Digital Video Broadcasting (DVB); Metadata generation and deterministic DVB-T-mega-frame/DVB-T2-MI stream generation from MPEG-2 Transport Stream(s) for a DVB Single Illumination System

DVB Document A175

March 2018

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This technical specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about
60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardisation, interoperability and future proof specifications.

1 Scope

The present document describes the Single Illumination System, which allows to deliver Parent Signals for direct reception by consumer receivers and, at the same time, for a deterministic generation of daughter streams for terrestrial retransmission. Parent Signals can be provided to the daughter site via all defined TS-based DVB means – be it satellite, cable or terrestrial. Metadata may be provided as part of the Parent Signal(s) (called “in-band” in this document). Part of the metadata may also be provided “out-of-band”.

A typical use case is described by this document and is based on satellite broadcast signals on parent side – be it DVB-S (EN 300 421 [8]), -S2 (EN 302 307-1 [9]) or -S2X signals (EN 302 307-2 [10]) – that are addressing consumer DTH receivers and terrestrial transmitters on daughter side in parallel. For terrestrial retransmission of the whole or partial content, the DTH signal(s) of MPEG-2 TS format (ISO/IEC 13818-1 [5]) from one or several satellites can be used and services and service components can be selected according to terrestrial broadcasters’ needs. Also other, non-satellite Parent Signals and out-of-band metadata feeds can be integrated into the preparation of the terrestrial DVB-T2, EN 302 755 [2], and/or DVB-T, EN 300 744 [3], signals on daughter site. SFN operation of the terrestrial networks driven by the Single Illumination System is enabled.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 773: “Digital Video Broadcasting (DVB); Modulator Interface (T2-MI) for a second generation digital terrestrial television broadcasting system (DVB-T2)”.

[2] ETSI EN 302 755: “Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system”.

[3] ETSI EN 300 744: “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television”.

[4] ETSI TS 101 191: “Digital Video Broadcasting (DVB); DVB mega-frame for Single Frequency Network (SFN) synchronization”.

[5] ISO/IEC 13818-1: "Information technology - Generic coding of moving pictures and associated audio information: Systems".

[6] ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[7] ETSI TS 102 034: “Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks”.

[8] ETSI EN 300 421: “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services”.

[9] ETSI EN 302 307-1: “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2”.

[10] ETSI EN 302 307-2: “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)”.

[11] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[12] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, DOI 10.17487/RFC1952, May 1996, <https://www.rfc-editor.org/info/rfc1952>.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Control Stream Generator	A device that generates metadata – namely Framing and Timing information (F&TI), DSA Configuration Information (DSACI) and Layer 1 signalling – for the daughter site signal(s)
Daughter Site Adapter	A device that creates the TMF/T2-MI streams to be output to the related DVB-T or –T2 modulators – based on the metadata it receives from the CSG, the outputs streams consist of content received from the parent side
DSA Configuration Information	Information that enables deterministic generation of DVB-T2-MI streams and/or DVB-T-mega-frame streams at the output of the Daughter Site Adapter
F&TI packet	A TS packet used for F&TI signalling in the DVB-T case. This packet consists of the payload of the MIP packet – basis for forming a MIP packet on daughter site. An additional megaframe_timestamping function specific to SIS is part of the F&TI packet as well. The PID of the F&TI packet is allocated by the CSG and converted to the standard MIP PID using the DSACI pid_processing element.
FIRST_SYNCD	The SYNCD value as defined in EN 302 755 [2] for the first BBFRAME of the relevant Interleaving Frame

NOTE: FIRST_SYNCD shall be an integer multiple of 8.

FIRST_DFL	The DFL value as defined in EN 302 755 [2] for the first BBFRAME of the relevant Interleaving Frame

NOTE: FIRST_DFL shall be an integer multiple of 8.

In-band	Part of one of the Parent Signals
Interleaving Frame	Unit over which dynamic capacity allocation for a particular PLP is carried out, made up of an integer, dynamically varying number of FEC blocks and having a fixed relationship to the T2-frames

NOTE: The Interleaving frame may be mapped directly to one T2-frame or may be mapped to multiple T2-frames, it may contain one or more TI-blocks.

Layer 1 signalling	Metadata that describes the physical layer configuration. In the DVB-T2 case the L1 signalling provides the receiver with a means to access physical layer pipes within the T2-frames; for further details see EN 300 744 [3] for DVB-T and EN 302 755 [2] for DVB-T2

Layer 2 signalling	Equivalent to PSI/SI – see ISO/IEC 13818-1 [5] and EN 300 468 [6], respectively – since this document covers Transport Streams (ISO/IEC 13818-1 [5]) only, layer 2 metadata that describes the service configuration of the tuned and optionally of other multiplexes
Mode Adaptation	Applicable to DVB-T2 only: The Mode Adaptation modules, which operate separately on the contents of each PLP, slice the input data stream into data fields which, after stream adaptation, will form baseband frames (BBFRAMEs), see also EN 302 755 [2]
Out-of-band	Not part of any of the Parent Signals, but provided to the DSA by other means
Parent Network	Any DVB- and TS-based delivery network carrying the Parent Signal. This can be a satellite, cable or terrestrial network.
Parent Signal	Transport Stream present at the DSA input providing some or all the services used to build the output DTT stream.

NOTE: There may be one or several Parent Signal(s) contributing to the output DTT stream (T2-MI stream or stream of T-mega-frames). Each Parent Signal shall consist of a time reference of the SIS system clock (PCR_{abs}).

PCR_{abs}	A counter at the frequency of the SIS system clock, in the form of a PCR 48-bits field as defined in ISO/IEC 13818-1 [5].

NOTE: At the SIS epoch, $PCR_{abs} = 0$. It is not subject to leap seconds.

PCR_{FTSP}	The time (as sample of PCR_{abs}) corresponding to the first complete TS packet of the first BBFRAME of the relevant interleaving frame and PLP

NOTE: This defines the collection window for a given interleaving frame and PLP.

PCR_base	90 kHz clock
Primary SIS Service	The Primary SIS Service (one per terrestrial output/per current_DSA_group_id) consists of the listed mandatory elements and can consist of the listed optional elements: • PCR_{abs} using the transport packet adaptation mechanism (see ISO/IEC 13818-1 [5]) on a PID that is designated as the PCR_PID in the PMT (mandatory) • F&TI, which may be sent on the PCR_{abs} PID (mandatory) • DSACI (optional) • Terrestrial-only PSI/SI tables belonging to hybrid or sheer terrestrial services, including PATs (for terrestrial TSs, with PID ≠ 00_{16}) and PMTs (all optional)

Reference Transport Stream	A Reference Transport Stream (RTS) is a Transport Stream made up entirely of null packets (see ISO/IEC 13818-1 [5]) but with a defined timing in terms of the SIS reference clock. Each packet shall have an associated PCR_{abs} value.
Service Information	Service Information is the layer 2 signalling specific to DVB usage of Transport Streams and consists of the MPEG Programme Service Information and the DVB Service Information
SIS Service	The SIS Service (one per Parent Signal) consists of the listed mandatory elements and can consist of the listed optional elements: • PCR_{abs} using the transport packet adaptation mechanism (see ISO/IEC 13818-1 [5]) on a PID that is designated as the PCR_PID in the PMT (mandatory) • F&TI, which may be sent on the PCR_{abs} PID (optional) • DSACI (optional) • Terrestrial-only PSI/SI tables belonging to hybrid or sheer terrestrial services, including PATs (for terrestrial TSs, with PID ≠ 00_{16}) and PMTs (all optional)
SIS system clock	A 27 MHz clock that has an active edge at the SIS epoch and executes exactly 27,000,000 cycles in every subsequent second of UTC
SIS epoch	The epoch for DVB-T2 time, i.e. 2000-01-01 T 00:00:00 UTC
T2-frame	Fixed physical layer TDM frame that is further divided into variable size sub-slices

NOTE: The T2-frame starts with one P1 and one or multiple P2 symbols.

