimsbf
for(j=0;j<m;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Semantics of link description section shall be:

- table_id: see Table 6;
- section_syntax_indicator: 1 bit field which shall be set to "1";
- section_length: 12 bits fields that shall specify the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 4 093 bytes so that the entire section has a maximum length of 4096 bytes;
- original_service_id: 16 bits field that shall be the identification of the group which links descriptors in this sub-table using the service_id of the representing service. It is allocated uniformly within the network;
- version_number: 5 bits field that shall be the version number of the sub-table. The version_number shall be incremented by 1 when a change in the information carried within the sub-table occurs. When the value reaches 31, it shall be returned to 0. When current_next_indicator is set to "1", then the version_number shall be defined by actual sub-table, by table_id and network_id. When current_next_indicator is set to "0", then version_number shall be the one of following sub-table defined by table_id and network_id;

- current_next_indicator: 1 bit indicator that, when set to "1", shall indicate that the sub-table is the currently applicable sub-table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub-table to be valid;

- section_number: 8 bits field that shall give the number of the section. The section number of the first section of sub-table shall be set to 0x00. Section number shall be incremented by 1 witch each additional section with the same table_id and network_id;

- last_section_number: 8 bits field that shall specify the number of last section of the sub-table (that is, the section with the highest section_number) of which this section is part;

- transport_stream_id: 16 bits field which shall serve as a label to identify the TS, from any other multiplexer within delivery system;

- original_service_id: 16 bits field that shall be the label identifying the service_id of originating delivery system;

- description_id: This 16-bit field indicates the id_number of collected description (allocated uniformly within the representing service);

- descriptors_loop_length: 12 bits field that shall give the total length in bytes of the following descriptor;

- CRC_32: 32 bits field that shall contain CRC value, as specified in Annex B.

8 Table descriptors

8.1 Descriptor location and identification

The Table 26 lists the descriptors defined within this Standard, giving the intended placement within the SI tables. This does not imply that their use in other tables is restricted.

Table 26 — Location and requirements of SI descriptors

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT	SIT	SDTT	LIT	ERT	ITT
Conditional access descriptor [a]	Mandatory for conditional access	X	X													
Copyright descriptor [a]	[c]		X				x									
AVC *vídeo descriptor*	Optional		X													
AVC *timing and* HRD *descriptor*	Optional		X													
Network name descriptor [b]	Mandatory			X												
Service list descriptor [a]	Mandatory in NIIT (actual Network) Optional in NIT (other network) Mandatory in BAT Optional in BIT			X	X				X							
Stuffing descriptor	Optional			X	X	X	X			X	X					
Bouquet name descriptor	Mandatory in BAT				X	X										
Service descriptor [b]	Mandatory in SDT (actual stream) Opcional in SDT (other stream*)*					X										
Country availability descriptor	Optional		X		X	X										
Linkage descriptor	Optional		X	X	X	X	X									
NVOD *reference descriptor*	Mandatory for NVOD					X										
Time shifted service descriptor [b]	Mandatory in time shift event					X										
Short event descriptor	Optional						X									

Table 26 (continuation)

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT	SIT	SDTT	LIT	ERT	ITT
Extended event descriptor	Optional						X									
Time shifted event descriptor	Optional						X									
Component descriptor	Optional		X				X									
Mosaic descriptor	Optional		X			X										
Stream identifier descriptor	Optional		X													
CA *identifier descriptor*	Optional				X	X	X									
Content descriptor	Optional						X									
Parental rating descriptor	Optional		X				X									
Local_time_offset_descriptor	Mandatory for local_time_offset execution							X								
Hierarchical transmission descriptor	Mandatory for hieraquical transmission		X													
Digital copy control descriptor	Optional		X			X	X									
Audio component descriptor	Optional						X									
Hyperlink descriptor	Optional						X									
Target area descriptor	Optional		X													
Data contents descriptor	Optional						X									
Vídeo decode control descriptor	Optional		X													
TS *information descriptor*	Optional			X												

Table 26 (continuation)

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT	SIT	SDTT	LIT	ERT	ITT
Extended broadcaster descriptor	Optional								X							
Logo transmission descriptor	Optional					X										
Basic local event	Optional													X		
Reference descriptor	Optional						X							X		
Node relation descriptor	Optional														X	
Short node information descriptor	Optional						X								X	
STC (*system time clock*) *reference descriptor*	Optional															X
Series descriptor	Optional						X									
Event group descriptor	Optional						X									
SI *parameter descriptor*	Optional								X							
Broadcast name descriptor	Optional								X							
Component group descriptor	Optional						X									
SI *prime* TS *descriptor*	Optional								X							
Board information descriptor	Optional									X						
LTD *link descriptor*	Optional						X									
Connected transmission descriptor	Mandatory for linkage transmission			X												
Content availibility descriptor	Optional		X			X	X									

Table 26 (continuation)

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT	SIT	SDTT	LIT	ERT	ITT
Service group descriptor	Optional			X												
Carousel compatible composite descriptor [a]	Optional		X				X									
Conditional playback descriptor [a]	Mandatory in case of conditional playback	X	X													
Terrestrial delivery system descriptor [a]	Mandatory			X												
Partial reception descriptor [a]	Mandatory for one-seg reception			X												
Emergency information description [a]	Mandatory for alarm emergency transmission		X	X												
Data component descriptor [a]	Mandatory for data transmission		X													
System management descriptor [a]	Mandatory in NIT		X	X												
Carousel ID *descriptor* [e]	Mandatory for GINGA application		X													
Association tag descriptor [e]	Mandatory for GINGA application		X													
Deferred association tag descriptor [e]	Mandatory for GINGA application		X													
AAC *descriptor* [f]	Mandatory		X													

Table 26 (continuation)

Descriptor	Transmission level	CAT	PMT	NIT	BAT	SDT	EIT	TOT	BIT	NBIT	LDT	SIT	SDTT	LIT	ERT	ITT
Partial transport stream descriptor [g]	Optional											X				
Partial transport stream time descriptor [g]	Optional											X				
Network identifier descriptor [g]	Optional											X				
Content availability descriptor [g]	Optional												X			

[a] In accordance with ARIB STD-B10.

[b] Can be substituted with the descripotr defined by service provider, if it has at least the same function.

[c] Locations and requirements of descriptors shall be followed in a future standard of the Brazilian digital television systems.

[d] This is not applicable when using the function with conditional access descriptor.

[e] In accordance with ABNT NBR 15606-3

[f] In accordance with EN 300 468:2007, Anexo H.

[g] In accordance with ARIB STD-B21.

8.2 Value of identifiers

The value of identifiers specified in the Table 26 shall be in accordance with ABNT NBR 15603-1:2007, Table 9..

The “tag value” or table identifier value defined by an enterprise can be 0x80 >= x <= 0xBF.

The “tag value” or table identifier value defined by an enterprise shall be registered and disclosed as part of enterprise signal.

8.3 Descriptor coding

8.3.1 General information

When the construct "descriptor ()" appears in the sections of item 7.2, it shall indicate that zero or more of the descriptors defined within this subclause shall occur.

The following semantics shall be applied to all descriptors defined in 7.2:

— descriptor_tag: 8 bits field which value identifies each descriptor. These values are described in ISO/IEC 13818-1. The values of descriptor_tag shall be in accordance with ABNT NBR 15603-1:2007, Table 9;

— descriptor_length: 8 bits field that shall specify the total number of bytes of the data portion of descriptor following the byte defining the value of this field.

8.3.2 Bouquet name descriptor

The bouquet name descriptor shall provide the bouquet name in text form, according to Table 27.

Table 27 — Bouquet name descriptor

Syntax	Number of bits	Identifier
bouquet_name_descriptor(){		
Descriptor_tag	8	Uimsbf
Descriptor_length	8	Uimsbf
for(i=0;i<N;i++){		
Char	8	Uimsbf
}		
}		

The semantics for the bouquet name descriptor shall be:

— char: 8 bits field that shall contain the bouquet name described in BAT table. The text information shall be coded according the character map described the service provider operational standard.

8.3.3 Conditional access system identifier descriptor

The conditional access system identifier descriptor (see EN 300 468:2007, subclause 6.2.5) shall indicate when a particular bouquet, service or event is associated with a conditional access system and identifies the type of CA system by means of CA_system_id.

The semantics for the CA identifier descriptor shall be:

— CA_system_id: This 16-bit field shall identifie the conditional access system. The standardization organization shall specify the allocation of the values of this field.

8.3.4 Component descriptor

The component descriptor shall identify the type of component stream and can be used to provide a text description of the elementary stream. Structure details are on EN 300 468:2007, subclause 6.2.8.

The semantics for the component descriptor shall be:

— stream_content: 4 bits field that shall specify the stream type (video, audio or data). This field coding shall be specified in Table 28;

— component_type: 8 bits field that shall specify the type of video, audio or data component. This field coding shall be specified in Table 28;

— component_tag: 8 bits field that shall have the same value as component_tag field in the stream identification descriptor (see 8.3.15) for the component stream (if present in PSI program map section);

— ISO_639_language_code: 24 bits field that shall identify the language of the component (in case of audio or data) and a text description which may be contained in the descriptor. An ISO_639_language_code contains a 3 character code as specified by ISO 639-2. Each character shall be coded in 8 bits according to ISO/IEC 8859-15 and inserted in the order with the 24 bits field;

EXAMPLE Portuguese, the Brazilian official language, has 3-character code "por", which is coded as: “ 0111 0000 0110 1111 0111 0010”

- text_char: This is an 8-bit field. A word describes the text in the component stream. Text information is coded using the character sets and methods describes in operational standard of services provides.

Table 28 — Stream_content and component_type

Stream_content	Component_type	Description
0x00	0x00 – 0xFF	Reserved for future use
0x01	0x00	Reserved for future use
0x01	0x01	MPEG 2 Video 480i(525i), 4:3 aspect ratio
0x01	0x02	MPEG 2 Video 480i(525i), 16:9 aspect ratio with pan vectors
0x01	0x03	MPEG2 Video 480i(525i), 16:9 aspect ratio without pan vectors
0x01	0x04	MPEG 2 Video 480i(525i), > 16:9 aspect ratio
0x01	0x05 – 0xA0	Reserved for future use
0x01	0xA1	MPEG 2 Video 480p(525p), 4:3 aspect ratio
0x01	0xA2	MPEG 2 Video 480p(525p), 16:9 aspect ratio with pan vectors
0x01	0xA3	MPEG 2 Video 480p(525p), 16:9 aspect ratio without pan vectors
0x01	0xA4	MPEG 2 Video 480p(525p), > 16:9 aspect ratio
0x01	0xA5 – 0xB0	Reserved for future use
0x01	0xB1	MPEG 2 Video 1080i(1125i), 4:3 aspect ratio
0x01	0xB2	MPEG 2 Video 1080i(1125i), 16:9 aspect ratio, with pan vectors
0x01	0xB3	MPEG 2 Video 1080i(1125i), 16:9 aspect ratio, without pan vectors
0x01	0xB4	MPEG 2 Video 1080i(1125i), > 16:9 aspect ratio
0x01	0xB5 – 0xC0	Reserved for future use
0x01	0xC1	MPEG 2 Video 720p(750p), 4:3 aspect ratio
0x01	0xC2	MPEG 2 Video 720p(750p), 16:9 aspect ratio, with pan vectors
0x01	0xC3	MPEG 2 Video 720p(750p), 16:9 aspect ratio, without pan vectors
0x01	0xC4	MPEG 2 Video 720p(750p), > 16:9 aspect ratio
0x01	0xC5- 0xD0	Reserved for future use
0x01	0xD1	MPEG 2 Video 240p, 4:3 aspect ratio
0x01	0xD2	MPEG 2 Video 240p, 4:3 aspect ratio, with pan vectors
0x01	0xD3	MPEG 2 Video 240p, 4:3 aspect ratio, without pan vector
0x01	0xD4	MPEG 2 Video 240p, 4:3 aspect ratio > 16:9
0x01	0xD5- 0xE0	Reserved for future use
0x01	0xE1	MPEG 2 Video 1080p(1125p), 4:3 aspect ratio
0x01	0xE2	MPEG 2 Video 1080p(1125p), 16:9 aspect ratio, with pan vectors
0x01	0xE3	MPEG 2 Video 1080p(1125p), 16:9 aspect ratio, without pan vectors
0x01	0xE4	MPEG 2 Video 1080p(1125p), > 16:9 aspect ratio
0x01	0xE5 – 0xFF	Reserved for future use

Table 28 (continuation)

Stream_content	Component_type	Description
0x02	0x00	Reserved for future use
0x02	0x01	AAC MPEG2 audio, 1/0 mode (single mono)
0x02	0x02	AAC MPEG2 audio, 1/0 + 1/0 mode (dual mono)
0x02	0x03	AAC MPEG2 audio, 2/0 mode (stereo)
0x02	0x04	AAC MPEG2 audio, 2/1 mode
0x02	0x05	AAC MPEG2 audio, 3/0 mode
0x02	0x06	AAC MPEG2 audio, 2/2 mode
0x02	0x07	AAC MPEG2 audio, 3/1 mode
0x02	0x08	AAC MPEG2 audio, 3/2 mode
0x02	0x09	AAC MPEG2 audio, 3/2 + LFE mode
0x02	0x0A - 0x3F	Reserved for future use
0x02	0x40	AAC MPEG2 audio audio description for the visually impaired
0x02	0x41	AAC MPEG2 audio for the hard of hearing
0x02	0x42 - 0xAF	Reserved for future use
0x02	0xB0 - 0xFE	User-defined
0x02	0xFF	Reserved for future use
0x03 – 0x4F	0x00 – 0xFF	Reserved for future use
0x05	0x00	Defined by user Reserved for future use
0x05	0x01	H264/AVC video 480i(525i), 4:3 aspect ratio
0x05	0x02	H264/AVC video 480i(525i), 16:9 aspect ratio, with pan vectors
0x05	0x03	H264/AVC video 480i(525i), 16:9 aspect ratio, without pan vectors
0x05	0x04	H264/AVC video 480i(525i), > 16:9 aspect ratio
0x05	0x05 – 0xA0	Reserved for future use
0x05	0xA1	H264/AVC video 480p(525p), 4:3 aspect ratio
0x05	0xA2	H264/AVC video 480p(525p), 16:9 aspect ratio, with pan vectors
0x05	0xA3	H264/AVC video 480p(525p), 16:9 aspect ratio, without pan vectors
0x05	0xA4	H264/AVC video 480p(525p), > 16:9 aspect ratio
0x05	0xA5 - 0xB0	Reserved for future use
0x05	0xB1	H264/AVC video 1080i(1125i), 4:3 aspect ratio
0x05	0xB2	H264/AVC video 1080i(1125i), 16:9 aspect ratio, with pan vectors
0x05	0xB3	H264/AVC video 1080i(1125i), 16:9 aspect ratio, without pan vectors
0x05	0xB4	H264/AVC video 1080i(1125i), 16:9 aspect ratio
0x05	0xB5 – 0xC0	Reserved for future use
0x05	0xC1	H264/AVC video 720p(750p), 4:3 aspect ratio

Table 28 (continuation)