T2-MI stream	Stream of T2-MI packets carrying the T2 data for a single T2 profile and optionally any non-profile data with a given value of T2-MI stream ID

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$\lceil x \rceil$	Round towards plus infinity: The smallest integer that is not less than x
$\lfloor x \rfloor$	Round towards minus infinity: The largest integer that does not exceed x
DFL(i)	Data Field Length of BBFRAME i, determined on the basis of $N_{bytesIF}$ and N_{blocks} for i not equal to 0, this field shall be a multiple of 8

$DPCR_{IF}$	The duration (as difference of PCR_{FTSP}) of the frame_packet_count packets of the relevant interleaving frame and PLP, this duration is encoded as a 32 bit integer number of 27 MHz
$DPCR_{MF}$	Difference between the PCR_{abs} values for two subsequent DVB-T mega-frames
FIRST_DFL	The DFL value as defined in EN 302 755 [2] for the first BBFRAME of the relevant Interleaving Frame, this field shall be a multiple of 8
FIRST_ISCR(n)	The ISCR value of the first complete packet of the first BBFRAME of the Interleaving Frame for the PLP
frame_packet_count(n)	The number of TS packets that start in the Interleaving Frame for the PLP
ISCR(p)	Input Stream Clock Reference for TS packet p within the Interleaving Frame
n_{bat}	Number of Bouquet Association Tables transmitted
N_{blocks}	Number of FEC blocks in Interleaving Frame
$N_{bytesIF}$	The number of bytes of payload to be spread over all but the first BBFRAME of the Interleaving Frame, determined on the basis of $N_{payloadIF}$ and *FIRST_DFL*
N_{MF}	Number of TS packets in a DVB-T mega-frame
n_{other}	Number of other Transport Streams
$N_{payloadIF}$	Number of payload bits in the Interleaving Frame
$N_{payloadIF}(n)$	Number of payload bits in the Interleaving Frame n
N_{pkt}	Number of TS packets needed to packetize a whole table according to the packetization rules applicable to the sections of this table
$N_{steps\_to\_live}$	Number of processing loops not leading to the selection of the related TS packet for insertion into the output TS before discarding it
n_{stuff}	Number of stuffing TS packets should stuffing section content be needed (for continuity_counter wrapping)
offset	PCR_base count provided as part of DSACI
packet_arrival_time(n)	Arrival time of TS packet with index n at DSA
$PCR_{abs}(k)$	k^{th} instance of a PCR_{abs} value in parent TS
$PCR_{FTSP}(n)$	PCR_{abs} of the first TS packet of the Interleaving Frame n
$PCR_{MF}(n)$	PCR_{abs} value associated with the MIP packet for DVB-T mega-frame n
packet_period	Time interval between two TS packets of the same regenerated PSI/SI table
table_repetition_period	DSACI parameter indicating the repetition period of a PSI/SI table regenerated by the DSA

time_since_epoch	Time elapsed after SIS epoch (2000-01-01 T 00:00:00 UTC), counted in 90 kHz ticks
UPL	User Packet Length as defined in EN 302 755 [2]

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AVD	Audio/Video/Data
BAT	Bouquet Association Table
CAT	Conditional Access Table
CSG	Control Stream Generator
DSA	Daughter Site Adapter
DSACI	Daughter Site Adapter Configuration Information
DTH	Direct-To-Home (from satellite)
DTT	Digital Terrestrial Television
EIT	Event Information Table
F&TI	Framing & Timing Information
L1	Layer 1 (physical layer)
L2	Layer 2 (data link layer)
PAT	Program Association Table
PID	Packet Identifier
PCR	Programme Clock Reference
PLP_GROUP	Group of PLPs
PMT	Program Map Table
PSI/SI	MPEG Program-Specific Information/DVB Service Information
RTS	Reference Transport Stream
SDT	Service Description Table
TDT	Time Description Table
T-MF	DVB-T mega-frame
T2-MI	DVB-T2 Modulator Interface
TS	Transport Stream

4 DVB Single Illumination System definition

4.1 System overview and architecture

The target is, on the one hand, to feed receivers installed at consumer's households with (a) "Parent" Signal(s) suitable for direct reception and, on the other hand, to use the same signal(s) to derive services and service components for terrestrial re-transmission. The Parent Signal(s) might typically be uplinked – via DVB-S (EN 300 421 [8]), -S2 (EN 302 307-1 [9]) or –S2X (EN 302 307-2 [10]) – to (a) satellite(s). The terrestial multiplexes could either by DVB-T2 or DVB-T multiplexes and shall be suitable for use in SFN networks.

Central functional blocks on the parent site are the Control Stream Generator (CSG) and the Transport Stream multiplexer that prepares the TS (ISO/IEC 13818-1 [5]) that is transmitted to consumers. The CSG prepares all metadata that is not Layer 2 signalling (PSI/SI), i.e. the Layer 1 signalling, the Framing & Timing Information (F&TI) as well as the DSA Configuration Information (DSACI).

The CSG obtains instructions from its control interface (man-machine interface, ...) for the configuration of the DVB-T2 (EN 302 755 [2]) and/or DVB-T (EN 300 744 [3]) transmission frames that its counter-part on the daughter site, the Daughter Site Adapter (DSA), will eventually build. On that basis it generates the required metadata. On the daughter side the DSA receives Parent Signal(s) from one or more DVB networks and extracts services and service components for terrestrial re-transmission. That selection is determined by the DSA Configuration Information (DSACI). The latter also provides instructions to the DSA for the preparation of the terrestrial PSI/SI. In the T2 case, Layer 1 signalling is only decapsulated and inserted into the output T2-MI stream. Framing & Timing information indicates frame boundaries – in the T2 case the boundaries between interleaving frames and in the T case the boundaries between T mega-frames. The latter signals are output by the DSA to the related modulator(s).

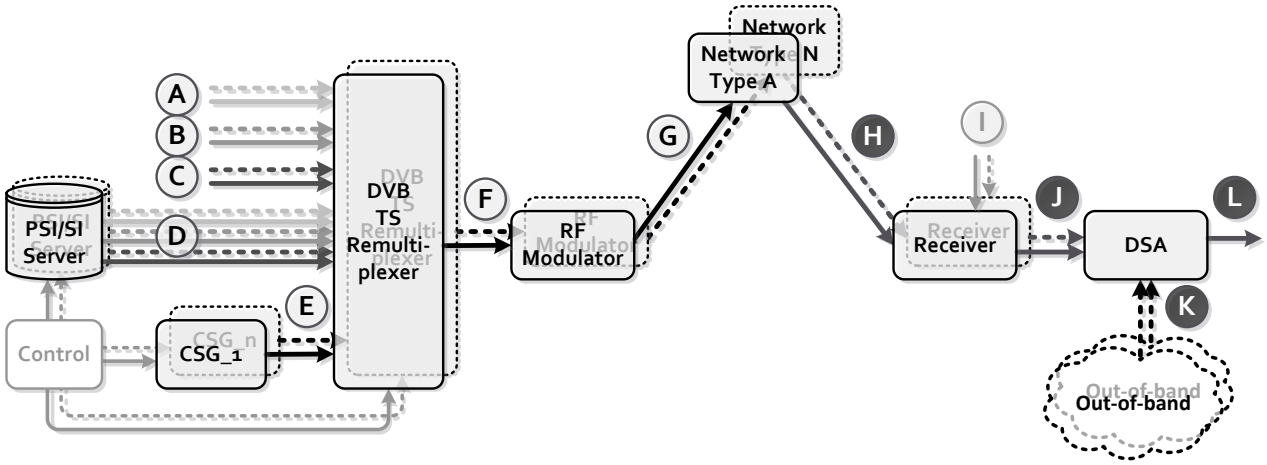

Figure 1: Conceptual block diagram of the Single Illumination transmission chain

Table 1: Conceptual interfaces within the Single Illumination chain

Inter-face	Signals at interfaces
A	TS packets containing specific DTT content
B	TS packets containing common Parent Network/DTT content
C	TS packets containing specific Parent Network content
D	TS packets containing PSI/SI sets for Parent Network, DTT and common Parent Network/DTT services
E	TS packets containing all metadata apart from PSI/SI
F	Parent TS
G	Parent TS modulated onto RF signal (uplink)
H	Parent TS modulated onto RF signal (downlink)
I	Control signal to receiver(s) for tuning, TS selection etc. This interface is not specified by the present document.
J	Parent TS (demodulated and descrambled (when needed if the service/component is sent free-to-air on the terrestrial network) downlink transport stream). These Parent Signals can be transported on any DVB-TS-based network (satellite/cable/terrestrial).
K	Out-of-band metadata: (Part of) DSACI. Out-of-band delivery of DSACI is not within the scope of this document.
L	T2-MI/T-MF stream (T2-MI stream encapsulated in TS packets in line with Data Piping described in TS 102 773 [1])