Stream_content	Component_type	Description
0x05	0xC2	H264/AVC video 720p(750p), 16:9 aspect ratio, with pan vectors
0x05	0xC3	H264/AVC video 720p(750p), 16:9 aspect ratio, without pan vectors
0x05	0xC4	H264/AVC video 720p(750p), > 16:9 aspect ratio
0x05	0xC5 - 0xD0	Reserved for future use
0x05	0xD1	H264/AVC video 240p, 4:3 aspect ratio
0x05	0xD2	H264/AVC video 240p, 16:9 aspect ratio, with pan vectors
0x05	0xD3	H264/AVC video 240p, 16:9 aspect ratio, without pan vectors
0x05	0xD4	H264/AVC video 240p, 4:3 aspect ratio > 16:9 aspect ratio
0x05	0xD5 - 0xE0	Reserved for future use
0x05	0xE1	H264/AVC video 1080p(1125p), 4:3 aspect ratio
0x05	0xE2	H264/AVC video 1080p(1125p), 16:9 aspect ratio, with pan vectors
0x05	0xE3	H264/AVC video 1080p(1125p), 16:9 aspect ratio, without pan vectors
0x05	0xE4	H264/AVC video 1080p(1125p), > 16:9 aspect ratio
0x05	0xD5 – 0xE0	Reserved for future use
0x06	0x00	Reserved for future use
0x06	0x01	HE-AAC MPEG4 audio, 1/0 mode (*single* mono)
0x06	0x02	HE-AAC MPEG4 audio, 1/0 + 1/0 mode (dual mono)
0x06	0x03	HE-AAC MPEG4 audio, 2/0 mode (stereo)
0x06	0x04	HE-AAC MPEG4 audio, 2/1 mode
0x06	0x05	HE-AAC MPEG4 audio, 3/0 mode
0x06	0x06	HE-AAC MPEG4 audio, 2/2 mode
0x06	0x07	HE-AAC MPEG4 audio, 3/1 mode
0x06	0x08	HE-AAC MPEG4 audio, 3/2 mode
0x06	0x09	HE-AAC MPEG4 audio, 3/2 + LFE mode
0x06	0x0A – 0x3F	Reserved for future use
0x06	0x40	HE-AAC MPEG4 pure audio description for the visually impaired
0x06	0x41	HE-AAC MPEG4 audio for the hard of hearing
0x06	0x42	HE-AAC MPEG4 mixed audio description for the visually impaired
0x06	0x43	HE-AAC v2 MPEG4 audio, modo 1/0 (mono)
0x06	0x44	HE-AAC v2 MPEG4 audio, modo 2/0 (stereo)
0x06	0x45	HE-AAC v2 MPEG4 pure audio description for the visually impaired
0x06	0x46	HE-AAC MPEG4 v2 audio for the hard of hearing
0x06	0x47	HE-AAC v2 MPEG4 mixed audio description for the visually impaired
0x06	0x48 – 0x50	Reserved for futere use
0x06	0x51	AAC MPEG4 audio, 1/0 mode (*single* mono)

Table 28 (continuation)

Stream_content	Component_type	Description
0x06	0x52	AAC MPEG4 audio, 1/0 + 1/0 mode (dual mono)
0x06	0x53	AAC MPEG4 audio, 2/0 mode (stereo)
0x06	0x54	AAC MPEG4 audio, 2/1 mode
0x06	0x55	AAC MPEG4 audio, 3/0 mode
0x06	0x56	AAC MPEG4 audio, 2/2 mode
0x06	0x57	AAC MPEG4 audio, 3/1 mode
0x06	0x58	AAC MPEG4 audio, 3/2 mode
0x06	0x59	AAC MPEG4 audio, 3/2 + LFE mode
0x06	0x60 – 0x9E	Reserved for future use
0x06	0x9F	AAC MPEG4 pure audio description for the visually impaired
0x06	0xA0	AAC MPEG4 audio for the hard of hearing
0x06	0xA1	AAC M PEG4 mixed audio description for the visually impaired
0x06	0xA2-0xA9	Reserved for future use
0x06	0xAA-0xFE	User defined
0x06	0xFF	Reserved for future use
0x07 – 0x0F	0x00 – 0xFF	Reserved for future use

8.3.5 Content descriptor

The function of content descriptor is to provide classification information for an event. See EN 300 468:2007, subclause 6.2.9 for structure details.

The semantics for content descriptor shall be:

- content_nibble_level_1: 4 bits field that shall represent the first level of a content identifier. Coding of this field shall be specified according with Annex C;

- content_nibble_level_2: 4 bits field that shall represent the second level of a content identifier. Coding of this field shall be specified according with Annex C;

- user_nibble: 4 bits field that shall be defined by broadcaster.

8.3.6 Country availability descriptor

In order to identify various combinations of countries efficiently, the descriptor may appear twice for each service, once giving a list of countries and/or groups of countries where the service is intended to be available, and the second giving a list of countries and/or groups where it is not available.

The latter list overrides the former list. If only one descriptor is used, which lists countries where the service is intended to be available, it indicates that the service is not intended to be available in every other country.

If only one descriptor is used, which lists countries where the service is not intended to be available, it indicates that the service is intended to be available in every other country. If no descriptor is used (the one which lists countries where the service is available or the one which lists countries where the service is not available), then it is not defined for which countries the service is intended to be available (see EN 300 468:2007, subclause 6.2.10.)

The semantics for the country availability descriptor shall be:

- country_code: 24 bits field that identify the country using a 3 charecter code as specified in ISO 3166-1. Each character will be coded in 8 bits according to ISO/IEC 8859-15 and inserted in order with the 24 bits field.

EXAMPLE Brazilian country has 3 character code "BRA", which is coded as: “0100 0010 0101 0010 0100 0001”

8.3.7 Extended event descriptor

The extended event descriptor shall be implemented according to EN 300 468:2007, subclause 6.2.15.

The semantics for extended event descriptor shall be:

- descriptor_number: 4 bits field that shall indicate the number of the descriptor. It shall be used to associate the information which does not fit into a single descriptor. The description_number of the first extended_event_descriptor of an association of extended_event_descriptors shall be "0x0". The descriptor_number shall be incremented by 1 with each additional extended_event_descriptor in this section (see EN 300 468:2007, subclause 6.2 15);

- ISO_639_language_code: 24 bits field that shall identify the language of the component (in case of audio or data) and a text description which may be contained in the descriptor. An ISO_639_language_code contains a 3 character code as specified by ISO 639-2. Each character shall be coded in 8 bits according to ISO/IEC 8859-15 and inserted in the order with the 24 bits field;

EXAMPLE Portuguese, the Brazilian official language, has 3-character code "por", which is coded as:“ 0111 0000 0110 1111 0111 0010”

- text_char: 8 bit field. A string of text_char fields specify the non itemized extended text sent by short_extended_descriptor. Text information is coded according ISO/IEC 8859-15.

8.3.8 Linkage descriptor

The linkage descriptor shall identify the service that may be presented if the consumer requests additional information related to a specific entity described by the SI system. The location of linkage descriptor in the syntax indicates the entity for which additional information is available.

EXEMPLO A linkage descriptor within the NIT shall point to a service that provides additional information about a network; a linkage descriptor in the BAT shall provide a link to a service informing about the bouquet, etc.

A conditional access replacement service can be identified using the linkage descriptor. This service may be selected automatically by the receiver if the conditional access denies access to the specific entity (a service as an example) described by the SI system (see Table 29).

Table 29 — Linkage descriptor

Syntax	Number of bits	Identifier
linkage_descriptior(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	bslbf
linkage_type	8	uimsbf
for(i=0;i<N;i++){		
private_data_byte	8	bslbf
}		
}		

The semantics for linkage descriptor shall be:

— transport_stream_id: 16-bit field that shall identify the TS that contains the indicated service information;

— original_network_id: 16-bit field shall inform the identification tag of the network_id from the originating delivery service;

— service_id: 16-bit field shall indicate the service information within the TS. The service_id is the same as the program_number in the corresponding program_map section. If the linkage_type field have the value 0x04, then the service_id is not relevant and shall be be fixed with the value 0x0000;

— linkage_type: 8 bits field that shall specify the type of linkage, for example, the information (see Table 30);

— private_data_byte: 8 bits field that shall be privately defined.

Table 30 — Coding type of linkages

Linkage_type	Description
0x00	Reserved for future use
0x01	Information service
0x02	EPG service
0x03	CA replacement service
0x04	TS containing complete network/bouquet SI
0x05	Service replacement service
0x06	Data broadcast service
0x07 - 0x7F	Reserved for future use
0x80 - 0xBF	User defined
0xC0 - 0xFD	Reserved for future use (standardization organization defined area)
0xFE	Reserved for re-transmission
0xFF	Reserved for future use

8.3.9 Mosaic descriptor

A mosaic component shall be a collection of different video images to form a coded video component.

The information shall be organized so that each specific information, when displayed, appears as a small area of the screen.

The mosaic descriptor shall give a partitioning of a digital video component into elementary cells, the allocation of elementary cells to logical cells, and give a link between the content of the logical cell and the corresponding information (for example, bouquet, service, event etc.), according to EN 300 468:2007 subclause 6.2.20.

The semantics for mosaic descriptor shall be:

- number_of_horizontal_elementary_cells: 3 bits field that shall indicate the number of cells of horizontal screen display, according to EN 300 468:2007 subclause 6.2.20;

- number_of_vertical_elementary_cells: 3 bits field that shall indicate the number of cells of vertical screen display, according to EN 300 468:2007, subclause 6.2.20;

- logical_cell_id: 6 bits field that shall be coded in a binary form.

Different adjacent elementary cells (see EN 300 468:2007 subclause 6.2.20) may be grouped together in a logic cell.

A logical_cell_number shall be associated to a group of adjacent elementary_cell_ids. The total number of logic cells shall not exceed the number of elementary cells (maximum = 64). Each elementary cell shall be allocated to one logic cell. More than one elementary cell may belong to one logical cell as following:

- logical_cell_presentation_info: 3 bits field that shall identify the type of presentation for a logical cell;

The logical_cell_presentation information shall allow an identification of presentation styles which are defined in EN 300 468:2007, subclause 6.2.20, as following:

- cell_linkage_info: 8 bits field that shall be identifies the information type loaded by a logical cell, see ETSI EN 300 468:2007, subclause 6.2.20.

For additional information about the fields, see EN 300 468:2007, subclause 6.2.20.

8.3.10 Near video on demand (NVOD) reference descriptor

The NVOD reference descriptor, in conjunction with time shifted service descriptor and time shifted event descriptors, shall provide a mechanism for an efficient description of the number of services which carry the same sequence of events, but with starting times offset from one another.

One of the groups with time-shifted services is referred to as NVOD, since a user can at any time access near to the start of an event by selecting the appropriate service of the group.

The NVOD reference descriptor (see EN 300 468:2007, subclause 6.2.25) shall present a list of the services forming part of a NVOD service. Each service shall be also described in the appropriate SDT sub-table by a time shifted service descriptor (see 8.3.18).

The time shifted service descriptor shall associate a time shifted service with a reference_service_id. The reference_service_id shall be the label under which a full description of the NVOD service is given, but the reference_service_id does not itself correspond to any program_number in the program_map_section.

The time shifted event descriptor shall be used in the event information for each time shifted service. Instead of duplicating the full information for each event, the time shifted event descriptor shall point to a reference_event_id in the reference service, The event complete information shall be pointed by the event information on the reference service.

The services which make up an NVOD service shall not be carried in the same TS, however, a reference service shall be described in the SI on each TS which carries any NVOD services..

The semantics for NVOD reference descriptor shall be according to EN 300 468:2007, subclause 6.2.25.

8.3.11 Parental rating descriptor

The parental rating descriptor shall provide a rating based on the local parental rating regulation. In case of Brazil, the reference is the ordinance number *1220"*,of Brazilian Justice Ministry, of July, 11th , 2007 (see Table 31). In case this descriptor is transmitted in multiple tables, the information priority order shall be the following: PMT -> EIT.

Table 31 — Parental rating descriptor

Syntax	Number of bits	Identifier
parental_rating_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
Country_code	24	bslbf
rating	8	
}		
}		

The semantics for parental rating descriptor shall be:

- country_code: 24 bits field that shall identify the country using a 3-character code as specified in ISO 3166-1. Each character will be coded in 8 bits according to ISO 8859-15 and inserted in order into the 24 bits field.

EXEMPLE Brazil has 3 characters code "BRA", which is coded as: "0100 0010 0101 0010 0100 0001".

- rating*:* 8 bits field that shall indicate the age and the content description. The bits are coded as showed in Figure 2.

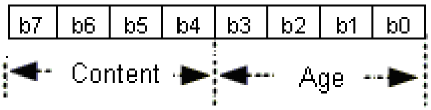


Figure 2 — Bits distribution for field rating

The 4 LSB (Low significant bits) shall indicate the age as show in Table 32.

Table 32 — Age description

Binary code	Age
0000	Reserved
0001	L
0010	10
0011	12
0100	14
0101	16
0110	18
0111 - 1111	Reserved

The 4 MSB (major significant bit) shall indicate the content as show in Table 33.

Table 33 — Content objective description

Binary code	Content
0001	Drugs
0010	Violence
0100	Sex
0011	Violence and drugs
0101	Sex and drugs
0110	Violence and sex
0111	Violence, sex and drugs
NOTE The MSB is reserved for future use	

8.3.12 Network name descriptor

The network name descriptor shall provide the network name. This descriptor shall be in the first NIT loop. (see Table 34)

Table 34 — Network name descriptor

Syntax	Number of bits	Identifier
network_name_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
}		

The semantics for network name descriptor:

- char: 8 bits field which the string of characters shall specify the name of the delivery system indicated by NIT. Text shall be coded using the method described in service provider operational standard.

8.3.13 Service descriptor

The service descriptor shall provide the name of service provider and the services in text form together with service_type (see Table 35).

Table 35 — Service descriptor

Syntax	Number of bits	Identifier
service_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
service_type	8	uimsbf
service_provider_name_length	8	uimsbf
for(i=0;i<N;i++){		
char	8	uimsbf
}		
service_name_length	8	uimsbf
for(i=0;i<N;i++){		
Char	8	uimsbf
}		
}		

The semantics for service descriptor shall be:

- service_type: 8 bits field that shall specify the service type. It shall be coded according to Table 36;

- char: 8-bit field which the sequence of characters shall inform the name of service provider or service. The details of characters shall be specified in the operation standard of service providers;

- service_name_length: 8 bits field that shall specify the number of bytes preceding it to describe the characters of service name;

- service_provider_name_length: 8 bits field that shall specify the number of bytes preceding the field service_provider_name_length to describe the characters of the name of service provider.

Table 36 — Service type coding

Service_type	Description
0x00	Reserved for future use
0x01	Digital television service
0x02	Digital audio service
0x03	Teletext service
0x04	NVOD reference service
0x05	NVOD time-shifted service
0x06	Mosaic service
0x07 – 0x09	Reserved for future use
0x0A	Advanced coding for digital radio service
0x0B	Advanced coding for mosaic service
0x0C	Data broadcasting service
0x0D	Reserved for common interface (see EN 50221)
0x0E	RCS Map (see EN 301 790)
0x0F	RCS FLS (see EN 301 790)
0x10	Serviço DVB MHP
0x11	Digital MPEG2 HD television service
0x12 – 0x15	Reserved for future use
0x16	Advanced coding for digital SD television service
0x17	Advanced coding for digital NVOD SD time-shifted television service
0x18	Advanced coding for digital NVOD SD referece television service
0x19	Advanced coding for digital NVOD HD television service
0x1A	Advanced coding for digital NVOD HD time-shifted television service
0x1B	Advanced coding for digital NVOD HD reference television service
0x1C – 0x7F	Reserved for future use
0x80 – 0xA0	Defined by the service provider
0xA1	Special video service
0xA2	Special audio service
0xA3	Special data service
0xA4	Engineering service (software download)
0xA5	Promotional video service
0xA6	Promotional audio service
0xA7	Promotional data service
0xA8	Data service for anticipated storage
0xA9	Exclusive data service for storage
0xAA	Bookmark service list
0xAB	Simultaneous server type service
0xAC	Independent file service
0xAD – 0xBF	Not defined (range defined by the standardization organization)
0xC0	Data service
0xC1 – 0xFF	Not defined

8.3.14 Service list descriptor

The service list descriptor shall provide a service list description by service_id and service_type (see Table 36). It shall be used to list the services and their respective types for each TS. The original_network_id and TS_id are required to identify a service and shall br provided in the beginning of second NIT loop. The service_id shall be identify the network and the type of service shall be defined in accordance with Table 37.

The service_id shall identify the network and the type of service according to Table 37.

Table 37 — Service list descriptor

Syntax	Number of bits	Identifier
service_list_descriptor(){		
descriptor_tag	8	Uimsbf
descriptor_length	8	Uimsbf
for(i=0;i<N;i++){		
service_id	8	Uimsbf
service_type		
}		
}		

The semantics for the service list descriptor shall be:

— service_id: 16 bits field that shall identify a service within a TS. The service_id shall be the same as the program_number in the corresponding program_map_section;

— service type: 8 bits field which shall that specifies shall be the service type. This field shall be coded according to Table 36.

8.3.15 Short event descriptor

The short event descriptor shall provide the name of event and a short description of the event in text form (see EN 300 468:2007, subclause 6.2.36).

The semantics for short event descriptor shall be:

— ISO 639_language_code: 24 fields that shall contain the ISO 639-2 three language code of the language of the following text fields. Each character shalll be coded in 8 bits according to ISO 8859-15 and inserted in order into the 24 bits field;

EXAMPLE Portuguese,the Brazilian official language, has 3-character code "por", which is coded as: “ 0111 0000 0110 1111 0111 0010”

— text_char: 8 bit field. A string of char fields specify the text description for the event. Text information is coded according ISO/IEC 8859-15.

For additional fields, see EN 300 468:2007, subclause 6.2.36.

8.3.16 Stream identifier descriptor

The stream identifier descriptor (see EN 300 468:2007, subclause 6.2.38) may be used in PMT to identify the stream components of a service so they can be differentiated. This differentiation can be made, for example, using a description provided by the component descriptor on EIT, in case this table is present (e.g. a stream component of a service is "video, 16:9 aspect ratio, with pan vector").The stream identifier descriptor shall be located before its respective ES_info_length_field.

The semantics for the stream identifier descriptor shall be according to EN 300 468:2007, subclause 6.2.38.

8.3.17 Stuffing descriptor

The stuffing descriptor shall provide a description of previously invalidated coded descriptors or insertion of dummy descriptors for table stuffing.

This descriptor may appear in any place in SI where a descriptor is allowed. It shall be used to stuff tables, for any reason, or to disable descriptors which are not valid anymore (for example, in the case of a remultiplexing). Access terminal shall skip the stuffing descriptor (see Table 38).

Table 38 — Stuffing descriptor

Syntax	Number of bits	Identifier
stuffing_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
stuffing_byte	8	bslbf
}		
}		

The semantics for stuffing descriptor shall be:

- stuffing_byte: each occurrence of this 8 bits field may be set to any value. Access terminals may discard the stuffing bytes.

8.3.18 Time shifted event descriptor

The time shifted event descriptor shall be used in the place of a short_event_descriptor to indicate an event which is copied and time shifted in relation to other event.

The semantics for time shifted event descriptor shall be according to EN 300 468:2007, subclause 6.2.43.

8.3.19 Time shifted service descriptor

The time shifted service descriptor shall be used in place of service descriptor to indicate services which are time shifted copies of other services.

The semantics for time shifted service descriptor shall be according to EN 300 468:2007, subclause 6.2.44.

8.3.20 Data component descriptor

The data component descriptor (see Table 39) shall be used to identify the type of coding of data.

Table 39 — Data component descriptor

Syntax	Number of bits	Identifier
data_component_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_component_id	16	uimsbf
for(i=0;i<N;i++){		
additional_data_component_info	8	uimsbf
}		
}		

The semantics for data component descriptor shall be:

- data_component_id: with 16 bits, this field shall be used to identify the data coding method. This field values may be defined in the future.

- additional_data_component_info: with 8 bits, this field shall be used to increase the identification number or storage of supplement information specified in each coding method. The syntax of information described in this area shall be specified otherwise for each data coding method.

8.3.21 System management descriptor

The system management descriptor (see Table 40 and Table 41) shall be used to identify broadcasting and non-broadcasting.

Table 40 — System management descriptor

Syntax	Number of bits	Identifier
system_management_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
system_management_id	16	uimsbf
for(i=0;i <N;i++){		
additional_identification_info	8	uimsbf
}		
}		

The semantics for system management descriptor shall be:

- system_management_id: 16 bits field that shall be composed as shown in the Table 42;

- broadcasting_flag: 2 bits field which shall indicate type of broadcasting/non-broadcasting in accordance with Table 43. The broadcasting shall be the transmission available for all receivers, while non-broadcasting shall have some type of conditional access;

- broadcasting_identifier: 6 bits field which shall indicate the standard broadcasting method in accordance with Table 44. In case of Brazilian sistem, this field always shall be 000011;

— additional_broadcasting_identification: 8 bits field which shall be specified by operation standard of service providers.

EXAMPLE 000001 ISDB System; other values: reserved.

— additional_identification_info: 8 bits field, as defined on Table 41, which shall be used to increase system management identification number.

Table 41 — Construction of system management identifier

Syntax	Number of bits	Identifier
system_management_id(){		
broadcasting_flag	2	Uimsbf
broadcasting_identifier	6	Uimsbf
additional_broadcasting_identification	8	Uimsbf
}		

Table 42 — Transmission type

Value	Semantics
00	Broadcasting
01, 10	Non Broadcasting
11	Undefined

Table 43 — Types of transmission system standards

Value	Semantics
‘000000	Undefined
‘000001	Not used
‘000010	Not used
‘000011	ISDB system
000100 – 000110	Not used
000111 – 111111	Undefined

8.3.22 Hierarchical transmission descriptor

The hierarchical transmission descriptor (see Table 44) shall be used to indicate the relation between the hierarchical streams when transmitting events hierarchically.

Table 44 — Hierarchical transmission descriptor

Syntax	Number of bits	Identifier
hierarchical_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved_future_use	7	bslbf
quality_level	1	bslbf
reserved_future_use	3	bslbf
reference_PID	13	uimsbf
}		

The semantics for hierarchical transmission descriptor shall be:

— quality_level: 1 bit field that shall indicate the hierarchical level. The hierarchical construction takes place in two levels, HQ and LQ. When the hierarchical level is 1, the stream shall be in high quality. When the level is 0, the stream shall be in low quality;

— reference_PID: 3 bits field that shall indicate the PID of elementary stream to be referred, for all the streams having hierarchical construction.

8.3.23 Digital copy control descriptor

The digital copy control descriptor (see Table 45) shall provide information to control copies generated in digital recording equipment. This information shall be provided by the broadcasting service provider (holder of copyrights)

This descriptor shall also be used to identify the maximum transmission rate for each event.

In case it is sent by several tables, the priority of information shall be as follow: PMT > EIT > SDT.

For the bit definition for the digital copy control descriptor is given on Annex D.

Table 45 — Digital copy control descriptor

Syntax	Number of bits	Identifier
digital_copy_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
component_control_flag	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type != 00){		
APS_control_data	2	bslbf
}		
Else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
if(component_control_flag == 1){		
component_control_length	8	uimsbf
For(j=0;j<N;j++){		
component_tag	8	uimsbf
digital_recording_control_data	2	bslbf
Maximum_bitrate_flag	1	bslbf
reserved_future_use	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type != 00){		
APS_control_data	2	bslbf
}		
else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
}		
}		

The semantics for digital copy control descriptor shall be:

- digital_recording_control_data: 2 bits field that shall indicate information to control copy generation. It shall be coded according to Table 46;

- maximum_bit_rate_flag: 1 bit field that, when it is set to 1, it means that following maximum transmitting rate field shall effective. When it is set to 0, it shall mean that following maximum transmitting rate field does not exist;

- component_control_flag: 1 bit field that indicates if shall be necessary to specify digital copy control information in each event component. When it is set to 1, the field after component control length shall be effective and the digital copy information shall be specified in each event component. When it is set to 0, the digital copy control information shall be specified for all event components and the field after component control length does not exist. When this descriptor is transmitted by PMT, the component_control_flag shall always be 0;

- maximum_bit_rate: 8 bits field that shall describe the transmission rate of transport stream packet for each event or elementary stream, increasing at each 1/4 Mbps. In case of variable transmission rates, the maximum value shall be described;

- component_control_length: 8 bits field that shall indicate the size in bytes of the next loop of component control;

- component_tag: 8 bits field that shall identify the elementary stream of component whose events shall have the same value of component tag of the stream identifier descriptor and the component descriptor.

Table 46 — Copy generation control descriptor

Digital copy control information	Description
00	Copy can be done without control
01	Defined by the service provider [a]
10	Copy may be done once [b]
11	Copy is forbidden

[a] The broadcasting service provider can define originally.

[b] The signal received by broadcasting can be recorded only once.

8.3.24 Emergency information descriptor

The emergency information descriptor shall be used in case of emergency alarm broadcasting performed by the broadcasters.

Table 47 — Emergency information descriptor

Syntax	Number of bits	Identifier
emergency_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
service_id	16	uimsbf
start_end_flag	1	bslbf
signal_level	1	bslbf
reserved_future_use	6	bslbf
area_code_length	8	uimsbf
for(j=0;j<N;j++){		
area_code	12	bslbf
reserved_future_use	4	bslbf
}		
}		
}		

The semantics for emergency information descriptor shall be:

— service_id: 16 bits field that shall indicate the broadcasting event number. It shall be the same as program_number;

— start_end_flag: 1 bit field that shall correspond to start signal and end signal of emergency alarm specified by responsible agencies. When this bit is set to "1", shall mean that emergency signal was started or it is being broadcasted. When it is set to "0", shall mean that emergency signal was ended;

— signal_level: 1 bit field that shall correspond to emergency alarm signal specified by responsible agencies. When set to "0", shall mean that emergency alarm signal is the first type of start signal. When set to "1", the alarm signal shall be the second type of start signal;

— area_code_length: 8 bits field which shall indicate the size in bytes of area code;

— area_code: 12 bits field which shall correspond to area code specified by responsible agencies. The allocation of area code shall be specified according to Annex E.

NOTE The type and local code will be defined by responsible regulation agencies.

8.3.25 Local time offset descriptor

The local time offset descriptor (see EN 300 468:2007, subsection 6.2.19) shall be used to adjust the UTC-3 time and the local time.

The semantics for local time offset descriptor shall be:

- country_code: 24 bits field that shall identify the country using 3 characters as specified in ISO 3166-1. Each character shall be coded in 8 bits according to ISO 8859-15 and inserted in order into the 24 bits field;

EXAMPLE Brazil has 3-character code "BRA", which is coded as: “0100 0010 0101 0010 0100 000”.

- country_region_id: 6 bits field which shall identify the regions (zones) of the country;

- local_time_offset_polarity: 1 bit field that shall indicate the polarity of the value of local_time_offset and next_time_offset. When set to "0", the local time shall advance from UTC_time. When set to "1", the local time shall be behind of UTC_time;

- local_time_offset: 16 bits field that shall have the current offset time from UTC-3 in a range between - 12 hours to + 12 hours in the area indicated by the combination of country_code and country_region_id. These 16 bits shall be coded as 4 digits in 4 bits BCD in the order of hour tens, hour, minute tens, and minutes;

- time_of_change: 40 bits field that shall specify the date and time in MJD and UTC-3 (see Annex A). This field shall be coded using 16 bits for MJD through 16 LSB and 24 bits coded as 6 digits in the 4 bits BCD;

- next_time_offset: 16 bits field that shall have the next time change after the change of UTC-3 value in the range between - 12 hours to + 12 hours in the area indicated by the combination of country_code and country_region_id. These 16 bits shall be coded as 4 digits in 4 bits BCD in the order of hour tens, hour, minute tens, and minutes.

8.3.26 Audio component descriptor

Audio component descriptor shall be used to indicate each parameter of audio elementary packet and to symbolically express the elementary stream (see Table 48).

Table 48 — Audio component descriptor

Syntax	Number of bits	Identifier
audio_component_descriptor(){		
descriptor_tag	8	Uimsbf
descriptor_length	8	Uimsbf
reserved_future_use	4	Bslbf
stream_content	4	Uimsbf
component_type	8	Uimsbf
component_tag	8	Uimsbf
stream_type	8	Uimsbf
simulcast_group_tag	8	Bslbf
ES_multi_lingual_flag	1	Bslbf
main_component_flag	1	Bslbf
quality_indicator	2	Bslbf
sampling_rate	3	Uimsbf
reserved_future_use	1	Bslbf
ISO_639_language_code	24	Bslbf
if(ES_multi_lingual_flag == 1){		
ISO_639_language_code_2	24	Bslbf
}		
for(i=0;i<N;i++){		
text_char	8	Uimsbf
}		
}		

The semantics for the audio component descriptor shall be:

- stream_content: 4 bits field that shall indicate the stream type. For audio stream the value shall be set to "0x06" (see Table 28);

- component_type: 8 bits field that shall specify the type of audio component and the coding according to Table 49;

- component_tag: 8 bits field that shall be a label to identify the component stream and shall have the same value as component_tag field in the stream identifier descriptor, whether it is present in the PSI program map section for component stream (see 8.3.16);

- stream_type: 8 bits field that shall indicate the audio stream type (see ABNT NBR 15603-1:2007, Table 6);

- simulcast_group_tag: 8 bits field that shall provide the same number to the operating simulcast component (transmit the same content by different coding methods). For component which does not operate simulcast, it shall be set to "0xFF";

- ES_multi_lingual_flag: 1 bit flag that shall be set to "1" when 2 languages (ES multilingual mode) are made in ES at 1/0 + 1/0 mode. In case of other mode, this field shall be reserved;

- main_component_flag: 1 bit flag shall be set to "1" when audio component is the main audio. In case of operation in 1/0 + 1/0, flag shall be set to "1" when the first audio component is the main audio;

- quality_indicator: 2 bits field that shall indicate the audio quality mode and its coding according to Table 50;

- sampling_rate: 3 bits field that shall show the sampling frequency and its coding is made according to Table 51;

- ISO_639_language_code: 24 bits field that shall identify audio component language. If ES is working in multilingual mode, this field shall indicate the first audio component language. This field has a 3-character code as specified by ISO 639-2. Each character shall be coded in 8 bits according to ISO 8859-1 and inserted in order into the 24 bits field;

EXAMPLE Portuguese, official language of Brazil has 3-character code "por", which is coded as: “0111 0000 0110 1111 0111 0010”

- ISO_639_language_code_2: 24 bits field that shall identifiy the second language of audio component language when ES is working in multilingual mode;

- text_char: 8 bits field, where a word shall describe the text in the component stream. Text information shall be coded using the fixed characters according to the methods described the operational standard of service providers.

Table 49 — Audio component descriptor

component_type	Descriptor
0x00	Reserved for future
0x01	HE-AAC MPEG4 audio, 1/0 mode (single mono)
0x02	HE-AAC MPEG4 audio, 1/0 + 1/0 mode (dual mono)
0x03	HE-AAC MPEG4 audio, 2/0 mode (stereo)
0x04	HE-AAC MPEG4 audio, 2/1 mode
0x05	HE-AAC MPEG4 audio, 3/0 mode
0x06	HE-AAC MPEG4 audio, 2/2 mode
0x07	HE-AAC MPEG4 audio, 3/1 mode
0x08	HE-AAC MPEG4 audio, 3/2 mode
0x09	HE-AAC MPEG4 audio, 3/2 + LFE mode
0x0A – 0x3F	Reserved for future use
0x40	HE- AAC MPEG4 pure audio description for the visually impaired
0x41	HE-AAC MPEG4 audio for the hard of hearing
0x42	HE- AAC MPEG4 mixed audio description for the visually impaired
0x43	HE-AAC v2 MPEG4 audio, 1/0 mode (single mono)
0x44	HE-AAC v2 MPEG4 audio, modo 2/0 (stereo)
0x45	HE- AAC v2 MPEG4 pure audio description for the visually impaired
0x46	HE-AAC MPEG4 v2 audio for the hard of hearing
0x47	HE- AAC v2 MPEG4 mixed audio description for the visually impaired
0x48 – 0x50	Reserved for futere use
0x51	AAC MPEG4 audio, 1/0 mode (single mono)
0x52	AAC MPEG4 audio, 1/0 + 1/0 mode (dual mono)
0x53	AAC MPEG4 audio, 2/0 mode (stereo)
0x54	AAC MPEG4 audio, 2/1 mode
0x55	AAC MPEG4 audio, 3/0 mode
0x56	AAC MPEG4 audio, 2/2 mode
0x57	AAC MPEG4 audio, 3/1 mode
0x58	AAC MPEG4 audio, 3/2 mode
0x59	AAC MPEG4 audio, 3/2 + LFE mode
0x60 – 0x9E	Reserved for future use
0x9F	AAC MPEG4 pure audio description for the visually impaired
0xA0	AAC MPEG4 audio for the hard of hearing
0xA1	AAC M PEG4 mixed audio description for the visually impaired
0xA2-0xA9	Reserved for future use
0xAF-0xFE	User defined
0xFF	Reserved for future use

Table 50 — Quality indicator

Quality indicator	Description
00	Reserved for future use
01	Mode 1 [a]
10	Mode 2 [a]
11	Mode 3 [a]
[a] For further information, see ABNT NBR 15602-2.	

Table 51 — Sampling frequency

linkage type	Description (kHz)
000	Reserved for future use
001	16 kHz
010	22.05 kHz
011	24 kHz
101	32 kHz
110	44.1kHz
111	48 kHz

8.3.27 Target region descriptor

The target region descriptor (see Table 52) shall be used to describe the region designated to an event or a part of the stream composing an event.

Table 52 — Target region descriptor

Syntax	Number of bits	Identifier
target_region_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
region_spec_type	8	uimsbf
target_region_spec()		
}		

The semantics for target region descriptor shall be:

- region_spec_type: 8 bits field that shall designate the description method for construction and coding of next target_region_spec () according to Table 53;

- target_region_spec (): field that shall indicate the syntax for the region specified in the method.