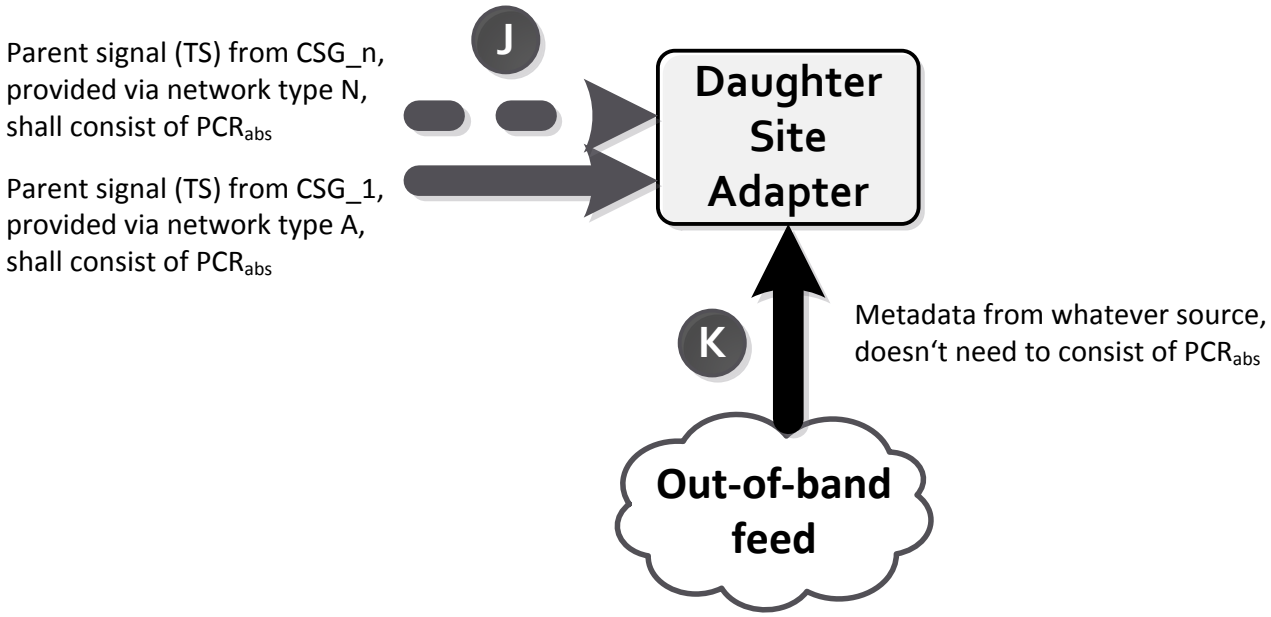

Figure 2: Inputs of the DSA

The present document describes the following:

- The content and structure of the “parent” Transport Stream(s) at interface H
- The processing that shall be carried out by the DSA in order to produce the “daughter” T2-MI or T-MF signals at interface L in a deterministic manner

The present document does not explicitly specify the processing carried out by the CSG, satellite multiplexer or other equipment at the parent site. This may be performed in any manner that results in (a) Transport Stream(s) at interface H that

- complies/comply with clause 5 of this specification, and
- after being processed as specified in clause 6, results in a T2-MI or T-MF stream at interface L that is compliant to TS 102 773 [1], EN 302 755 [2], EN 300 744 [3], TS 101 191 [4], ISO/IEC 13818-1 [5] and EN 300 468 [6]

4.2 Clock referencing throughout the DSA chain

The starting point for all the DSAs is the set of TSs received from the Parent Network(s). Every packet of every TS has an associated PCR$_{abs}$ value; this value is carried in ordinary PCRs in the special "SIS Service" and values for packets within the same parent TS not carrying such a PCR are interpolated according to a deterministic formula, see clause 6.3.1.1 (arrival timestamping).

The output TS of the TS multiplexer located inside the DSA, see figures 4 and 6 in clause 6.1, is based on a “Reference TS" generated (conceptually) by the DSA. This TS also has a PCR$_{abs}$ value associated with each packet; some of these (i.e. one per physical layer frame) are signalled in the F&TI packet and, again, a deterministic formula is specified for interpolation between them, see clause 6.4.2 (RTS generation).

The absolute PCRs in the input and output TSs run at the same rate - this has to be ensured by the CSG. Furthermore, they're all based on counters that conceptually started at the SIS epoch (2000-01-01T00:00:00 UTC).

The deterministic remultiplexing rules define how packets are placed in the output TS according to their arrival timestamps (the PCR_{abs} value of the incoming packet) and the PCR_{abs} values of the Reference TS into which they are inserted. All DSAs will make the same decision as to which input packet goes into which outgoing packet.

Once this is determined, the remaining step is to adjust any real PCRs contained in the services being remultiplexed. This is done by restamping the PCR according to the difference in absolute arrival and departure timestamps, see clause 6.4.5.2 (deterministic scheduling).

4.3 Mechanism to hide SIS-specific services away from direct reception by consumer receivers

In order not to confuse consumer (e.g. DTH) receivers with content or metadata transported as part of the Parent Signals but not addressing the aforementioned consumer receivers (i.e. targeting SIS DSAs only), such content and metadata is signalled with its related PSI/SI in a specific way, as described below. SIS-specific services and service components consist of the following:

- SIS Services are consisting of
    - PCR_{abs} (mandatory for all SIS Services)
    - F&TI (mandatory component of Primary SIS Service, optional for others)
    - DSACI (if carried in-band, it shall be provided via an SIS Service)
    - SIS-specific PSI/SI, e.g. terrestrial PSI/SI belonging to hybrid or sheer terrestrial services, including PATs (for terrestrial TSs, with PID $\neq 00_{16}$) and PMTs (all optional; if present, these tables shall be part of SIS Services)
- Services addressing DSAs only, e.g. sheer terrestrial services

SIS-specific settings of parameters in certain PSI/SI tables:

- SIS Services:
    - PAT of the Parent Signal: The SIS Service (one per Parent Signal) is listed in the PAT as all “visible” services that address consumer receivers directly.
    - PMT: stream_type 06_{16} (Rec. ITU-T H.222.0 | ISO/IEC 13818-1 PES packets containing private data), see ISO/IEC 13818-1 [5], is assigned to all SIS-specific metadata. A data_broadcast_id_descriptor is inserted consisting of a data_broadcast_id set to 101_{16} referring to this document and id_selector_byte settings depending on the metadata present:
        - F&TI: id_selector_byte = 01_{16}

        - DSACI: id_selector_byte = 02_{16}
        - Terrestrial PSI/SI tables belonging to a hybrid or sheer terrestrial service/service component: id_selector_byte = 03_{16}
    - SDT: The service descriptor indicates a service_type $0C_{16}$ (data broadcast service), see EN 300 468 [6], for the SIS Service (one per Parent Signal).

- Services addressing DSAs only, e.g. sheer terrestrial services:
    - PAT of the Parent Signal: Services addressing DSAs only are listed in the PAT as all “visible” services that address consumer receivers directly.
    - SDT: The service descriptor indicates a service_type $0C_{16}$ (data broadcast service), see EN 300 468 [6], for the hidden services addressing DSAs only.

5 Specification of Parent Signals

5.1 Overview

The Parent Signals at interface H shall comprise one or more DVB Transport Streams compliant to ISO/IEC 13818-1 [5] and suitable for direct reception by DVB-compliant consumer receivers. Those Transport Streams contain DVB services as defined in EN 300 468 [6] and SIS metadata for generation of the T2-MI or T mega-frame stream at the output of the DSA, i.e. Framing & Timing Information as well as DSA Configuration Information for the generation of the DTT multiplex.