Table 53 — Target region descriptor

region_spec_type field value	Semantics
0x00 – 0xFF	Reserved

8.3.28 Data content descriptor

The data content descriptor shall be used to describe in detail the individual information about each data contents in the broadcasting event according to Table 54.

Table 54 — Data content descriptor

Syntax	Number of bits	Identifier
data_content_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_component_id	16	uimsbf
entry_component	8	uimsbf
selector_length	8	uimsbf
for(i=0;i<N;i++){		
selector_byte	8	uimsbf
}		
num_of_component_ref	8	uimsbf
for(i=0;i<num_of_component_ref;i++){		
component_ref	8	uimsbf
}		
ISO_639_language_code	24	bslbf
text_length	8	uimsbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

The semantics for data content descriptor shall be:

- data_component_id: 16 bits field that shall describe the same value as data_component_identifer of data_component_descriptor;
- entry_component: 8 bits field that shall designate the component stream (including data which shall be referred first) among multiple component streams composing data broadcasting contents, using component tag;
- sector_length: 8 bits field that shall specify the length of next selector area;
- sector_byte: 8 bits field that shall make necessary information available to get data from a sequence of selector areas. The syntax described for this area shall be independently specified for each data component;

- num_of_component_ref: 8 bits field that shall indicate the total number of component streams in the event. It shall be necessary to playback and record the content indicated by this descriptor (but the component streams designated by entry component field shall be excluded). This number corresponds to byte length of the next component reference loop;

- component_ref: 8 bits field that shall describe the component_tag of content in the event. It shall be necessary to watch or record contents (but the component stream designates by entry component shall be excluded);

- ISO_639_language_code: 24 bits field that shall identify the descriptor language of character used in the following service descriptor containing a 3 characters code specified by ISO 639-2;

- text_length: 8 bits field that shall indicate the byte length of the next content descriptor;

- text_char: 8 bits field that shall describe the content being broadcasted.

8.3.29 Hyperlink descriptor

The hyperlink descriptor shall be used to describe the linkage of other event, event contents and event-related information according to Table 55.

Table 55 — Hyperlink descriptor

Syntax	Number of bits	Identifier
hyperlink_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
hyper_linkage_type	8	uimsbf
link_destination_type	8	uimsbf
selector_length	8	uimsbf
for(i=0; i<selector_length; i++){		
selector_byte	8	uimsbf
}		
for(i=0; i<N; i++){		
private_data	8	uimsbf
}		
}		

The semantics for hyperlink descriptor shall be:

- hyper_linkage_type: 8 bits field that shall indicate the linkage type and shall be coded in accordance with Table 56;

- link_destination_type: 8 bits field that shall indicate the linkage type and shall be the code value which is in accordance with Table 57 to 64;

- selector_length: 8 bits field that shall indicate the size in bytes of the next area selector;

- selector_byte: 8 bits field which shall specify a sequence of selector areas which shall describe the destination of the next links of the type specified by each destination link;

- original_network_id: 16 bits field that shall provide network_id identifiers from originating system where the linked service belongs;

- transport_stream_id: 16 bits field that shall provide identifiers of transport stream where the linked service belongs;

- service_id: 16 bits field that shall provide identifiers of services in the linked transport streams and shall describe the same service_id as a program_number in the corresponding selection of program maps;

- original_network_id: 16 bits field that shall provide network_id identifiers from originating system which the linked service belongs;

- transport_stream_id: 16 bits field that shall provide transport stream identifiers where the linked service belongs;

- service_id: 16 bits field that shall provide identifiers of services in the linked transport stream and shall describe the same service_id as a program_number in the corresponding selection of program maps;

- event_id: 16 bits field that shall describe the same identification number of the linked event;

- original_network_id: 16 bits field that shall provide network_id identifiers from originating system where the linked carousel module belongs;

- transport_stream_id: 16 bits field that shall provide transport stream identifiers where the linked carousel module belongs;

- service_id: 16 bits field that shall provide identifiers of services in the linked transport stream where the carousel module belongs and shall describe the same service_id as a program_number in the corresponding section of program maps;

- event_id: 16 bits field that shall describe the identification number of the event where the linked carousel module belongs;

- component_tag: 8 bits field that shall describe the transmission transport stream identifier linked to the carousel module;

- moduleId: 16 bits field that shall describe the identification number of the linked carousel module;

- original_network_id: 16 bits field that shall provide network_id identifiers from originating system where the linked content module belongs;

- transport_stream_id: 16 bits field that shall provide packet stream identifiers where the linked content module belongs;

- service_id: 16 bits field that shall provide identifiers of services in the linked transport stream where the content module belongs and shall describe the same service_id as a program_number in the corresponding selection of program maps;

- content_id: 32 bits field that shall describe the identification number of the content uniformly linked to the service;

- original_network_id: 16 bits field that shall provide network_id identifiers from originating system where the linked content module belongs;

- transport_stream_id: 16 bits field that shall provide packet stream identifiers where the linked content module belongs;

- service_id: 16 bits field that shall provide identifiers of services in the linked transport stream where the content module belongs and shall describe the same service_id as a program_number in the corresponding selection of program maps;

- content_id: 32 bits field that shall describe the identification number of the content module uniformly linked to the service;

- component_tag: 8 bits field that shall describe the identifier of component stream transmitting the linked carousel module;

- moduleID: 16 bits field that shall describe the identification number of the connection module;

- information_provider_id: 16 bits field that shall provide of information provided by identifier of events related to the sub-table where the linked node belongs;

- event_relation_id: 16 bits field that shall identify events related to subtables where the linked destination belongs;

- node_id: 16 bits field that shall be responsible for identifying the linked destination node;

- url_char: series of fields that shall describe the URL of contents of the accumulated data services. They shall describe the methods of URL specified in ARIB STD-B24.

Table 56 — Hyperlink descriptor

hyper_linkage_type	Semantics
reserved (0x00)	Reserved
combined_data (0x01)	Used for indicate data transmission event which transports SI related to television events, when the SI is transported in other frame or other event. When making reservation or playback of the data transmission event of the broadcaster, shall be described detailed in SI of television event
combined_stream (0x02)	Used for indicate a television event which is related to the SI transported in the transmission data event, when SI is transported in other frame or other event. When making reservation or playback of the data transmission event of the broadcaster, shall be described detailed in SI of data transmission event
content_to_index (0x03)	Used for indicate event which transport internal index information related to the television event, when the internal index information is transported in other frame or other event. For accumulate or use the related index information, when accumulating or playing television event, shall be described detailed in the internal index information of the television event
index_to_content (0x04)	Used for indicate television event which relates to the event internal information index when the internal information index is transported in other frame or other index. For accumulate or playing related television event , when accumulating or using index information, shall be described detailed in the information index event.
guide_data (0x05)	Used for indicate data transmisson event which transport the event guide information related to this event, when the event guide information is transported in other data transmission event. To get detailed information about the event guide application when demanded, it shall indicate the designated data
(0x06)	Undefined
content_to_metadata (0x07)	Used for indicate event and service which transport metadata related to the television events or data transmission events, when the metadata is transmitted in a frame or service different than the television event or data transmission event. For accumulate or use the metadata related, when accumulating or playing the televison event or data transmission event, shall be described as detailed information about the television event or data transmission event
metadata_to_content (0x08)	Used for indicate television event or data transmission event which relates to the metadata transported into the event or service when metadata is transmitted in a frame or service different than the television event or data transmission event. For accumulate or playback the television event or related data transmission event, when storing or using metadata, it shall be described as detailed information of metadata event.
portal_URI (0x09)	Used to indicate the Uri of the destination gateway link in the transmissionof the server type. The URI of the destination gateway link corresponds to the URI of the BML document provided by the broadcaster for the contract between broadcaster and the viewers.
authority_URI (0x0A)	Used for indicate the URI of the authority in server type transmission. The authority is the caracter chain used as a name space for each broadcaster when accumulating server-type content in server-type receivers
(0x0B – 0x3F)	Undefined
index_module (0x40)	Used only for the LIT as information index in the data transmission event, to indicate correspondency of local event identifier and data transmission event module. The operation relies on the application in the receiver which uses the internal index information.
(0x41 – 0x7F)	Undefined
user_private (0x80 – 0xFF)	Linkage type defined by the users

Table 57 — Hyperlink type descriptor

link_destination_type	selector_length	Linkage target
reserved (0x00)	-	-
link_to_service (0x01)	6	Service
link_to_event (0x02)	8	Event
link_to_module (0x03)	11	Event specific module
link_to_content (0x04)	10	Content
link_to_content_module (0x05)	13	Content specific module
link_to_ert_node (0x06)	6	Table node related to event
link_to_stored_content (0x07)	Variable Length	Accumulated content
reserved_future_use (0x08 – 0x7F)	-	Reserved for future use
user_private (0x80 – 0xFE)	-	Destination type linkage of user definition
reserved (0xFF)	-	-

Table 58 — Area selector description (link_destination_Type: 0x01)

Syntax (link_destination_type:0x01)	Number of bits	Identifier
link_service_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
}		

Table 59 — Area selector description (link_destination_Type: 0x02)

Syntax (*link_destination_type:0x02*)	Number of bits	Identifier
link_event_info(){		
original_network_id	16	*uimsbf*
transport_stream_id	16	*uimsbf*
service_id	16	*uimsbf*
event_id	16	*uimsbf*
}		

Table 60 — Area selector description (link_destination_Type: 0x03)

Syntax (link_destination_type:0x03)	Number of bits	Identifier
link_module_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
component_tag	16	uimsbf
moduleId	16	uimsbf
}		

Table 61 — Area selector description (link_destination_Type: 0x04)

Syntax (link_destination_type:0x04)	Number of bits	Identifier
link_content_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
content_id	32	uimsbf
}		

Table 62 — Area selector description (link_destination_Type: 0x05)

Syntax (link_destination_type:0x05)	Number of bits	Identifier
link_content_module_info(){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
content_id	32	uimsbf
component_tag	8	uimsbf
moduleId	16	uimsbf
}		

Table 63 — Area selector description (link_destination_Type: 0x06)

Syntax (link_destination_type:0x06)	Number of bits	Identifier
link_ert_node_info(){		
information_provider_id	16	uimsbf
event_relation_id	16	uimsbf
node_id	16	uimsbf
}		

Table 64 — Area selector description (link_destination_Type: 0x07)

Syntax (link_destination_type:0x07)	Number of bits	Identifier
link_stored_content_info(){		
for(i=0;i<N;i++){		
url_char	8	uimsbf
}		
}		

8.3.30 Video decode control descriptor

The video decode control descriptor shall be used to control the video decoding to receive still picture composed of MPEG pictures transmitted at a low speed. Also, this descriptor shall be used to smooth the video display at slice points, using the changing of video coding method (see Table 65).

Table 65 — Video decode control descriptor

Syntax	Number of bits	Identifier
Video_decode_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
still_picture_flag	1	bslbf
sequence_end_code_flag	1	bslbf
video_encode_format	4	bslbf
reserved_future_use	2	bslbf
}		

The semantics for video decode control descriptor shall be:

— still_picture_flag: 1 bit field that, when set to "1" it shall mean that this component is still picture (MPEG-picture). When it is "0" it shall mean that this component is a moving picture;

— sequence_end_code_flag: 1 bit field shall indicate whether or not this video component has a sequence end code defined by ABNT NBR 15602-1. When it is "1", it shall mean the existence of sequence end code and, when it is "0", it shall mean the absence of sequence end code;

— video_encode_format: 4 bits field that shall indicate the encode format, according to Table 66.

Table 66 — Video encoding format

Video encoding format	Description
0000	1080p
0001	1080i
0010	720p
0011	480p
0100	480i
0101	240p
0110	120p
0111	Reserved
1000-1111	For video enconding format extension

8.3.31 Terrestrial delivery system descriptor

The terrestrial delivery system descriptor shall indicate the physical conditions of terrestrial transmission path (see Table 67).

Table 67 — Terrestrial delivery system descriptor

Syntax	Number of bits	Identifier
terrestrial_delivery_system_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
area_code	12	bslbf
guard_interval	2	bslbf
transmission_mode	2	bslbf
for(i=0; i < N;i++){		
frequency	16	uimsbf
}		
}		

The semantics of terrestrial delivery system descriptor shall be:

- area_code: 12 bits field that shall indicate the service area code. The area codes shall be specified elsewhere;
- guard_interval: 2 bits field that shall indicate the guard interval in accordance with Table 68;
- transmission_mode: 2 bits field that shall indicate the mode information in accordance with Table 69;
- frequency: 16 bits field that shall indicate the center frequency. The frequency unit shall be in accordance with ABNT NBR 15601. In case of MFN, the multiple frequencies used shall be listed. The decimal value for this field shall be in accordance wiyh the equation:

  $(473 + 6 \times (X – 14) + 1/7) \times 7 = xxx$ MHz

  where “X” is the channel number.

Table 68 — Guard interval

Guard interval	Description
00	1/32
01	1/16
10	1/8
11	1/4

Table 69 — Mode information

Transmission mode	Description
00	Mode 1
01	Mode 2
10	Mode 3
11	Undefined

8.3.32 Partial reception descriptor

The partial reception descriptor shall describe the service_id transmitted by partial reception hierarchy of the terrestrial transmission path (see Table 70).

Table 70 — Partial reception descriptor

Syntax	Number of bits	Identifier
partial_reception_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i < N;i++){		
service_id	16	uimsbf
}		
}		

The semantics for partial reception descriptor shall be:

- service_id: 16 bits field that shall indicate the service_id of information service in a partial reception hierarchy. The service_id is identical to program_number in the corresponding section of program map.

8.3.33 Series descriptor

The series descriptor shall be used to describe series events (see Table 71).

Table 71 — Series descriptor

Syntax	Number of bits	Identifier
series_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
series_id	16	uimsbf
repeat_label	4	uimsbf
program_pattern	3	uimsbf
expire_date_valid_flag	1	uimsbf
expire_date	16	uimsbf
episode_number	12	uimsbf
last_episode_number	12	uimsbf
for(i=0; i < N;i++){		
series_name_char	8	uimsbf
}		
}		

The semantics for series descriptor shall be:

- series_id: 16 bits field that shall identify series uniquely;

- repeat_label: 4 bits field that shall give the label identifying the broadcasting series, informing its duration and its repetition. Original series broadcasting shall be given with "0x0";

- program_pattern: 3 bits field that shall indicate the program pattern in accordance with Table 72. This may show when the next event of the series shall be exhibited;

- expire_date_valid_flag: 1 bit flag that shall indicate that the following expire_date is valid. The value of the end date of the series is valid, this value shall be set to "1";

- expire_date: 16 bits field that shall indicate the limit date of series using the same format as lower 16 bits of MJD. Even when the last event could not be recognized for some reason, the access terminal recognizes that the series is ended when the date is sent;

- episode_number: 12 bits field that shall indicate the episode number described by descriptor. It can be indicated from 1 to 4 095. When the episode number exceeds this value, it shall be set separately. When an event number could not be defined due a series event, use "0x000";

- last_episode_number: 12 bits field that shall indicate the total number of episodes of the series. It can be indicated from 1 to 4 095. When the episode number exceeds this value, it shall be set separately. When the last time is not defined yet, it shall set to "0x000";

- series_name_char: shall be a string of characters indicating the series name. The character details shall be defined according to the service provider operational standard.

Table 72 — Program pattern

Program pattern	Description
0x0	Nonscheduled (other than defined as 0x1 to 0x7)
0x1	Regular program (every day, every day except weekend, only weekends etc.)
0x2	Program played about once a week
0x3	Program played about once a month
0x4	Program played several times a day
0x5	Division of long hour program
0x6 - 0x7	Undefined

8.3.34 Event group descriptor

When there is a relation between multiple events, the event group descriptor shall be used to group such events (see Table 73).

Table 73 — Event group descriptor

Syntax	Number of bits	Identifier
Event_group_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
group_type	4	uimsbf
event_count	4	uimsbf
for(i=0; i < event_count; i++){		
service_id	16	uimsbf
event_id	16	uimsbf
}		
if(group_type==4 || group_type==5){		
for(i=0; i < N; i++){		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
}		
}		
else {		
for(i=0; i < N; i++){		
private_data_byte	8	uimsbf
}		
}		
}		

The semantics for event group descriptor shall be:

- group_type: 4 bits field that shall indicate the group type in accordance with Table 74;
- event_count: 4 bits field that shall indicate the number of following event_id loops;
- service_id: 16 bits field that shall indicate the service_id of the related information service. It shall be the same as the used in PMT program_number field;
- event_id: 16 bits field that shall indicate the event_id of the related event;
- original_network_id: 16 bits field that shall indicate the original_network_id of related event transmitted at the time of event relay or event movement across other networks;
- transport_stream_id: 16 bits field that shall indicate the transport_stream_id of related event transmitted at the time of event relay or event movement across other networks.
- *private_data_byte*: 8 bits field that shall be defined according to user needs.

Table 74 — Group type

Group type	Description
0x1	Common event
0x2	Event relay
0x3	Event movement
0x4	Event relay to other networks
0x5	Event movement from other networks
0x0, 0x6 - 0xF	Undefined

8.3.35 SI parameter descriptor

The SI parameter descriptor shall be defined in accordance with Table 75.

Table 75 — SI parameter descriptor

Syntax	Number of bits	Identifier
SI_parameter_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
parameter_version	8	uimsbf
update_time	16	uimsbf
for(i=0;i<N;i++){		
table_id	8	uimsbf
table_description_length	8	uimsbf
for(j=0; j<N; j++){		
table_description_byte	8	uimsbf
}		
}		
}		

The semantics for SI parameter descriptor shall be:

- parameter_version: 8 bits field that shall indicate the version of SI parameters. This value shall be incremented by 1 when the parameter is updated;
- update_time: 16 bits field that shall define the less significant 16 bits of MJD when the parameter becomes valid;
- table_id: 8 bits field that shall indicate the table_id described in the table_description_table field;
- table_description_length: 8 bits field that shall indicate the byte length of table_description_byte;
- table_description_byte: 8 bits field. It shall be the sequence of tables describing the areas and defining parameters for each table specified in the operational standard of the service providers.

8.3.36 Broadcaster name descriptor

The broadcaster name descriptor shall describe the broadcaster name (see Table 76).

Table 76 — Broadcaster name descriptor

Syntax	Number of bits	Identifier
partial_reception_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i < N;i++){		
Char	8	uimsbf
}		
}		

The semantics for broadcaster name descriptor shall be:

- char: 8 bits field that shall be a string of characters informing the broadcaster name. The character details shall be specified in the operational standard of the service providers.

8.3.37 Component group descriptor

The component group descriptor shall define and identify the component grouping in the event (see Table 77).

Table 77 — Component group descriptor

Syntax	Number of bits	Identifier
component_group_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
component_group_type	3	uimsbf
total_bit_rate_flag	1	uimsbf
num_of_group	4	uimsbf
for(i=0; i < num_of_groupt; i++){		
component_group_id	4	uimsbf
num_of_CA_unit	4	uimsbf
for(j=0; j < num_of_CA_unit; j++){		
CA_unit _id	4	uimsbf
num_of_component	4	uimsbf
for(k=0;k<num_of_component; k++){		
component_tag	8	uimsbf
}		
}		
if(total_bit_rate_flag == 1){		
total_bit_rate	8	uimsbf
}		
text_length	8	uimsbf
for(i=0;i < text_length;i++){		
text_char	8	uimsbf
}		
}		
}		

The semantics for component group shall be:

— component_group_type: 3 bits field that indicates the component group type in accordance with Table 78;

— total_bit_rate_flag: 1 bit field that shall indicate the description status of the total bit rate in the component group in the event. When this value is set to "0", the total bit rate field of component group shall not exist in the corresponding descriptor. When this bit is set to "1", the field exists;

— num_of_group: 4 bits field that shall indicate the component group number in the event;

— component_group_id: 4 bit field that shall indicate the component group identifier in accordance with Table 79;

— num_of_CA_unit: 4 bits field that shall indicate a CA/non-CA unit within the component group;

— CA_unit_id: 4 bits field that shall describe the CA_unit_id, to which the component belongs in accordance with Table 80;

- num_of_component: 4 bits field that shall indicate the number of components which belong to corresponding component group and to CA/non-CA unit indicated in the immediately previous CA_unit_id;

- component_tag: 8 bits field that shall indicate the component tag value belonging to the component group;

- total_bit_rate: 8 bits field that shall describe the total bit rate of the component in the component group, rounding up the TS packet in each 1/4 Mbps;

- text_length: 8 bits field that shall indicate the size in bytes of the following component group description;

- text_char: shall be a string of characters describing the component group. The character details shall be defined according to the service provider operational standard.

Table 78 — Component group type

Component group type	Description
0	Multiview service
001 - 111	Undefined

Table 79 — Component group identifier

Component group identifier	Description
0x0	Main group
0x1 – 0xF	Sub group

Table 80 — CA_unit_id

CA_unit_id	Description
0x0	Non-CA unit group
0x1	CA unit group including default ES group
0x2 – 0xF	CA unit group other than above

8.3.38 SI prime_TS descriptor

The SI prime_TS descriptor shall identify the SI prime_TS (TS having a special format for SI transmission) and its transmission parameter (see Table 81).

Table 81 — SI prime_TS descriptor

Syntax	Number of bits	Identifier
SI_prime_TS_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
parameter_version	8	uimsbf
update_time	16	uimsbf
SI_prime_ts_network_id	16	uimsbf
SI_ prime_transport_stream_id	16	uimsbf
for(i=0; i < N;i++){		
table_id	8	uimsbf
table_description_length	8	uimsbf
for(j=0; j < N; j++){		
table_description_byte	8	uimsbf
}		
}		
}		

The semantics for SI prime_TS descriptor shall be:

- parameter_version: 8 bits field that shall indicate the version of SI parameter. It shall be incremented by 1 each time the parameter is updated;
- update_time: 16 bits field that shall use the same format as the less significant 16 bits of MJD updated when the defined parameter becomes to be valid;
- SI_prime_ts_network_id: 16 bits field that shall indicate the SI_prime_ts_network_id;
- SI_prime_transport_stream_id: 16 bits field that shall indicate the SI_prime_transport_stream_id;
- table_id: 8 bits field that shall indicate the table_id of next table_description_byte;
- table_description_length: 8 bits field that shall indicate the size in bytes of next table_description_byte;
- table_description_byte: 8 bits field that shall be a series of parameter description tables specified by the standards of service providers.

8.3.39 Board information descriptor

The board information descriptor shall indicate the title and content of information in a text format (see Table 82).

Table 82 — Board information descriptor

Syntax	Number of bits	Identifier
board_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
title_length	8	uimsbf
for(i=0; i < N;i++){		
title_char	8	uimsbf
}		
text_length	8	uimsbf
for(i=0; i < N;i++){		
text_char	8	uimsbf
}		
}		

The semantics for board information descriptor shall be:

- title_length: 8 bits field that shall indicate the size in bytes of next title;

- title_char: a string of characters that shall informe the board information title. The character details shall be defined according to the service provider operational standard;

- text_length: 8 bits field that shall indicate the size in bytes of next text;

- text_char: shall be a string of characters describing the board information content. The character details shall be defined according to the service provider operational standard.

8.3.40 LDT linkage descriptor

The LDT linkage descriptor shall be used to describe the information linkage obtained by LDT table (see Table 83).

Table 83 — LDT linkage descriptor

Syntax	Number of bits	Identifier
LDT_linkage_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
original_service_id	16	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
for(i=0; i < N;i++){		
description_id	16	Uimsbf
reserved_future_use	4	Uimsbf
description_type	4	Uimsbf
user_defined	8	bslbf
}		
}		

The semantics for LDT linkage descriptor shall be:

- original_service_id: 16 bits field that shall indicate the original_service_id of linked LDT sub-table;
- transport_stream_id: 16 bits field that shall indicate the ts_id of LDT sub-table where the linked LDT sub-table is included;
- original_network_id: 16 bits field that shall indicate the network_id of delivery system where the LDT sub-table is included;
- description_id: 16 bits field that shall indicate the id_number of the linked descriptor;
- description_type: 8 bits field that shall indicate the linked descriptor, according to Table 84;
- user_defined: the service provider can define these 8 bits independently.

Table 84 — Descriptor type

Value	Semantics
0x0	Undefined
0x1	Described with short_event_descriptor
0x2	Described with extended_event_descriptor (it is used an independent type without describing the item_description)
0x3	Described with extended_event_descriptor
0x4 – 0xE	Reserved for future use
0xF	Others (including descriptors and non-specified linkages)

8.3.41 Connected transmission descriptor

The connected transmission descriptor shall indicate the physical condition of terrestrial audio transmission path of connected transmission (see Table 85).

Table 85 — Connected transmission descriptor

Syntax	Number of bits	Identifier
connected_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
connected_transmission_group_id	16	uimsbf
segment_type	2	bslbf
modulation_type_A	2	bslbf
modulation_type_B	2	bslbf
reserved_future_use	2	bslbf
for(i=0; i < N;i++){		
additional_connected_transmission_info	8	uimsbf
}		
}		

The semantics for connected transmission descriptor shall be:

- connected_transmission_group_id: 16 bits field that provides the identification name of connected transmission group;
- segment_type: 2 bits field that indicates the segment type according to Table 86;
- modulation_type_A: 2 bits field that indicates the modulation type A according to Table 87;
- modulation_type_B: 2 bits field that indicates the modulation type B according to Table 87. If the segment type is equal to "1 segment", this field value is meaningless;
- additional_connected_transmission_info: 8 bits field that shall be used to store the additional information specified in the operational standard of service providers.

Table 86 — Segment type

Segment type	Description
00	1 segment
01	3 segments
10	Reserved for future use
11	Refer to TMCC signal

Table 87 — Modulation type

Modulation type	Description
00	Differential modulation
01	Synchronous modulation
10	Reserved for future use
11	Refer to TMCC signal

8.3.42 TS information descriptor

The TS information descriptor shall indicate the relation between service identifier and transmission layer during hierarchical transmissions (see Table 88).

Table 88 — TS Information descriptor

Syntax	Number of bits	Identifier
ts_information_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
remote_control_key_id	8	uimsbf
length_of_ts_name	6	uimsbf
transmission_type_count	2	uimsbf
for(i=0; i < length_of_ts_name; i++){		
ts_name_char	8	uimsbf
}		
for(j=0; j < transmission_type_count; j++){		
Transmission_type_info	8	bslbf
num_of_service	8	uimsbf
for(k=0;k<num_of_service; k++){		
service_id	16	uimsbf
}		
}		
for(l=0;l <N;l++){		
reserved_future_use	8	bslbf
}		
}		

The semantics for TS information descriptor shall be:

- remote_control_key_id: 8 bits field that shall indicate the virtual channel number to which the applicable TS shall be linked. Its transmission is mandatory;

- length_of_ts_name: 6 bits field that shall indicate the byte length of TS name description;

- transmission_type_count: 2 bits field that shall indicate the number of loops for subsequent information on the number of transmission types;

- ts_name_char: 8 bits field that shall be a series of TS name description field that describes the name of applicable TS. The character details shall be defined according to the service provider operational standard;

- transmission_type_info: 8 bits field which shall be used for discrimination of hierarchical layers and other parameters transmission. It shall be defined by operational standards of each service provider;

- num_of_service: 8 bits field that shall indicate the number of loops for subsequent service identifiers;

- service_id: 16 bits field that shall indicate the service identifier transmitted in each hierarchy of transmission type.

8.3.43 Extended broadcaster descriptor

The extended broadcaster descriptor shall specify the extended broadcaster identification information as terrestrial broadcaster identification and the relationships with other extended broadcasters and broadcasters of other networks (see Table 89).

Table 89 — Extended broadcaster descriptor

Syntax	Number of bits	Identifier
extended_broadcaster_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
broadcaster_type	4	uimsbf
Reserved_future_use	4	bslbf
If (broadcaster_type == 0x1){		
Terrestrial_broadcaster_id	16	uimsbf
Number_of_affiliation_id_loop	4	uimsbf
Number_of_broadcaster_id_loop	4	uimsbf
for(j=0; i <N1,i++){		
affiliation_id	8	uimsbf
}		
for(j=0;j<N2;j++){		
Original_network_id	16	uimsbf
broadcaster_id	8	uimsbf
}		
for(k=0;k <N3;k++){		
private_data_byte	8	bslbf
}		
}		
else if(broadcaster_type == 0X2){		
terrestrial_sound_broadcaster_id	16	uimsbf
number_of_sound_broadcaste_affiliation_id_loop	4	uimsbf
number_of_broadcaster_id_loop	4	uimsbf
for(i=0;i<N1;i++){		
Sound_broadcaster_affiliation_id	8	uimsbf
}		
for(j=0;j<N2;j++){		
Original_network_id	16	uimsbf
Broadcaster_id	8	uimsbf
}		
for(k=0;k<N3;k++){		
private_data_byte	8	bslbf
}		
}		
else		
for(i=0;i<N;i++){		
reserved_future_use	8	bslbf
}		
}		
}		

The semantics for extended broadcaster descriptor shall be:

- broadcaster_type: 4 bits field whose coding shall be according to Table 90;
- terrestrial_broadcaster_id: 16 bits field that shall indicate the terrestrial broadcaster indicated in this field;
- number_of_affiliation_id_loop: 4 bits field that shall indicate the number of loops for subsequent service identifiers;
- number_of_broadcaster_id_loop: 4 bit field that indicate the number of loops for subsequent broadcaster identifiers;
- affiliation_id: 8 bits field that shall be used for affiliate identification of applicable terrestrial broadcaster identifier;
- original_network_id: 16 bits field that shall work as a label to identify the original delivery system;
- broadcaster_id: 8 bits field that shall identify the broadcaster in the original network;
- terrestrial_sound_broadcaster_id: 16 bits field that shall indicate the terrestrial sound broadcaster indicated in this field;
- number_of_sound_broadcaster_affiliation_id_loop: 4 bits field that shall indicate the number of loops for subsequent sound broadcaster affiliation identifiers;
- number_of_sound_broadcaster_id_loop: 4 bits field that shall indicate the number of loops for subsequent terrestrial sound broadcaster identifiers;
- sound_broadcaster_affiliation_id: 8 bits field that be used to identify the sound broadcasting affiliation of the applicable terrestrial sound broadcaster identifier.

Table 90 — Broadcaster type

Value	Type
0x1	Digital terrestrial television broadcast
0x2	Reserved for future use
0x3 - 0xF	Undefined

8.3.44 Logo transmission descriptor

The logo transmission descriptor shall be used to describe information as characters string for simple logos and pointer for data of logos in the CDT format (see Table 91).

Table 91 — Logo transmission descriptor

Syntax	Number of bits	Identifier
logo_transmission_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
logo_transmission_type	8	uimsbf
if(logo_transmission_type == 0x01){		
reserved_future_use	7	bslbf
logo_id	9	uimsbf
reserved_future_use	4	bslbf
logo_version	12	uimsbf
download_data_id	16	uimsbf
}		
else if(logo_transmission_type == 0x02){		
reserved_future_use	7	bslbf
logo_id	9	uimsbf
}		
else if(logo_transmission_type == 0x03){		
for(i=0;i<N;i++){		
logo_char	8	uimsbf
}		
}		
else {		
for(j=0;j<M;j++){		
reserved_future_use	8	bslbf
}		
}		
}		

The semantics for logo transmission descriptor shall be:

- logo_transmission_type: 8 bits field that shall indicate the logo transmission scheme shown in the Table 92 (see ARIB STD-B21);

- logo_id: 9 bits data that shall denote the ID value of logo data defined in the applicable service (see ARIB STD-B21);

- download_data_id: 16 bits field that shall identify data from which the download shall be made. Its value shall be the same as CDT table_id_extension where logo data shall be located (see ARIB STD-B21);

- logo_version: 12 bits field that shall denote the version number of applicable logo_id (see ARIB STD-B21);

- logo_char: 8 bits field that shall describe the 8 units code character string for simple logo.

Table 92 — Logo transmission scheme

logo_transmission_type value	Meaning
0x01	CDT transmission scheme 1: when referring to CDT directly with download data identification
0x02	CDT transmission scheme 2: when referring to CDT indirectly with download data identification
0x03	Simple logo system
All, except above	Reserved for future use

8.3.45 Content availability descriptor

The content availability descriptor (see Table 93), shall describes information to control the recording and the video output. It shall be used in combination with digital copy control descriptor by broadcast service provider (copyright holder) to control the recording and the output of programs.