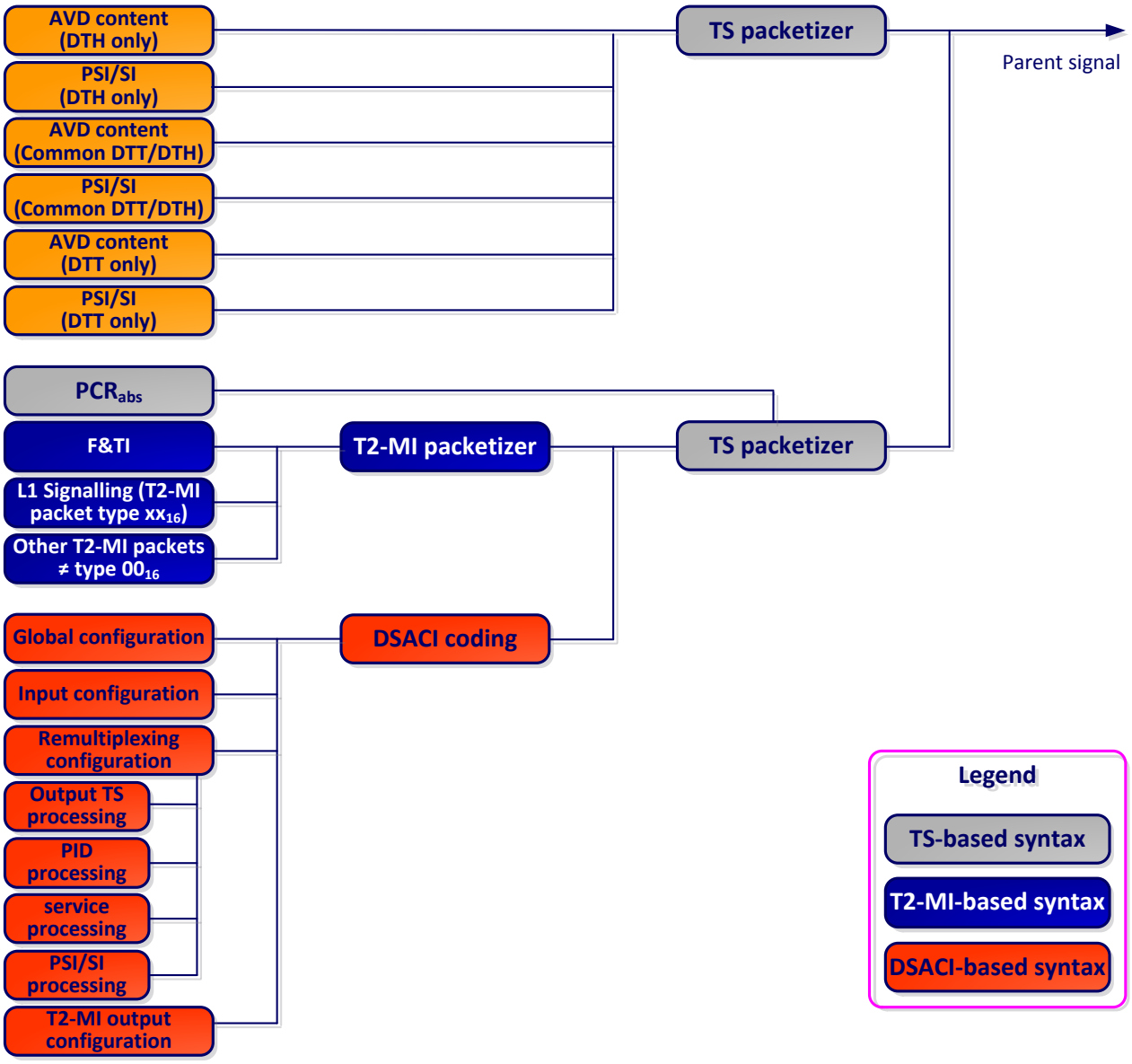

Figure 3: Parent Signal composition illustrating metadata and Audio/Video/Data content, DVB-T2 case

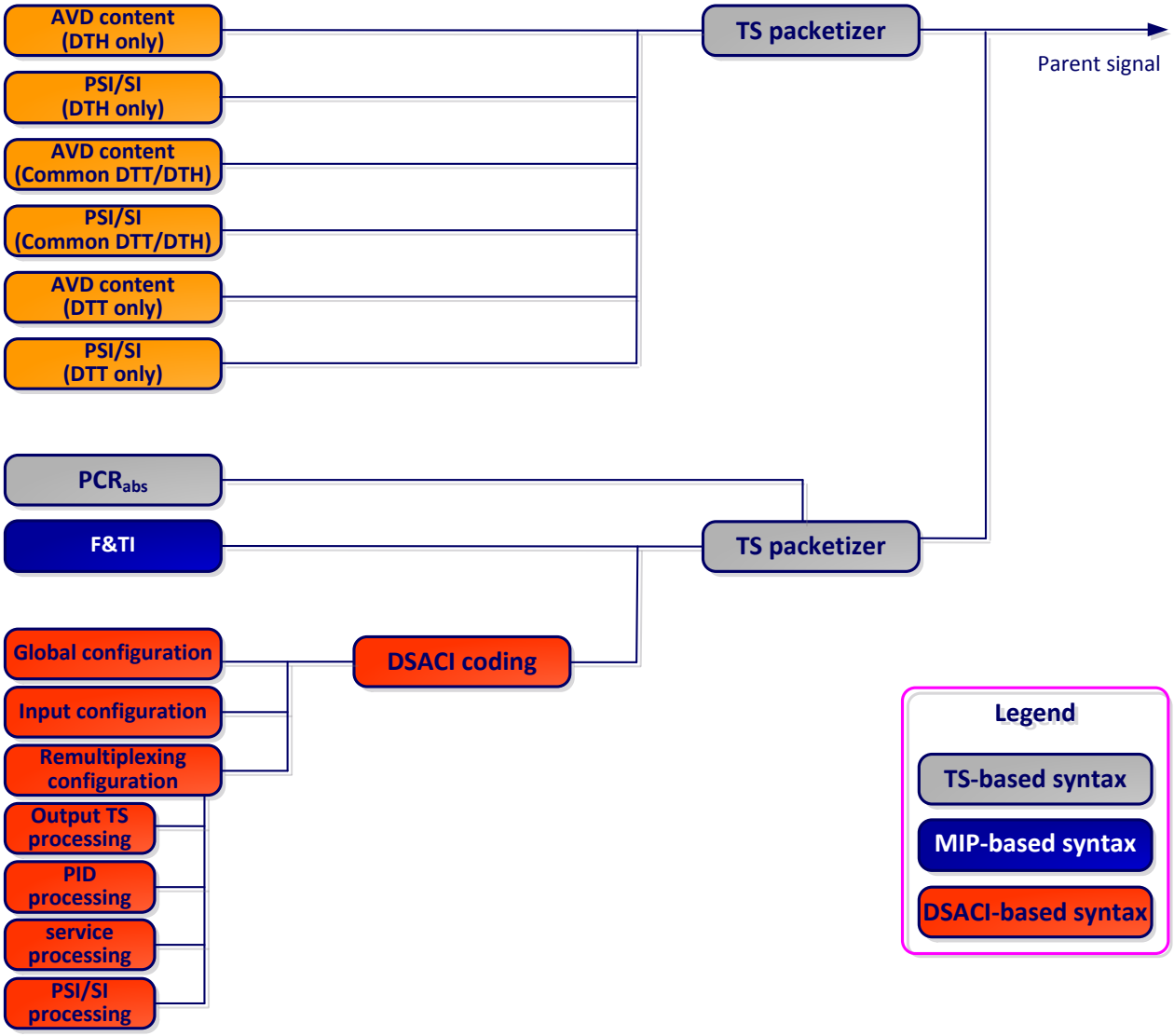

Figure 4: Parent Signal composition illustrating metadata and Audio/Video/Data content, DVB-T case

Up to three types of services may be carried:

- Services addressing only direct reception of the parent TS, while they are discarded by the daughter sites (e.g. DTH-only services)
- Common services, addressing both, direct reception of the parent TS and further transmission on daughter site (e.g. DTH and DTT)
- Services addressing only further transmission from daughter site (e.g. DTT-only services, transported via satellite in a hidden way)

Each of the Parent Signals shall also include an absolute time reference of the SIS system clock through the following component:

- A component of the SIS Service carrying PCR_{abs} in the adaptation field

For DVB-T retransmission at daughter sites, the Parent Signal carrying the Primary SIS Service shall also include the information for the construction of DVB-T mega-frames. Should that information be part of other SIS Services, it would be ignored.