Table 93 — Content availability descriptor

Syntax	Number of bits	Identifier
content_availability_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
reserved_future_use	2	bslbf
image_constraint_token	1	bslbf
retention_mode	1	bslbf
retention_state	3	bslbf
encryption_mode	1	bslbf
for(i=0,i<N,i++){		
reserved_future_use	8	uimsbf
}		
}		

The semantics for content availability descriptor shall be:

- image_constraint_token: 1 bit field that shall indicate when the image quality of video output signal shall constrained. The resolution of video output signal shall be constrained when this field is set to "0", and does not to be constrained when this field is set to "1";
- retention_mode: when this 1 bit field is "0", temporal accumulation shall be possible even if copy is prohibited by the digital_recording_control_data of the digital copy control descriptor. When this field is "1", temporal accumulation shall not be possible;
- retention_state: 3 bits field that shall indicate the allowable time of temporal accumulation after the reception of contents. Its coding is shown in Table 94,

- encryption_mode (output protection bit): 1 bit field that shall indicate when the output of high-speed digital interface is protected. When this field is "0", the output of high-speed digital interface shall be protected. When this field is "1", protection shall not br required.

Table 94 — Time of temporal accumulation

Allowed time of temporal accumulation	Description
111	1 hour and half
110	3 hours
101	6 hours
100	12 hours
011	1 day
010	2 day
001	1 weeks
000	No limit

8.3.46 Carousel compatible composite descriptor

The carousel compatible composite descriptor uses descriptors defined in the data carousel broadcasting scheme as subdescriptors, and shall describe the accumulation control of stream-type contents etc. by applying the descriptive functions of the subdescriptors (see Table 95).

Table 95 — Carousel compatible composite descriptor

Syntax	Number of bits	Identifier
carousel_compatible_composite_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
sub_descriptior()		
}		
}		

The semantics for carousel compatible composite descriptor shall be:

- *sub_descriptor*( ): a subdescriptor shall be placed in this area. The descriptor in the module information area and the private area defined in the data carousel broadcasting scheme (see ABNT NBR 15606) shall be used as subdescriptors, and the descriptive function of each descriptor shall be inherited. The functions of subdescriptors shall be acoording to Annex F.

8.3.47 AVC video descriptor

The AVC video descriptor shall be used to describe the basic coding parameters of AVC video stream, according to ITU Recommendation H.264 and ISO/IEC 14496-10 (see Table 96). When this descriptor is not described in the PMT, the AVC stream shall not contain AVC still pictures or 24-hour AVC pictures. For more information, see ITU-T Recommendation H.222.0 and ISO/IEC 13818-1.

Table 96 — AVC video descriptor

Syntax	Number of bits	Identifier
AVC_video_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
profile_idc	8	uimsbf
constraint_set0_flag	1	bslbf
constraint_set1_flag	1	bslbf
constraint_set2_flag	1	bslbf
AVC_compatible_flags	5	bslbf
level_idc	8	uimsbf
AVC_still_present	1	bslbf
AVC_24_hour_picture_flag	1	bslbf
reserved	6	bslbf
}		

The semantics for AVC video descriptor shall be:

- profile_idc: shall show the AVC video stream profil (see ITU Recommendation H.264:2005, subclause 7.4.2.1 and ISO/IEC 14496-10);
- constraint_set0_flag: see ITU Recommendation H.264, subclause 7.4.2.1, and ISO/IEC 14496-10;
- constraint_set1_flag: see ITU Recommendation H.264, subclause 7.4.2.1, and ISO/IEC 14496-10;
- constraint_set2_flag: see ITU Recommendation H.264, subclause 7.4.2.1, and ISO/IEC 14496-10;
- AVC_compatible_flags: shall have the same value as reserved_zero_5bits in the sequence parameter specified in ITU Recommendation H.264 and ISO/IEC 14496-10;
- level_idc: shall show the AVC video stream level (see ITU Recommendation H.264, subclause 7.4.2.1 and ISO/IEC 14496-10);
- AVC_still_present: when this field is "1", the AVC video stream shall contain AVC still pictures. When this field is "0", the AVC video stream contains AVC still pictures;
- AVC_24_hour_picture_flag: when this field is "1", the AVC video stream shall contain 24 hours pictures, and shall be AVC access units containing presentation times exceeding 24 hours. When this field is "0", the AVC video stream should not contain AVC 24 hours pictures.

8.3.48 AVC timing and HRD descriptor

The AVC timing and HRD descriptor (see Table 97) shall be used to describe video stream time information and hypothetical reference decoder (HRD) information of the ITU Recommendation H.264 and ISO/IEC 14496-10. When the AVC video stream shall not transmit the video usability information (VUI) parameter, this descriptor shall be described in the PMT. For more information, see ITU Recommendation H.222.0 and ISO/IEC 13818-1.

Table 97 — AVC timing and HRD descriptor

Syntax	Number of bits	Identifier
AVC_timing_and_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
hrd_management_valid_flag	1	bslbf
Reserved	6	bslbf
picture_and_timing_info_present	1	bslbf
if(picture_and_timing_info_present == 1){		
90kHz_flag	1	bslbf
Reserved	7	bslbf
if(90kHz_flag == 0){		
N	32	uimsbf
K	32	uimsbf
}		
num_units_in_tick	32	uimsbf
}		
fixed_frame_rate_flag	1	bslbf
temporal_poc_flag	1	bslbf
picture_to_display_conversion_flag	1	bslbf
Reserved	5	bslbf
}		

The semantics for AVC timing and HRD descriptor shall be:

— hrd_management_valid_flag: when this 1 bit field is set to "1", the SEI buffering period shall be defined according to ITU Recommendation H.264:29005, Annex C, and ISO/IEC 14496-10, and shall be contained in the AVC. Besides, bytes shall be transferred from MBn to EBn according to the schedule of transfer to the coded picture buffer (CPB) in the network abstraction layer hypothetical reference decoder (NAL HRD). When this field is "0", the alternative method defined in the ITU Recommendation H.222.0 and ISO/IEC 13818-1 shall be used for transfer from MBn to EBn;

— picture_and_timing_info_present: when this field is "1", this descriptor shall have the 90kHz_flag and shall have the parameters for clock system mapping;

— 90kHz_flag: when this field is "1", AVC time base shall be 90 kHz. The AVC time base period shall be specified by AVC's time_scale defined in ITU Recommendation H.264:2005, Annex E, and ISO/IEC 14496-10. The *N* and *K* parameters shall describe the relationship between AVC's time_scale and system_clock_reference with the following equation, where *K* shall equal to or greater than N:

$$time\_sale = \frac{(N \times system\_clock\_frequency)}{K}$$

— num_units_in_tick: see ITU Recommendation H.264:2005, Annex E, and ISO/IEC 14496-10;

— fixed_frame_rate_flag: see ITU Recommendation H.264:2005, Annex E, and ISO/IEC 14496-10. When this flag is "1", the coded frame rate shall be constant within AVC video elementary stream. When this flag is "0", there is no information about AVC video stream frame rate in the descriptor;

- temporal_poc_flag: when this field is "1" and the fixed_frame_rate_flag is "1", the AVC video stream shall transmit picture order count (POC) information (see ITU Recommendation H.264:2005, Annex E, and ISO/IEC 14496-10. When this field is "0", the information about relationship between POC information of AVC video stream and the time shall not be transmitted;

- picture_to_display_conversion_flag: when this field is set to "1", the AVC video stream shall transmit information about displaying coded pictures. When this field is set to "0", the pic_struct_present_flag VUI parameter of AVC video stream shall have the value of "0".

8.3.49 Conditional playback descriptor

The conditional playback descriptor shall be in accordance with Table 98.

Table 98 — Conditional playback descriptor

Syntax	Number of bits	Identifier
Conditional_playback_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
CA_system_id	16	uimsbf
private_data	3	
CA_PID	13	
for(i=0;i<N;i++){		
private_data_byte	8	uimsbf
}		
}		

The semantics for conditional playback descriptor shall be:

- CA_system_id: 16 bits field that sahll indicate the number identifying the system access control;

- CA_PID: 13 bits field that shall define the TS PID including linkage information. The interpretation of this field is determined by the Table in which the descriptor is sent:

  - when the conditional playback descriptor is sent with PMT, the PID specified in the TS packet transmits the ECM;

  - when the conditional playback descriptor is sent with CAT, the PID specified in the TS packet transmits the EMM;

- private_data_byte: this is an 8 bit field. When the descriptor is included in PMT, the first byte of this field is reserved and the second and the third bytes are used to define the ECM PID for Kc transmission.

8.3.50 Conditional access descriptor

The conditional access descriptor shall define the information for conditional access management and elementary streams, respectively defined as EMM and ECM. It also can be used in TS_program_map or program_stream_map (see Table 99).

If any elementary stream is scrambled, the conditional access descriptor shall be present in the program that provides this ES. If any management information of conditional access shall exist within a transport stream, the descriptor of conditional access shall be present in the conditional access table (CAT).

If the conditional access descriptor is found in the TS_program_map_section (table id = 0x02), the CA_PID shall provide information related to access control, as ECM. Its presence as a program shall indicate information applicable for linkage off the entire program. In this same case, the CA_PID presence can extend the information applicable for linkage with the program elements, condition that also shall be performed for private data.

When the conditional access descriptor is found in CA_section (table_id = 0x01), the CA_PID shall provide or not information for access control management, as EMM.

Table 99 — Conditional access descriptor

Syntax	Number of bits	Identifier
Conditional_playback_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
CA_system_id	16	uimsbf
Reserved	3	bslbf
CA_PID	13	uimsbf
for(i=0;i<N;i++){		
private_data_byte	8	uimsbf
}		
}		

The semantics for conditional access descriptor shall be:

- CA_system_ID: 16 bits field that shall be responsible for indicating the conditional access type of applicable system to link ECM and/or EMM streams. The coding for this field shall be private;
- CA_PID: 13 bits field that shall indicate the transport stream PID which contains ECM or EMM information for the conditional access system as specified in the CA_system_id.

8.3.51 AAC audio descriptor

8.3.51.1 General information

The identification value for the descriptor tag of the AAC audio descriptor shall be 0x7C. The descriptor length shall indicate the number of data bytes.

The AAC_descriptor shall identify HE AAC encoding of elementary streams according to the ISO/IEC 14496-3, in order to provide configuration information for the receiver.

The descriptor shall be located in the PMT table from the PSI and shall be used only once in the program mapping section followed by the relevant information in the ES_info_length field for all streams encoded with HE AAC audio according to the the ISO/IEC 14496-3.

The descriptor tag provides original identification of presence of the elementary stream encoded with AAC audio. Other optional fields in the descriptor may be used to indicate the mode of the AAC audio component (AAC_type) encoded in the stream.

8.3.51.2 AAC descriptor syntax

The AAC descriptor shall be present in the PMT table from PSI, in order to identify which streams shall be encoded with AAC audio. The descriptor shall be located only once in the mapping section, followed by the relevant information in the ES_info_length field.

8.3.51.3 AAC descriptor semantics

The syntax of the AAC_descriptor provides information about the MPEG-4 AAC and MPEG-4 HE AAC elementary streams that shall be identified in the sections of the PMT table of the PSI. The descriptor shall be located only once in the program mapping section, followed by the relevant information in the ES_info_length field for any stream with MPEG-4 AAC or MPEG-4 HE AAC audio.

Annex A
(normative)

Date and time conversion

Time shall be in accordance with UTC-3 time (official time).

The conversion of Modified Julian Date (MJD) and Brazil standard time shall be in accordance with Figure A.1.

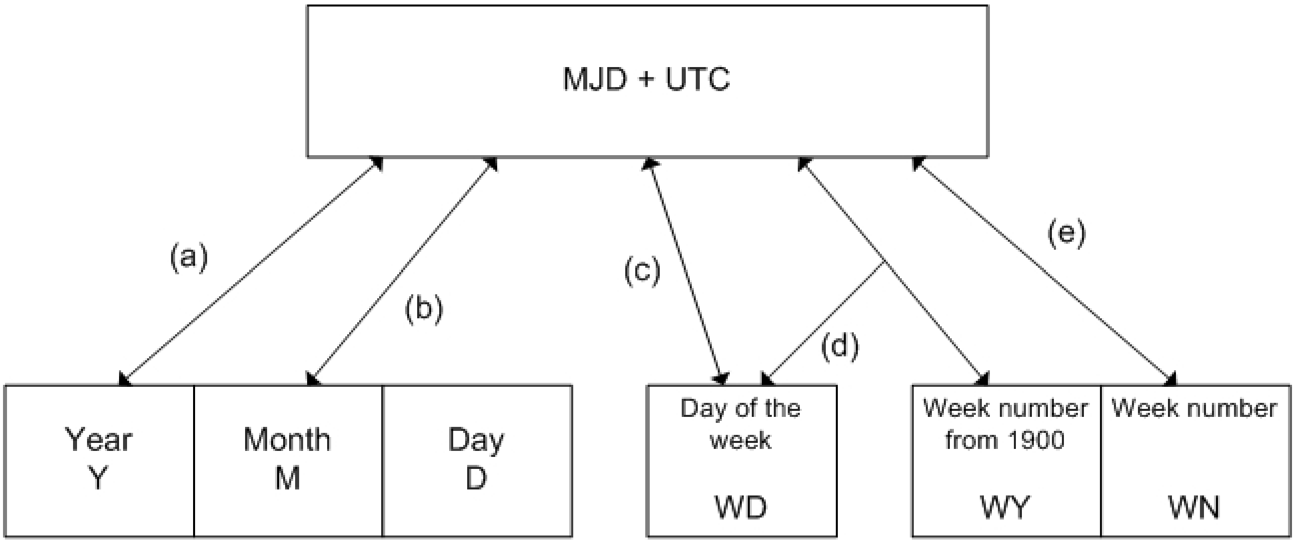


Figure A.1 — Conversion of MJD and UTC-3

The relation between year-month-date and MJD shall be in accordance with the following expressions:

a) method to find year, month and date (Y, M, D) from MJD:

Y' = int [(MJD – 15078,2) / 365,25]

M' = int {[MJD – 14956,1 - int (Y' 365,25)] / 30,6001}

D = MJD – 14 956 - int (Y' x 365,25) - int (M' x 30,6001)

For M' = 14 or M' = 15, K = 1. For other cases, K = 0

Y = Y' + K

M = M' - 1 - K x 12

b) method to find MJD from year, month and date (Y, M, D):

MJD = 14956 + D + int[(Y – L) × 365.25] + int[(M + 1 + L × 12) × 30.6001]

For M = 1 or M = 2, L = 1. In other cases: L = 0

c) method to find week day (WD) from MJD:

$$WD = [(MJD + 2) \bmod 7] + 1$$

d) method to find MJD from WY, WN and WD:

$$MJD = 15012 + WD + 7 \times \{WN + int[(WY \times 1461 / 28) + 0.41]\}$$

e) method to find WY and WN from MJD:

$$W = int[(MJD / 7) – 2144.64]$$

$$WY = int[(W \times 28 / 1461) – 0.0079]$$

$$WN = W – int[(WY \times 1461 / 28) + 0.41]$$

where

Y is the year from 1900 (for example, 2003 is Y=103);

M is the month (January = 1 to December = 12);

D is the day (1 to 31);

WY is the week number year since 1900;

WN is the week number in accordance with ISO 8601;

WD is the week day (Monday = 1 to Sunday = 7);

K, L, M', W, Y' are intermediate variables;

Int is the integer part, ignoring remainder;

mod 7 is the remainder number after dividing integer by 7.

EXAMPLE MJD = 5218 W = 4315

Y = (19)82 WY = (19)82

M = 9 (September) WN = 36

D = 6 WD = 1 (Monday)

NOTE These formulas are effective from March 1, 1900 to February 28, 2100.

Annex B
(normative)

CRC decoder

The 32 bits CRC decoder shall be operated in bit level and shall be constituted of 14 adders (+) and 32 delay elements z(i). Input of the CRC decoder shall be added to the output of z(31), and the result shall be divided into the input of z(0) and input of one side of the rest of each of the adders.

Input of the other side of the rest of the adders shall be refined in z(i), and output of the rest of each adders shall be connected to the input of z(i+1), with i= 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22, 25 (see.Figure B.1).

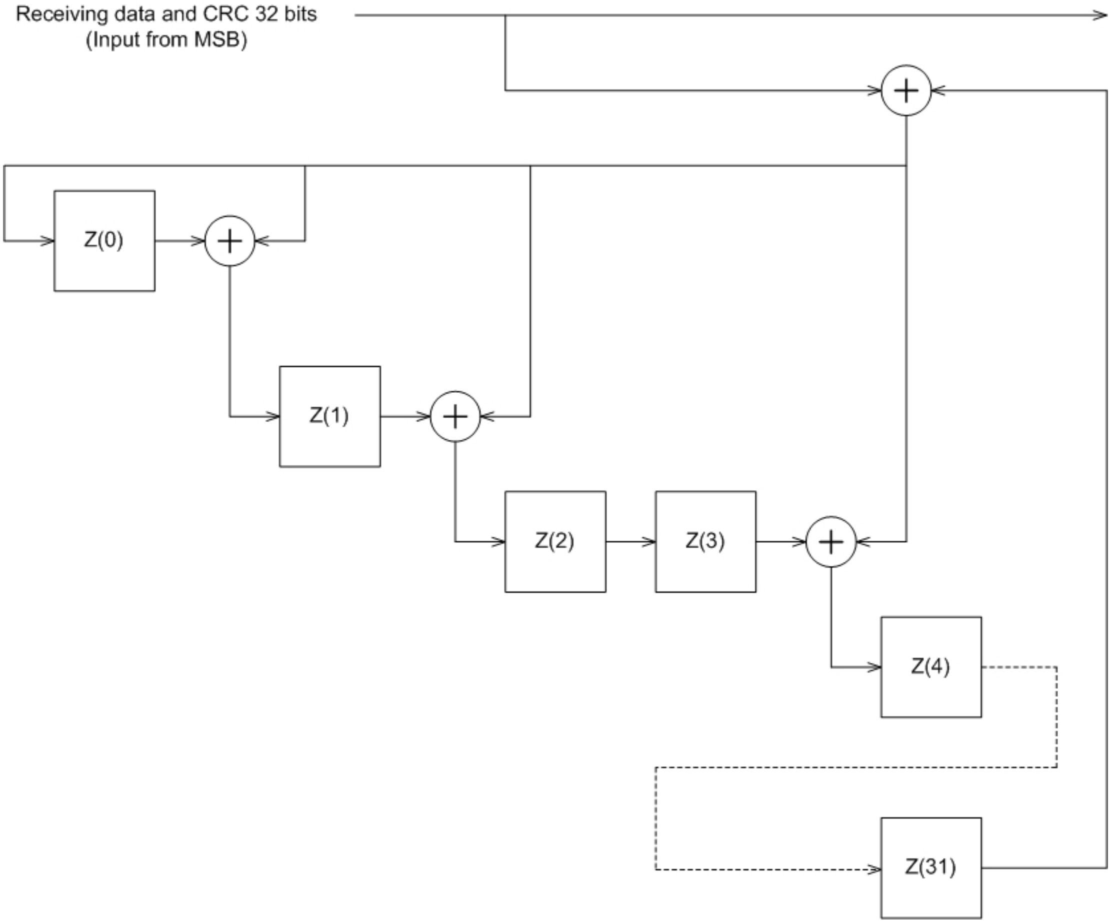


Figure B.1 — 32 bits CRC decoder model

CRC shall be calculated by the following polynomial:

x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

Reception at the input of the CRC decoder shall be made in byte unit. Each byte shall be shifted to CRC decoder in 1 bit each, in the order of MSB.

EXEMPLE Where byte 0x01 (last byte of start code prefix), first 7 "0"s are input to the CRC decoder and then 1 "1" is input. Output of each delay element z(i) is set to initial value "1", before data of 1 section is processed by CRC. After initialized, each byte of section including 4 bytes of CRC_32 is provided to input of the CRC decoder. After the last bit of the last CRC_32 byte is shifted to the decoder, which means that when added to output z(31) and then input to z(0), output of all delay element z(i) is read out. When there is no error, output of each z(i) is zero. In the CRC encoder, CRC_32 field is encoded in such value that it is assured.

Annex C
(normative)

Genre designation in content descriptor

The event genre of the content descriptor shall be defined by the classification presented in Table C.1. For events harder to classify, the genre should be defined as others.

Genre and sub-genre classification shall be in accordance with Table C.1 and C.2.

Table C.1 — Descriptor genre classification

Genre classification	Content descriptor
0x0	News
0x1	Sports
0x2	Education
0x3	Soap opera
0x4	Mini-series
0x5	Series
0x6	Variety
0x7	Reality show
0x8	Information
0x9	Comical
0xA	Children
0xB	Erotic
0XC	Movie
0xD	Raffle, television sales, prizing
0xE	Debate/interview
0xF	Other

Table C.2 — Genre and sub-genre classification

Large genre	Middle genre	Content description
0x0	Not used	News
0x0	0x0	News
0x0	0x1	Report
0x0	0x2	Documentary
0x0	0x3	Biography
0x0	0x4	
0x0	0x5	
0x0	0x6	
0x0	0x7	
0x0	0x8	
0x0	0x9	
0x0	0xA	
0x0	0xB	
0x0	0xC	
0x0	0xD	
0x0	0XE	
0x0	0XF	Other
0x1	Not used	Sports
0x1	0x0	Sports
0x1	0x1	
0x1	0x2	
0x1	0x3	
0x1	0x4	
0x1	0x5	
0x1	0x6	
0x1	0x7	
0x1	0x8	
0x1	0x9	
0x1	0xA	
0x1	0xB	
0x1	0xC	
0x1	0xD	
0x1	0xE	
0x1	0xF	Other
0x2	Not used	Educative
0x2	0x0	Educative
0x2	0x1	
0x2	0x2	
0x2	0x3	
0x2	0x4	
0x2	0x5	
0x2	0x6	
0x2	0x7	

Table C.2 (continuation)

Large genre	Middle genre	Content description
0x2	0x8	
0x2	0x9	
0x2	0xA	
0x2	0xB	
0x2	0XC	
0x2	0xD	
0x2	0XE	
0x2	0XF	Other
0x3	Not used	Soap opera
0x3	0x0	Soap opera
0x3	0x1	
0x3	0x2	
0x3	0x3	
0x3	0x4	
0x3	0x5	
0x3	0x6	
0x3	0x7	
0x3	0x8	
0x3	0x9	
0x3	0xA	
0x3	0xB	
0x3	0xC	
0x3	0xD	
0x3	0XE	
0x3	0XF	Other
0x4	Not used	Mini-series
0x4	0x0	Mini-series
0x4	0x1	
0x4	0x2	
0x4	0x3	
0x4	0x4	
0x4	0x5	
0x4	0x6	
0x4	0x7	
0x4	0x8	
0x4	0x9	
0x4	0xA	
0x4	0xB	
0x4	0xC	
0x4	0xD	
0x4	0XE	
0x4	0XF	Other

Table C.2 (continuation)

Large genre	Middle genre	Content description
0x5	Not used	Series
0x5	0x0	Series
0x5	0x1	
0x5	0x2	
0x5	0x3	
0x5	0x4	
0x5	0x5	
0x5	0x6	
0x5	0x7	
0x5	0x8	
0x5	0x9	
0x5	0xA	
0x5	0xB	
0x5	0xC	
0x5	0xD	
0x5	0XE	
0x5	0XF	Other
0x6	Not used	Variety show
0x6	0x0	Auditorium
0x6	0x1	Show
0x6	0x2	Musical
0x6	0x3	Making of
0x6	0x4	Feminine
0x6	0x5	Game show
0x6	0x6	
0x6	0x7	
0x6	0x8	
0x6	0x9	
0x6	0xA	
0x6	0xB	
0x6	0xC	
0x6	0xD	
0x6	0XE	
0x6	0XF	Other
0x7	Not used	Reality show
0x7	0x0	Reality show
0x7	0x1	
0x7	0x2	
0x7	0x3	
0x7	0x4	
0x7	0x5	
0x7	0x6	
0x7	0x7	
0x7	0x8	
0x7	0x9	
0x7	0xA	
0x7	0xB	

Table C.2 (continuation)

Large genre	Middle genre	Content description
0x7	0xC	
0x7	0xD	
0x7	0xE	
0x7	0xF	Other
0x8	Not used	Information
0x8	0x0	Cooking
0x8	0x1	Fashion
0x8	0x2	Country
0x8	0x3	Health
0x8	0x4	Travel
0x8	0x5	
0x8	0x6	
0x8	0x7	
0x8	0x8	
0x8	0x9	
0x8	0xA	
0x8	0xB	
0x8	0xC	
0x8	0xD	
0x8	0xE	
0x8	0xF	Other
0x9	Not used	Comical
0x9	0x0	Comical
0x9	0x1	
0x9	0x2	
0x9	0x3	
0x9	0x4	
0x9	0x5	
0x9	0x6	
0x9	0x7	
0x9	0x8	
0x9	0x9	
0x9	0xA	
0x9	0xB	
0x9	0xC	
0x9	0xD	
0x9	0xE	
0x9	0xF	Other
0xA	Not used	Children
0xA	0x0	Children
0xA	0x1	
0xA	0x2	
0xA	0x3	
0xA	0x4	
0xA	0x5	
0xA	0x6	

Table C.2 (continuation)

Large genre	Middle genre	Content description
0xA	0x7	
0xA	0x8	
0xA	0x9	
0xA	0xA	
0xA	0xB	
0xA	0XC	
0xA	0xD	
0xA	0XE	
0xA	0XF	Other
0xB	Not used	Erotic
0xB	0x0	Erotic
0xB	0x1	
0xB	0x2	
0xB	0x3	
0xB	0x4	
0xB	0x5	
0xB	0x6	
0xB	0x7	
0xB	0x8	
0xB	0x9	
0xB	0xA	
0xB	0xB	
0xB	0xC	
0xB	0xD	
0xB	0xE	
0xB	0xF	Other
0XC	Not used	Movie
0XC	0x0	Movie
0XC	0x1	
0XC	0x2	
0XC	0x3	
0XC	0x4	
0XC	0x5	
0XC	0x6	
0XC	0x7	
0XC	0x8	
0XC	0x9	
0XC	0xA	
0XC	0xB	
0XC	0xC	
0XC	0xD	
0XC	0xE	
0XC	0xF	Other

Table C.2 (continuation)

Large genre	Middle genre	Content description
0xD	Not used	Raffle, television sales, prizing
0xD	0x0	Raffle
0xD	0x1	Television sales
0xD	0x2	Prizing
0xD	0x3	
0xD	0x4	
0xD	0x5	
0xD	0x6	
0xD	0x7	
0xD	0x8	
0xD	0x9	
0xD	0xA	
0xD	0xB	
0xD	0xC	
0xD	0xD	
0xD	0xE	
0xD	0xF	Other
0XE	Not used	Discussion/interview
0xE	0x0	Discussion
0xE	0x1	Interview
0xE	0x2	
0xE	0x3	
0xE	0x4	
0xE	0x5	
0xE	0x6	
0xE	0x7	
0xE	0x8	
0xE	0x9	
0xE	0xA	
0xE	0xB	
0xE	0xC	
0xE	0xD	
0xE	0xE	
0xE	0xF	Other

0XF	Not used	Other
0XF	0x0	Adult cartoon
0XF	0x1	Interactive
0XF	0x2	Policy
0XF	0x3	Religion
0XF	0x4	
0XF	0x5	
0XF	0x6	
0XF	0x7	

Table C.2 (continuation)

Large genre	Middle genre	Content description
0XF	0x8	
0XF	0x9	
0XF	0xA	
0XF	0xB	
0XF	0xC	
0XF	0xD	
0XF	0xE	Enginnering services
0XF	0xF	Other

Annex D
(informative)

Example of bit definition for the digital copy control descriptor by the service provider

A example of service provider defining bit of digital copy control descriptor is shown in Table D.1.

Table D.1 — Digital copy control descriptor

Syntax	Number of bits	Identifier
digital_copy_control_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
digital_recording_control_data	2	bslbf
maximum_bitrate_flag	1	bslbf
component_control_flag	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type != 00){		
APS_control_data	2	bslbf
}		
Else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
if(component_control_flag == 1){		
component_control_length	8	uimsbf
For(j=0;j<N;j++){		
component_tag	8	uimsbf
digital_recording_control_data	2	bslbf
Maximum_bitrate_flag	1	bslbf
reserved_future_use	1	bslbf
copy_control_type	2	bslbf
if(copy_control_type != 00){		
APS_control_data	2	bslbf
}		
else{		
reserved_future_use	2	bslbf
}		
if(maximum_bitrate_flag == 1){		
maximum_bitrate	8	uimsbf
}		
}		
}		

For the example of Table D.1, the semantics for the digital copy control descriptor is the following one:

- copy_control_type: 2 bits field that shall indicate the type information to control copy generation and encoding in accordance with Table D.2;

- digital_recording_control_data: 2 bits field that shall indicate information to control copy generation and encoding in accordance with Table D.3;

- APS_control_data: 2 bits field that indicates data from control analog output copy when the copy_control_type is 01 and encoding in accordance with Table D.4.

Table D.2 — Copy control type information

Copy control type information	Description
00	Not defined
01	Output with encoding for video interface [a]
10	Not defined
11	Output without encoding for video interface
[*a] It is used the encoding method specified by the service provider.	

Table D.3 — Digital recording control data

Digital recording control data	Description	
	When copy_control_type is 11	When copy_control_type is 01
00	Can be copied without control condition	Can be copied without control condition
01	Not used	Forbidden copy
10	Can be copied once only	Can be copied once only
11	Forbidden copy	Forbidden copy

Table D.4 — Analog output copy control data

Analog output copy control data	Description
00	Can be copied without control condition
01	Video resolution limited to 350 000 pixels
10	
11	

Annex E
(normative)

Area_code specification

The numbering of each microregion of the country has a 5 digits identification. To suit this information in the 12 bits specified for area_code, the lower 7 bits shall be stuffed with the three last IBGE identification digits for microregion converted to binary form, with the limitation that is cannot exceed 127.

NOTE The Brazilian Geography and Statistics Institute (IBGE) defines values for each Federation Unit and micro-regions existing in the country.

If the number needs less than 7 bits for its identification, the bits to the left shall be stuffed with zero. The 5 most significant bits refers to Federative Unit (see Figure E.1) and shall be stuffed in accordance with Table E.1.

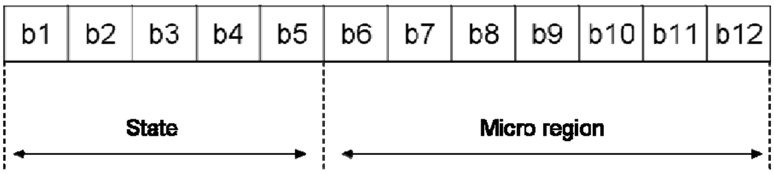
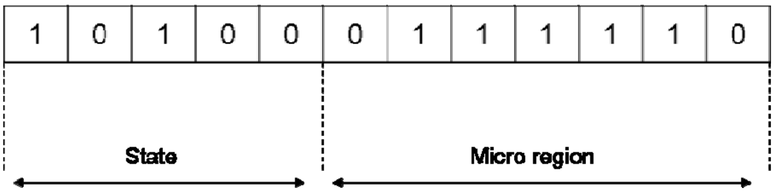


Figure E.1 – Distribution of bits for area_code

Table E.1 — State identification

Identification	State
00001	Rondônia
00010	Acre
00011	Amazonas
00100	Roraima
00101	Pará
00110	Amapá
00111	Tocantins
01000	Maranhão
01001	Piauí
01010	Ceará
01011	Rio Grande do Norte
01100	Paraíba
01101	Pernambuco
01110	Sergipe
01111	Alagoas
10000	Bahia
10001	Minas Gerais
10010	Espírito Santo
10011	Rio de Janeiro
10100	São Paulo
10101	Paraná
10110	Santa Catarina
10111	Rio Grande do Sul
11000	Mato Grosso do Sul
11001	Mato Grosso
11010	Goiás
11011	Distrito Federal
11100 - 11111	Reserved

EXAMPLE SP - Metropolitan Region - Mogi das Cruzes:
Value tabled by IBGE: 35062
Last 3 digits used for area_code: 62 (0111110)B
Identification of Sao Paulo according Table E.1: (10100)B
In this example, area_code value is given in accordance with Figure E.2.



1	0	1	0	0	0	1	1	1	1	1	0

Figure E.2 — Example - *Area_code* for Mogi das Cruzes' region

Annex F
(normative)

Subdescriptors used in the carousel compatible composite descriptor

The descriptors in the module information area and the private area defined in the data carousel transmission scheme area shall be used in the subdescriptors area of the carousel compatible composite descriptor. The tag values of the subdescriptors are listed in Table F.1.

NOTE This Annex specifies the functions of those subdescriptors regarding to their use for service information (type descriptor and name descriptor).