- DVB-T Reference Stream mega-frame timestamps in F&TI packet form

For DVB-T2 retransmission at daughter sites, the Parent Signals carrying the Primary SIS Service shall also include the following components for the construction of T2-MI:

- L1 signalling applicable to the daughter streams at interface L (see clause 5.4.1)
- Framing & Timing Information for generation of Baseband Frames (see clause 5.6)
- As applicable, other T2-MI packets to be inserted in the T2-MI stream

For DVB-T2 retransmission from daughter sites, the Parent Signals may also include the following components for the construction of T2-MI:

- Signalling information for generation of T2-MI over Transport Stream

The Parent Signals may also include the signalling configuration for deterministic re-multiplexing containing the following components, see DSACI described in clause 5.7.

5.2 Service Information (PSI/SI)

PSI/SI signalling as defined in ISO/IEC 13818-1 [5] and EN 300 468 [6] shall be part of the Parent Signals for the types of services present in the multiplexes addressing the Daughter Site Adapter:

- Parent Network-only services
- Common services addressing Parent Network receivers and terrestrial transmitters:
  The Parent Signal shall also include supplementary PSI/SI for DTT and DSACI Configuration Information, which in combination with the PSI/SI for the Parent Network enables the DSA to generate PSI/SI for the same services when re-transmitted terrestrially. The supplementary DTT PSI/SI is sent compliant to ISO/IEC 13818-1 [5] and EN 300 468 [6] as described in clause 5.8.

  This document describes different approaches for the generation of DTT PSI/SI, see clause 6.4.4. These appraoches will require the following information as part of the Parent Signal:

  - Tables whose provision is stopped:
    The Parent Signal shall contain indication for stopping the provision of CATs as part of DSACI.
  - Tables passed through with no change:
    The Parent Signal shall contain DSACI signalling for PID remapping of the Parent Network PSI/SI table (PID remapping with same PID on Parent Network and DTT may be used).

- Tables which are modified using section patching:
  The Parent Signal shall include Parent Network PSI/SI sections and DSACI signalling for the patching modification.

- Tables that are regenerated by the DSA:
  The Parent Signal shall include Parent Network PSI/SI sections and DSACI signalling for the table regeneration and multiplexing.

- DTT-only services, for which PSI/SI is carried as for all other services – with these exceptions:

  - SDT: DTT-only services shall be declared in the Service Description Table as data services, i.e. characterized by service_type $0C_{16}$ (data broadcast service).

Note: Reference time might be obtained from TDT (UTC there), which is present mandatorily.

5.3 SIS Services carrying PCR_{abs}

Each Parent Signal shall include a generated SIS Service. This service carries PCR_{abs} using the transport packet adaptation mechanism (see ISO/IEC 13818-1 [5]) on a PID that is designated as the PCR_PID in the PMT.

The Primary SIS Service (see section 5.7.3.2.1) shall consist of the F&TI component, while in other SIS Services from other Parent Signals the F&TI component is optional. In a single DSACI, a unique SIS Service is flagged as primary, hence the F&TI component of other SIS Services may be ignored. In the case that the Parent Signal includes the F&TI component, this F&TI component may be sent on the PCR_{abs}_PID.

A CSG shall be able to generate a DSACI file for configuration of re-transmitted services in the terrestrial modulation. That DSACI file shall – if provided – be part of the Primary SIS Service. Other SIS Services may include DSACI for other terrestrial multiplexes.

Any SIS Service may also consist of terrestrial PSI/SI for sheer terrestrial and hybrid services.

5.4 Layer 1 signalling

5.4.1 DVB-T2 case

L1 signalling consists of T2-MI packets of type 10_{16} (L1-Current) and, if applicable, type 11_{16} (L1-Future), as defined in TS 102 773 [1]. T2-MI packets of type 10_{16} are used for DSA processing. T2-MI packets of type 11_{16} are simply piped through the DSA. See sub-clause 6.6 for details.

5.4.2 DVB-T case

L1 signalling for DVB-T shall be carried in an F&TI packet – based on the MIP structure (see TS 101 191 [4]).

5.5 Other T2-MI packet types (DVB-T2 case)

For DVB-T2 use cases, a single CSG prepares all T2-MI packet types - except for the Baseband Frame type 00_{16}. T2-MI packets of type BBF are created by the DSA and are inserted by the T2-MI multiplexer on daughter site. The same CSG shall also provide T2-MI packets of type $F0_{16}$ carrying Framing & Timing Information (see section 5.6). The latter packets shall not be part of the T2-MI stream output at interface L, they are dedicated to DSA processing only.

5.6 Framing & Timing Information (F&TI)

5.6.1 Overview

The Framing & Timing Information provides metadata required by the DSA for building – in the DVB-T2 case – BBFRAMEs, Interleaving Frames and T2-frames. In the DVB-T case mega-frames are built. The details of this deterministic process are specified in clause 6.

5.6.2 DVB-T2 case

F&TI for DVB-T2 shall be carried in a T2-MI packet (see sub-clause 5.8 below) of type $F0_{16}$. The T2-MI packet shall be coded as shown in table 2 below. One such packet shall be sent for each T2-frame – prior to the provision of the data and metadata for that T2-frame.

Table 2: Syntax of Framing & Timing Information (F&TI), DVB-T2 case

Syntax	# of bits	Format
packet_type	*8*	*Uimsbf*
packet_count	*8*	*Uimsbf*
superframe_idx	*4*	*Uimsbf*
rfu	*9*	*Void*
t2mi_stream_id	*3*	*Uimsbf*
payload_len	*16*	*Uimsbf*
frame_idx	8	Uimsbf
for (u=0; u < number_of_PLPs: u++) {		
plp_id	8	Uimsbf

intl_frame_start & rfu	1 + 7	Uimsbf
MATYPE1	8	Uimsbf
MATYPE2	8	Uimsbf
TTO_E	5	Uimsbf
TTO_M	7	Uimsbf
TTO_L	8	Uimsbf
FIRST_ISCR	22	Uimsbf
BUFS	10 + 2	Uimsbf
FIRST_DFL	16	Uimsbf
FIRST_SYNCD	16	Uismbf
NEXT_FIRST_SYNCD	16	Uismbf
MODE	8	Uimsbf
PCR_{FTSP}	48	Uimsbf
Frame_Pkt_Count	32	Uimsbf
$DPCR_{IF}$	32	Uimsbf
}		
pad	0 … 7	Void
CRC32	32	Uimsbf

NOTE: The parameters written in *Italic characters* at the top and the bottom of table 2 above reflect the mandatory parameters being part of each T2-MI packet, see TS 102 773 [1].

packet_type shall be equal to $F0_{16}$

packet_count shall be as defined in TS 102 773 [1]

superframe_idx: Indicates the superframe to which this T2-MI packet of type “F&TI” refers.

t2mi_stream_id: As defined in TS 102 773 [1]

payload_len: As defined in TS 102 773 [1]

frame_idx: the FRAME_IDX (as defined in EN 302 755 [2]) within the superframe of the T2-Frame to which the Interleaving Frame being described is mapped. Where multi-frame interleaving or Frame-skipping are used for a PLP , the signalling relevant to a given Interleaving Frame shall occur for the frame_idx corresponding to the first T2-Frame to which the Interleaving Frame is mapped, and shall be repeated for subsequent values of frame_idx until the first T2-Frame to which the next Interleaving Frame is mapped.

number_of_PLPs: Number of PLPs belonging to the current Interleaving Frame. This value is derived from the setting of the parameter "payload_len" above.

plp_id: The plp_id as defined in TS 102 773 [1] to which the following fields apply

intl_frame_start: This shall be set to 1

MATYPE1, MATYPE2: The values of the corresponding fields of the BBHEADER for the relevant PLP

TTO_E: The TTO_E value for the first BBFRAME of the relevant Interleaving Frame for the relevant PLP

TTO_M: The TTO_M value for the first BBFRAME of the relevant Interleaving Frame for the relevant PLP

TTO_L: The TTO_L value for the first BBFRAME of the relevant Interleaving Frame for the relevant PLP. If $ISCR_{short}$ is used, this field shall have the value 0.

FIRST_ISCR: The ISCR value of the first complete packet of the first BBFRAME of the Interleaving Frame for the PLP

BUFS: The BUFS value for the PLP, as defined in EN 302 755 [2]

FIRST_DFL: The DFL value of the first BBFRAME of the Interleaving Frame for the PLP, as defined in EN 302 755 [2]. This field shall be a multiple of 8.

FIRST_SYNCD: The SYNCD value of the first BBFRAME of the Interleaving Frame for the PLP, as defined in EN 302 755 [2]. This field shall be a multiple of 8. The value of FIRST_DFL shall be set such that the PADDING field of the BBFRAME is large enough to allow for insertion of In-band Signalling, if used.

NEXT_FIRST_SYNCD: The FIRST_SYNCD value of the next F&T packet (SYNCD value of the first BBFRAME of next Interleaving Frame for the PLP).

NOTE: Let n the index of the relevant interleaving frame

$$NEXT\_FIRST\_SYNCD(n) = FIRST\_SYNCD(n+1)$$

MODE: The MODE of the PLP (00=Normal Mode, 01=High Efficiency Mode, other values reserved)

PCR_FTSP: The time (as sample of PCR_{abs}) corresponding to the first complete TS packet of the first BBFRAME of the relevant interleaving frame and PLP

NOTE: This defines the collection window for a given interleaving frame and PLP

frame_packet_count: The number of TS packets that start in the Interleaving Frame for the PLP.

$DPCR_{IF}$: The duration (as difference of PCR_{FTSP}) of the frame_packet_count packets of the relevant interleaving frame and PLP. This duration is encoded as a 32 bit integer number of 27 MHz.

NOTE: Let n the index of the relevant interleaving frame

$$DPCR_{IF} = PCR_{FTSP}(n+1) - PCR_{FTSP}(n)$$

pad: Padding, between 0 and 7 bits, all set to '0', for filling up to an integer number of Bytes.

CRC32: CRC as defined in EN 302 755 [2]

5.6.3 DVB-T case

Framing & Timing Information (F&TI) for DVB-T shall be carried in a MIP packet (see TS 101 191 [4]), to which the megaframe_timestamping function is added. The transport_packet_header has its PID defined by DSACI (see section 5.7), i.e. the default PID for MIPs is not used. This way an F&TI packet is formed by the CSG.

This packet following the MIP payload syntax shall carry a megaframe_timestamping function that belongs to a function loop broadcast to all transmitters (0000_{16} is used as tx_identifier). On daughter site, the DSA shall remove the function when building actual MIP packets for the DVB-T transmitter (see section 6.5.2).

Exactly one such packet shall be sent for each T mega-frame.

This specification defines the megaframe_timestamping function as follows:

Table 3: Syntax of the megaframe_timestamping function, DVB-T case

Syntax	Number of bits	Identifier
megaframe_timestamping_function() {		
function_tag	8	uimsbf
function_length	8	uimsbf
PCR_ABS_base	33	uimsbf
reserved	6	undefined
PCR_ABS_extention	9	uimsbf
}		

function_tag: Identifier for the megaframe_timestamping function ($F0_{16}$).

function_length: Indicates the total length of the function field (see TS 101 191 [4]) in bytes (8 bytes).

PCR_ABS_base: PCR_base value that represents the absolute timestamp of the first packet of the subsequent mega-frame.

PCR_ABS_extension: PCR_extension value that represents the absolute timestamp of the first packet of the succeeding mega-frame

5.7 DSA Configuration Information (DSACI)

5.7.1 Overview

The definition of DSA Configuration Information is applicable to both terrestrial transmission systems covered by this document, i.e. DVB-T2 (EN 302 755 [2]) and DVB-T (EN 300 744 [3]).

The DSA is configured on the basis of the DSA Configuration Information (DSACI) provided by the CSG. DSACI is formatted as an XML file, which provides all parameter settings required. Below all elements belonging to DSACI are outlined, namely the global configuration, the input configuration, the re-multiplexing and the output configuration.

This XML file may be provided to the DSA either in-band or out-of-band, see clause 6.2. In the latter case the transfer of the file to the DSA is user-defined and not covered by this specification. For in-band provision a carousel approach based on private sections is used, see clauses 5.7.5 and 6.2.1.

5.7.2 DSACI Structure

5.7.2.1 XML schema representation

Table 4 below outlines the top level XML schema structure of the DSA Configuration Information.

Table 4: Structure of the DSACI XML schema

type_global_configuration
type_input_configuration
type_remultiplexing
type_output_processing

5.7.3 XML types and XML elements of the DSA Configuration Information

5.7.3.1 Global configuration

5.7.3.1.1 Overview

The XML type_global_configuration consists of the XML elements current_DSA_group_id, global_version_number, global_application_time and SIS_edition.

5.7.3.1.2 XML schema representation

Table 5 below outlines the XML schema structure of the aforementioned parameters:

Table 5: XML schema syntax of the type_global_configuration

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

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="DSACI">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="global_configuration" type="type_global_configuration" />
            <xs:element name="input_configuration" type="type_input_configuration" />
            <xs:element name="remultiplexing" type="type_remultiplexing"/>
            <xs:element name="output_processing" type="type_output_processing"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>

<!-- type_global_configuration -->
   <xs:complexType name="type_global_configuration">
      <xs:sequence>
         <xs:element name="current_DSA_group_id" type="xs:int"/>
         <xs:element name="global_version_number" type="type_version_number"/>
         <xs:element name="global_application_time" type="xs:int"/>
         <xs:element name="SIS_edition" type="type_SIS_edition"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_SIS_edition-->
   <xs:complexType name="type_SIS_edition">
      <xs:sequence>
         <xs:element name="major" type="xs:int"/>
         <xs:element name="middle" type="xs:int"/>
         <xs:element name="minor" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.1.3 Definition of XML elements and XML types

current_DSA_group_id: Indicates the group of DSAs this DSACI parameter set, i.e. the contents of the entire DSACI structure outlined in clause 5.7.2, applies to.

global_version_number: Indicates the version of the entire current DSACI set. Any change to any of its parameters results in an incrementation by 1 of this global version number.

global_application_time: Time instance, from which the version of this DSACI set indicated by global_version_number becomes applicable. Indication is given as number of 90 kHz ticks since SIS epoch.

SIS_edition: Indicates the edition of the SIS standard the coding is based on in a three-digit form that corresponds to the ETSI versioning scheme, i.e. V1.1.1 for this edition of this document. SIS_edition represents the following XML elements:

Major: Major digit of the ETSI version number – "1" for this edition of this document.

Middle: Central digit of the ETSI version number – "1" for this edition of this document.

Minor: Last digit of the ETSI version number – "1" for this edition of this document.

5.7.3.2 Input configuration

5.7.3.2.1 Overview

The XML type_input_configuration identifies the Transport Streams at the input stage of the DSA services and service components are to be extracted from for daughter site re-transmission. This XML type also assigns a replacement identifier – source_id – to the double input_TS_id/input_ON_id. Furthermore it identifies the PID of the PMT belonging to the SIS Service and indicates the single Primary SIS Service.

5.7.3.2.2 XML schema representation

Tables 6 below outlines the XML schema structure of the type_input_configuration:

Table 6: XML schema syntax of the type_input_configuration

```
<!-- type_input_configuration-->
   <xs:complexType name="type_input_configuration">
      <xs:sequence>
         <xs:element maxOccurs="unbounded" minOccurs="1" name="input" type="type_input_parent" />
      </xs:sequence>
   </xs:complexType>

<!-- type_input_parent: number of instances equals number of input TSs defined -->
   <xs:complexType name="type_input_parent">
      <xs:sequence>
         <xs:element name="input_TS_id" type="xs:int"/>
         <xs:element name="input_ON_id" type="xs:int"/>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="PMT_PID_SIS_service" type="type_pid"/>
         <xs:element name="Primary_SIS_Service_Flag" type="xs:boolean"/>
      </xs:sequence>
   </xs:complexType>

   <xs:simpleType name="type_pid">
      <xs:restriction base="xs:int">
         <xs:minInclusive value="0"/>
         <xs:maxInclusive value="8191"/>
      </xs:restriction>
   </xs:simpleType>
```

5.7.3.2.3 Definition of XML elements and XML types

input_TS_id: Identifier of an input TS to be selected for DSA processing.

input_ON_id: Identifier of the Original Network the aforementioned input TS belongs to. This parameter together with the input_TS_id are uniquely identifying the related Transport Stream.

source_id: An identifier that replaces the double input_TS_id/input_ON_id for further use in other XML types and XML elements.