Table F.1 — Subdescriptors used in the carousel compatible composite descriptor

Tag value	Descriptor	Function	Module information area	Private area
0x01	Type_descriptor	Module type (MIME form etc.)	X	
0x02	Name_descriptor	Module name (file name)	X	
0x03	Info_descriptor	Module information (character type)	X	X
0x04	Module_link_descriptor	Link information (module id)	X	
0x05	CRC32_descriptor	CRC32 of total module	X	
0x06	Location_descriptor		X	X
0x07	Est_download_time_descriptor	Estimated download time	X	X
0x08 – 0x7F	Reserved for future use			
0x80 – 0xBF	Available to a broadcaster. Any value in this range may be defined as a tag value of a descriptor			

The type descriptor (see Table F.2) shall indicate the type of objected addressed by the carousel compatible composite descriptor containing this descriptor.

Table F.2 — Type descriptor

Syntax	Number of bits	Identifier
type_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

The semantics for type descriptor shall be:

- text_char: 8 bits fields that shall be a series of areas indicates the type of media complying with RFC 1521 and RFC 1590. The characters details shall be defined based on the operational standard of the service provider.

The name descriptor (see Table F.3) shall indicate the file name for accumulating the object addressed by the carousel compatible composite descriptor containing this descriptor.

Table F.3 — Name descriptor

Syntax	Number of bits	Identifier
name_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimsbf
for(i=0;i<N;i++){		
text_char	8	uimsbf
}		
}		

The semantics for the name descriptor shall be:

- *text_char*: 8 bits fields that shall be a series of areas indicating the file name for accumulating the applicable object.

The syntax for others descriptors showed on Table F.1 shall be in accordance with the ABNT NBR 15606-3:2007, subclause 5.4.1.

Annex G
(normative)

Specification for tuning physical and logical channel

The physical channel shall be defined within a frequency bandwidth of 6 MHz.

The virtual channel shall be the identification of several services existing in a physical channel.

The virtual channel shall be obtained from the field remote_control_key_id of TS_information_descriptor, located in second loop of NIT.

The differentiation among several services within same virtual channel shall be done using the fields service_type and service_number, contained in the least significant 5 bits of field service_id, according to the following:

— remote_control_key_id: shall assume values between 1 and 99, inclusive;

— service_type: the information of service_type can be obtained from service_id (see Annex H);

— service_number: the service number shall be the information of service_number + 1. The information of service_number can be obtained from service_id (see Annex H).

For the current broadcasters in analog system, in transition to digital system, the value of the field remote_control_key_id designated for the digital channel of a broadcaster shall be equal to the numbering of its analog channel.

The way that the information of remote_control_key_id will be stored by the receiver shall obligatorily be in accordance with Figure G.1, however the way that service_type and service_number will be shown to the user can vary depending on set top box implementation.

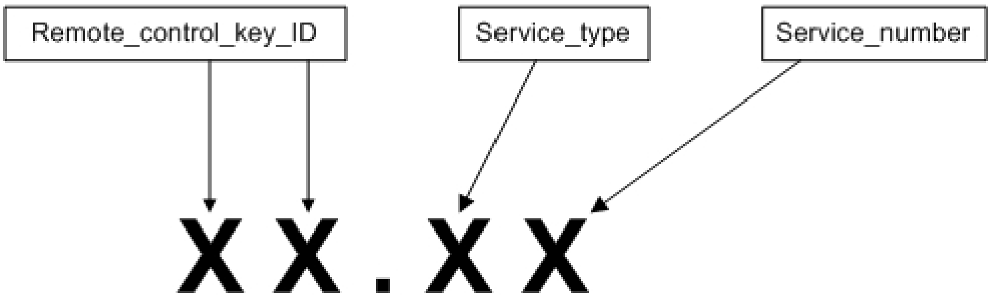


Figure G.1 — Identification of logical channel

Table G.1 and G.2 show examples of scenarios of channel tuning typing its physical or logical numbers.

Table G.1 — Scenarios in 13 segments receivers

Service for tuning	Option A	Option B	Option C	Option D	Option E	Option F
05.01	5	05	5.1	05.1	5.01	05.01
05.08	X	x	5.8	05.8	5.08	05.08
23.01	X	23	x	23.1	X	23.01
23.08	X	x	x	23.8	X	23.08
05.11	X	x	x	x	5.11	05.11
05.38	X	x	x	x	5.38	05.38
23.21	X	x	x	x	X	23.21
23.38	X	X	x	x	X	23.38

Table G.2 — Scenarios in 1 segment receivers

Service for tuning	Option A	Option B	Option C	Option D	Option E	Option F
23.31	X	23	x	23.1	X	23.31
23.38	X	x	x	23.8	X	23.38

Annex H
(normative)

Specification of the fields related to the broadcaster identification: original_network_id, network_id e service id

H.1 General

The fields for original_network_id, network_id and service_id shall be completed with the codes identifiers provided by ANATEL (main studio: code 248) in the Brazilian territory, based on the prefix of each Brazilian generating station.

The 16 bits field original_network_id, located in the network information table, shall be designated as unique identifier of each generating station.

The 16 bits field of network_id, also located in the network information table, shall be designated as unique identifier of each generating and shall have the same value of *original_network_id*.

The rebroadcasting station shall inherit original values of original_network_id and network_id fields from the signal provider.

The 16 bits field service_id located in the service descriptor table shall be unique by generator and have the type and number identifier for the service to be broadcasted.

H.2 Original_network_id

The original_network_id shall identify uniquely each generating station existing in Brazil. Therefore, it shalll be linked to each generating station empowered by ANATEL. This prefix is represented by six digits where the first two digits always display ZY (for example, ZYA205). The third value (left to right) is represented by the letters A, B, P, Q e T. and the last three values are represented by a number ranging from 000 to 999.

To compose the original_network_id value, the first two letters shall be discarded and the third letter (left to right) shall be defined a value according to the Table H.1.

Table H.1 — Corresponding numbers to letters

Letter	Number
A	0
B	1
P	2
Q	3
T	4

EXAMPLE A generation station that has the identification ZYB205 shall discard the first two letters (ZY) and replace the letter B in the value 1 as showed in Table H.1. Thus the original_network_id is. $(1205)_D$ Converting this amount to Hex, the value of original_network_id will be 0x04B5.

H.3 Service_id

The fields related to service_id shall be unique by generating station and shall contain the type and number identifier of service to be broadcasted.

IN order to have service_id unique by generator, it shall be inserted into its higher 11 bits the value of original_network_id field. The following 2 bits of service_id represent the parameters for service type defined in Table H.2.

Table H.2 — Service type classification (*service_type*)

Service type	Identification
00	TV
01 or 10	Data (except one-seg)
11	one-seg

The following three bits shall represent the service number transmitted by broadcasters for each service_type, ranging from 000 to 111, representing a maximum of 8 services (see Figure H.1). The value 000 shall be used for the broadcaster main service.

EXAMPLE A broadcaster who has the original_network_id equal to $(00010000001)_B$, shall use the service_id $(0001000000100000)_B$ for the first television service $(0001000000111000)_B$ for the first one-seg service.

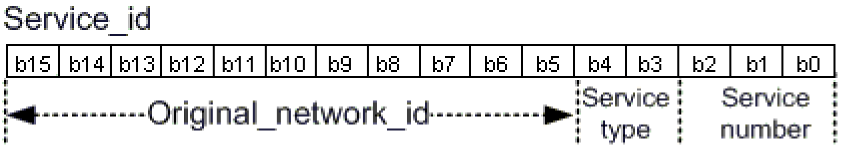


Figure H.1 — The service_id standardization

H.4 Network_id

The network_id value of a generator shall be set with the same value of original_network_id set. The rebroadcasting stations shall inherit values from the signal provider.

Annex I
(normative)

Broacasting specification for profiles H-EIT, M-EIT e L-EIT

I.1 General

The Brazilian digital television system shall allow the broadcasting of 3 EIT types: H-EIT, M-EIT and L-EIT. The event information obtained through these tables shall be shown in the area corresponding to each EPG type (see Table I.1).

The receiver that receives the information from H-EIT table, shall display it in the reserved are for "H-type EPG". If it receives an M-EIT, it shall display it in the reserved area for "M-type EPG". The same happens when receiving a L-EIT. It shall be displayed in the reserved area for "L-type EPG".

Table I.1 — Definition of the names used by EIT

Name	Definition
EIT	Term used to identify the H-EIT, M-EIT e L-EIT
EIT[p/f]	Term used to identify the H-EIT[p/f]/M-EIT[p/f]/L-EIT[p/f]
EIT[schedule]	Term used to identify the H-EIT[schedule basic]/H-EIT[schedule extended]
H-EIT	Term used to identify the H-EIT[p/f]/H-EIT[schedule basic]/H-EIT[schedule extended
M-EIT	Term used to identify the M-EIT[p/f]/M-EIT[p/f after]
L-EIT	Term used to identify the L-EIT[p/f]/L-EIT[p/f after]

I.2 Identification

PID to identify the different EIT types shall follow the Table I.2:

Table I.2 — PID of different types of EIT

EIT type	PID
H-EIT	0x0012
M-EIT	0x0026
L-EIT	0x0027

The "table_id" identification of EIT sections shall follow the Table I.3.

Table I.3 — Table_id of EIT sections

EIT type	PID
H-EIT[p/f]	0x4E
H-EIT [*schedule basic*]	0x50 - 0x57
H-EIT [*schedule extended*]	0x58 - 0x5F
M-EIT	0x4E
L-EIT	0x4E

I.3 Possible descriptors of each EIT type

The Table I.4 shows the possible descriptors in each EIT type.

Table I.4 — Possible descriptors of each EIT type

Nº	Tag value	Descriptor	H-EIT [p/f]	H-EIT [Schedule basic]	H-EIT [Schedule extended]	M-EIT [p/f]	M-EIT [p/f after]	L-EIT [p/f]	L-EIT [p/f after]
1	0X4D	Short event descriptor	Mandatory	Mandatory	Not applied	Mandatory	Mandatory	Mandatory	Mandatory
2	0x4E	Extended event descriptor	Optional	Not applied	Optional	Not applied	Not applied	Not applied	Not applied
3	0x50	Component descriptor	Mandatory [a]	Mandatory [a]	Not applied	Mandatory [a]	Mandatory [a]	Not applied	Not applied
4	0x54	Content descriptor	Optional	Optional	Not applied	Optional	Optional	Optional	Optional
5	0xC1	Digital copy control descriptor	Optional	Optional	Not applied	Optional	Optional	Optional	Optional
6	0xC4	Audio component descriptor	Mandatory [a]	Mandatory [a]	Not applied	Mandatory [a]	Mandatory [a]	Not applied	Not applied
7	0xC7	Data contents descriptor	Optional	Optional	Not applied	Optional	Optional	Not applied	Not applied
8	0xD5	Series descriptor	Optional	Optional	Not applied	Optional	Optional	Not applied	Not applied
9	0xD6	Event group descriptor	Optional	Optional	Not applied	Optional	Optional	Not applied	Not applied
10	0x55	Parental rating control	Mandatory	Mandatory	Not applied	Mandatory	Mandatory	Mandatory	Not applied
11	0XDE	Content availability descriptor	Optional	Optional	Not applied	Optional	Optional	Optional	Optional
12	0x42	Stuffing descriptor	Optional	Optional	Optional	Optional	Optional	Optional	Optional
13	0xD9	*component_group_ descriptor*	Optional	Optional	Optional	Optional	Optional	Optional	Optional

[a] At least one descriptor shall be inserted.

I.4 Basic type EIT delivery

The EIT types which always shall be transmitted for each service shall be called basic types EIT delivery. The EIT types and where they shall be transmitted are in the Table I.5.

Table I.5 — EIT types

Receiver type	Basic type of EIT delivery	Broadcasting layer
Fixed receiver	H-EIT	Same layer where the service is broadcasted
Mobile receiver	M-EIT	Same layer where the service is broadcasted
One-seg receiver	L-EIT	Same layer where the service is broadcasted

The indication of what basic type shall be broadcasted: H-EIT, M-EIT and L-EIT shall be made within the loop of each service through EIT_user_defined_flag section, which shall be 3-bit field described in the SDT table.

I.5 Extended type of EIT delivery

In addition to the basic types which can be transmitted, if necessary, it shall be possible to transmit other EIT type for a same service. This option shall be known as extended type of EIT delivery.

When transmitting a partial segment, the transmission of basic delivery type for EIT is mandatory in relation to partial segment: L-EIT, but if it is desired to provide more service options linked to "H-type EPG", it shall be possible to transmit the H-EIT as an extended type of EIT delivery. However, there are some restrictions for using the extended types of EIT delivery.

I.6 Restrictions for using the extended types of EIT delivery

The Tables I.6 to I.11 shows six settings simulating the broadcasting possibilities of Brazilian system, with the same transmission possibility of basic and extended EIT.

Table I.6 — Transmission in low protection segments (layer A) for a fixed receiver

Service	Segment settings
	layer A
Low protection segment	H-EIT (basic)

Table I.7 — Transmission in low protection segments lLayer A) for a mobile receiver

Service	Segment settings
	Layer A
Low protection segment	M-EIT (basic)
	H-EIT (extended)

Table I.8 — Transmission of a high protection segment (layer A) for reception of one-seg and low protection segments (layer B) for fixed receivers

Service	Segment setting	Segment setting
	Layer A	Layer B
High protection segment	M-EIT (extended)	H-EIT (extended)
Low protection segment	Not used	H-EIT (Basic)

Table I.9 — Transmission of a high protection segments (layer A) for mobile reception and low protection segments (Layer B) for fixed receivers

Service	**Segment settings**	**Segment settings**
	Layer A	Layer B
High protection segment	M-EIT (basic)	H-EIT (extended)
Low protection segment	Not used	H-EIT (Basic)

Table I.10 — Transmission of a high protection segment (layer A) for reception of one-seg and other with low protection (layer B) for mobile receivers

Service	**Segment settings**	**Segment settings**
	Layer A	**Layer B**
High protection segment	L-EIT (basic)	M-EIT (extended)
		H-EIT (extended)
Low protection segment	Not used	M-EIT (basic)
		H-EIT (extended)

Table I.11 — Transmission of a high protection segment (Layer A) for reception of one-seg, medium protection segments (Layer B) for mobile receivers and low protection segments (Layer C) for fixed reception

Service	**Segment settings**	**Segment settings**	**Segment settings**
	Layer A	**Layer B**	**Layer C**
High protection segment	L-EIT (basic)	H-EIT (extended)	H-EIT (extended)
Medium protection segment		H-EIT (basic)	H-EIT (extended)
Low protection segment	Not used	Not used	H-EIT (basic)

Annex J
(normative)

Stream type

The Table J.1 specifies the stream types used to determinate the different types of elementary streams transmitted in the PMT.

Table J.1 — Stream type

Value	Description
0x00	Undefined
0x01	Video according with ISO/IEC 11172
0x02	Video according with ITU-T Recommendation H262
0x03	Audio according with ISO/IEC 11172-3
0x04	Audio according with ISO/IEC 13818-3
0x05	Section
0x06	PES package
0x07	MHEG according with ISO/IEC 13522-5
0x08	According with ITU-T Recommendation H222.0 :2002, Annex 1
0x09	According with ITU-T Recommendation H.222.1
0x0A	According with ISO/IEC 13818-6 (type A)
0x0B	According with ISO/IEC 13818-6 (type B)
0x0C	According with SO/IEC 13818-6 (type C)
0x0D	According witnISO/IEC 13818-6 (type D)
0x0E	According with ITU-T Recommendation H222.0 auxiliary data
0x0F	Audio according with ISO/IEC 13818-7 (ADTS transport syntax)
0x10	Video according with ISO/IEC 14496-2
0x11	Audio according with ISO/IEC 14496-3
0x12	According with ISO/IEC 14496-1 SL – packet stream or Flexmux stream transported in the PES packets
0x13	According with ISO/IEC 14496-1 SL – packet stream or FlexMux stream conveyed in ISO/IEC 14496
0x14	Download syncronization protocol according ISO/IEC 13818-6
0x15	Metadata transported by a PES packet
0x16	Metadata transported by a metadata_sections
0x17	Metadata transported by the data carousel ISO/IEC 13818-6
0x18	Metadata transported by the object carousel
0x19	Metadata transported by a syncronized download protocol ISO/IEC 13818-6
0x1A	IPMP stream specified in 13818-11
0x1B	Video according with ITU-T Rec. H.264_ISO/IEC 14496-10
0x1C- 0x7D	Undefined
0x7E	Data pipe
0x7F	IPMP stream
0x80-0xFF	Private usage

Bibliography

[1] ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

[2] ARIB STD B23:2004, *Application execution engine platform for digital broadcasting*

[3] ARIB STD B25:2006, *Conditional access system specifications for digital broadcasting*