PMT_PID_SIS_Service: Packet Identifier of the PMT belonging to the SIS Service being part of the Transport Stream identified by the parameters above. In the case that no SIS Service belongs to the TS making up the current instance of the loop, this parameter shall be set to '0'.

Primary_SIS_Service_Flag: In the case that an SIS Service belongs to the identified TS, this identifier indicates, if this is the Primary SIS Service or not, as follows:

false: Not the Primary SIS Service

true: The Primary SIS Service

5.7.3.3 Remultiplexing

5.7.3.3.1 Overview

The XML type_remultiplexing assigns output Transport Streams to Physical Layer Pipes – in the case of DVB-T2. In the DVB-T case the PLP_ID has no meaning and shall be ignored. Each PLP/TS assignment also consists of the elements pid_processing, service_pmt_processing and psisi_processing. The latter element, in turn, consists of the elements <pat>, <cat>, <sdt_bat> and <eit>.

5.7.3.3.1.1 XML schema representation

Table 7 below outlines the XML schema structure of the <remultiplexing> section.

Table 7: XML schema syntax of the type_remultiplexing

```
<!-- type_remultiplexing-->
   <xs:complexType name="type_remultiplexing">
      <xs:sequence>

         <xs:element name="output_TS" type= "type_output_TS" minOccurs="1"
maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_output_TS: number of instances equals number of output TSs defined -->
   <xs:complexType name="type_output_TS">
      <xs:sequence>
         <xs:element name="PLP_id" minOccurs="0" maxOccurs="1" type="xs:int"/>
         <xs:element name="output_TS_id" type="xs:int"/>
         <xs:element name="output_ON_id" type="xs:int"/>
         <xs:element name="pid_processing" type= "type_pid_processing"/>
         <xs:element name="service_pmt_processing" type="type_service_pmt_processing"/>
         <xs:element name="psisi_processing" type="type_psisi_processing"/>
         <xs:element name="Nsteps_to_live" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.3.1.2 Definition of XML elements and XML types

PLP_id: Identifies the PLP carrying the Transport Stream identified with the parameters output_TS_id, output_ON_id and output_network_id, coding as defined in EN 302 755 [2].

output_TS_id: Identifies the output Transport Stream the loop instance is covering.

output_ON_id: Identifies the Original Network the aforementioned output TS stems from. Together with the output_TS_id it identifies the TS uniquely.

pid_processing: XML element described in sub-clause 5.7.3.3.2 of this document.

service_pmt_processing: XML element described in sub-clause 5.7.3.3.3 of this document.

psisi_processing: XML element described in sub-clause 5.7.3.3.4 of this document.

Nsteps_to_live: Number of processing loops not leading to the selection of the related TS packet for insertion into the output TS before discarding it.

5.7.3.4 PID processing

5.7.3.4.1 Overview

The XML type_pid_processing replaces Packet Identifiers at the input of the DSA by PIDs as they shall appear at the output of the DSA. Packet identification being part of subsequent DSACI XML types is in almost all cases based on the element output_PID.

5.7.3.4.2 XML representation

Table 8 below outlines the XML schema structure of the type_pid_processing:

Table 8: XML schema syntax of the type_pid_processing

```
<!-- type_pid_processing -->
   <xs:complexType name="type_pid_processing">
      <xs:sequence>
         <xs:element name="pid" type="type_pid_proc" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_pid_proc: number of instances equals number of defined output_PIDs -->
   <xs:complexType name="type_pid_proc">
      <xs:sequence>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="input_PID" type="type_pid"/>
         <xs:element name="output_PID" type="type_pid"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.4.3 Definition of XML elements and XML types

source_id: Identifies the related Transport Stream uniquely and replaces the TS_Id/ON_Id double.

input_PID: Packet Identifier of the related component within the input TS identified by the source_id.

output_PID: Packet Identifier of the same component as the one identified with the input_PID abobe within the related output TS, replaces the aforementioned input_PID.

5.7.3.5 Service and PMT processing

5.7.3.5.1 Overview

The XML type_service_pmt_processing maps service identifiers, service names and provider names from the related input settings to the corresponding output settings – for all services belonging to a single output TS. It also defines how PMTs are to be processed and identifies the still applicable CA Systems for each service component. Note that descrambling for free-to-air terrestrial provision of service components takes place in the receivers being part of the DSA machinery.

5.7.3.5.2 XML schema representation

Table 9 below outlines the XML schema structure of the type_service_pmt_processing:

Table 9: XML schema syntax of the type_service_pmt_processing

```
<!-- type_service_pmt_processing -->
   <xs:complexType  name="type_service_pmt_processing">
      <xs:sequence>
         <xs:element name="service" type="type_service" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_service: number of instances equals number of defined services in an output TS-->
   <xs:complexType name="type_service">
      <xs:sequence>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="input_service_id" type="xs:int"/>
         <xs:element name="output_service_id" type="xs:int"/>
         <xs:element name="output_service_name" type="xs:string"/>
         <xs:element name="output_provider_name" type="xs:string"/>
         <xs:element name="eit_schedule_flag" type="xs:boolean"/>
         <xs:element name="eit_present_following_flag" type="xs:boolean"/>
         <xs:element name="running_status" type="xs:int"/>
         <xs:element name="free_CA_mode" type="xs:int"/>
         <xs:element name="output_PMT_PID" type="type_pid"/>
         <xs:element name="pmt_processing" type="type_pmt_processing"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_pmt_processing_mode -->
   <xs:complexType name="type_pmt_processing_mode">
      <xs:choice>
         <xs:element name="pmt_passthrough" type="pmt_passthrough_processing"/>
         <xs:element name="pmt_patching" type="pmt_patching_processing"/>
         <xs:element name="pmt_regeneration" type="pmt_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>

<!-- pmt_passthrough_processing -->
   <xs:simpleType name="pmt_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- pmt_patching_processing -->
   <xs:complexType name="pmt_patching_processing">
      <xs:sequence>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="ECM" type="type_ECM"/>
      </xs:sequence>
   </xs:complexType>

<!-- pmt_regeneration_processing -->
   <xs:complexType name="pmt_regeneration_processing">
      <xs:sequence>
         <xs:element name="table_repetition_period" type="xs:int"/>
         <xs:element name="offset" type="xs:int"/>
         <xs:element name="PCR_PID" type="type_pid"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="output_pid" type="type_pid"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="ECM" type="type_ECM"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_ECM -->
   <xs:complexType name="type_ECM">
      <xs:sequence>
         <xs:element name="CAS_id" type="xs:int"/>
         <xs:element name="output_ECM_PID" type="type_pid"/>
      </xs:sequence>
```

```
</xs:complexType>
```

5.7.3.5.2.1 Definition of XML elements and XML types

source_id: Identifies the source of the input Transport Stream the service to be selected stems from.

input_service_id: Service identifier of a selected input service.

output_service_id: Service identifier of the selected service mentioned above at the DSA output, replaces the aforementioned input_service id.

output_service_name: Service name of the aforementioned service at the DSA output.

output_provider_name: Provider name of the aforementioned service at the DSA output.

eit_schedule_flag: Sets the EIT_schedule_flag in the SDT, see EN 300 468 [6].

eit_present_following_flag: Sets the EIT_present_following_flag in the SDT, see EN 300 468 [6].

running_status: Sets the running_status field in the SDT and the EIT (if present), see EN 300 468 [6].

free_CA_mode: Sets the free_CA_mode flag in the SDT and EIT (if present), see EN 300 468 [6].

output_PMT_PID: PID of the PMT at the output.

type_pmt_processing_mode: Indicates the mode of the PMT processing as follows:

PMT_passthrough_processing: The incoming table section is unchanged. Only PID remapping may be applied if required by the pid_processing XML element being part of the XML element remultiplexing.

pmt_patching_processing: The relevant attributes as described in table 17 shall be modified according to values provided by other DSACI elements. Additionally, CAS-related information is updated according to the retained CAS_id's, i.e. part of the CAS_id's might be replaced by stuffing descriptors, because the terrestrially retransmitted service component consists of fewer CASs than the one received from the parent site.

pmt_regeneration_processing: The PMTs to be output at DSA's interface L are created based on DSA Control Information received from parent site.

table_repetition_period: Indicates the repetition period of the PMT identified by the output_PMT_PID as the number of 90 kHz ticks.

offset: Defines an offset relative to the beginning of the insertion window for the insertion of the first related table within this insertion window. See clause 6.4.4.4.1 for the insertion window mechanism.

PCR_PID: Indicates the Packet Identifier of the Programme Clock Reference.

CAS_id: Indicates a Conditional Access System identifier to be retained.

output_ECM_PID: Packet Identifier of the Entitlement Control Messages at the DSA output.

5.7.3.6 PSI/SI processing

5.7.3.6.1 Overview

The XML type_psisi_processing consists of the XML elements pat, cat, sdt_bat and eit.

5.7.3.6.2 XML schema representation

Table 10 below outlines the XML schema structure of the type_psisi_processing:

Table 10: XML schema syntax of the type_psisi_processing

```
<!-- type_psisi_processing -->
   <xs:complexType name="type_psisi_processing">
      <xs:sequence>
         <xs:element name="pat" type="type_pat_processing"/>
         <xs:element name="cat" type="type_cat_processing"/>
         <xs:element name="sdt_bat" type="type_sdt_bat_processing"/>
         <xs:element name="eit" type="type_eit_processing"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.6.3 Definition of XML elements and XML types

pat: XML element described in clause 5.7.3.7 of this document.

cat: XML element described in clause 5.7.3.8 of this document.

sdt_bat: XML element described in clause 5.7.3.9 of this document.

eit: XML element described in clause 5.7.3.10 of this document.

5.7.3.7 PAT processing

5.7.3.7.1 Overview

The XML type_pat_processing_mode covers the processing of Program Association Tables – see ISO/IEC 13818-1 [5] – and indicates the repetition period of those, their time offset to the beginning of the insertion window – see clause 6.4.4.4.1 for the insertion window mechanism – as well as the version number of regenerated PATs.

5.7.3.7.2 XML schema representation

Table 11 below outlines the XML structure of the type_pat_processing_mode:

Table 11: XML schema syntax of the type_pat_processing_mode

```
<!-- type_pat_processing_mode -->
   <xs:complexType name="type_pat_processing_mode">
      <xs:choice>
         <xs:element name="pat_passthrough" type="pat_passthrough_processing"/>
         <xs:element name="pat_patching" type="pat_patching_processing"/>
         <xs:element name="pat_regeneration" type="pat_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>
```

```
<!-- pat_passthrough_processing -->
   <xs:simpleType name="pat_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- pat_patching_processing -->
   <xs:simpleType name="pat_patching_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- pat_regeneration_processing -->
   <xs:complexType name="pat_regeneration_processing">
      <xs:sequence>
         <xs:element name="table_repetition_period" type="xs:int"/>
         <xs:element name="offset" type="xs:int"/>
         <xs:element name="PAT_version_number" type="type_version_number"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.7.3 Definition of XML elements and XML types

type_pat_processing_mode: Indicates the mode of the PAT processing as follows:

pat_passthrough_processing: The incoming table is unchanged. Only PID remapping may be applied if required by the pid_processing sub-section.

pat_patching_processing: The relevant attributes as described in table 17 shall be modified according to values provided by other DSACI elements.

pat_regeneration_processing: The PATs to be output at DSA's interface L are created based on DSACI.

table_repetition_period: Indicates the repetition period of the PATs as the number of 90 kHz ticks.

offset: Defines an offset relative to the beginning of the insertion window for the insertion of the first related table within this insertion window. See clause 6.4.4.4.1 for the insertion window mechanism.

PAT_version_number: Indicates the version_number of the PAT to be created, see ISO/IEC 13818-1 [5].

5.7.3.8 CAT processing

5.7.3.8.1 Overview

The XML type_cat_processing_mode covers the processing of Conditional Access Tables – see ISO/IEC 13818-1 [5] – and indicates in the case of patching the CAS identifiers as well as the EMMs to be retained, and in the case of regeneration this XML type assigns the repetition period of those, their time offset to the beginning of the insertion window – see clause 6.4.4.4.1 for the insertion window mechanism – as well as the version number of the CATs to be regenerated and output.

5.7.3.8.2 XML schema representation

Table 12 below outlines the XML schema structure of the type_cat_processing_mode:

Table 12: XML schema syntax of the type_cat_processing_mode

```
<!-- type_cat_processing_mode -->
   <xs:complexType name="type_cat_processing_mode">
      <xs:choice>
         <xs:element name="cat_stopping" type="cat_stopping_processing"/>
         <xs:element name="cat_passthrough" type="cat_passthrough_processing"/>
         <xs:element name="cat_patching" type="cat_patching_processing"/>
         <xs:element name="cat_regeneration" type="cat_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>

<!-- cat_stopping_processing -->
   <xs:simpleType name="cat_stopping_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- cat_passthrough_processing -->
   <xs:simpleType name="cat_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- cat_patching_processing -->
   <xs:complexType name="cat_patching_processing">
      <xs:sequence>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="ca_provider"
type="type_ca_provider"/>
      </xs:sequence>
   </xs:complexType>

<!-- cat_regeneration_processing -->
   <xs:complexType name="cat_regeneration_processing">
      <xs:sequence>
         <xs:element name="table_repetition_period" type="xs:int"/>
         <xs:element name="offset" type="xs:int"/>
         <xs:element name="CAT_version_number" type="type_version_number"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="ca_provider"
type="type_ca_provider"/>
      </xs:sequence>
   </xs:complexType>

<!-- ca_provider: number of instances equals number of output EMM PID's -->
   <xs:complexType name="type_ca_provider">
      <xs:sequence>
         <xs:element name="CAS_id" type="xs:int"/>
         <xs:element name="output_EMM_PID" type="type_pid"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.8.3 Definition of XML elements and XML types

type_cat_processing_mode: Indicates the mode of the CAT processing as follows:

cat_stopping_processing: No CATs are part of the related output TS.

cat_passthrough_processing: The incoming table is unchanged. Only PID remapping may be applied if required by the pid_processing sub-section.

cat_patching_processing: CAT patching, i.e. EMMs at the DSA output are accompanied by output_EMM_PID's, which are different from their counterparts at the DSA input. This procedure is performed for all retained CAS_id's. CAS-related information is updated according to the sequence of retained CAS_id's. Part of the CAS_id's might be replaced by stuffing descriptors, because the

terrestrially retransmitted service component consists of fewer CASs than the one received from the parent site.

cat_regeneration_processing: The CATs to be output at DSA's interface L are created based on DSA Control Information and CATs originally addressing DTH receivers.

table_repetition_period: Indicates the repetition period of the CAT as the number of 90 kHz ticks.

offset: Defines an offset relative to the beginning of the insertion window for the insertion of the first related table within this insertion window. See clause 6.4.4.4.1 for the insertion window mechanism.

CAT_version_number: Version number of the related CAT to be output by the DSA, see EN 300 468 [6].

ca_provider: Describes a combination of a CAS_id and an output_EMM_PID.

CAS_id: Conditional Access System identifier, see EN 300 468 [6].

output_EMM_PID: Packet Identifier of the EMM-carrying packets at the output of the DSA according to the regenerated CAT.

5.7.3.9 SDT and BAT processing

5.7.3.9.1 Overview

The XML type sdt_bat_processing_mode covers pass-through, patching and regeneration of Service Description Tables and Bouquet Association Tables – see EN 300 468 [6] – and indicates in the case of patching the DSACI sets for other multiplexes as well as it maps input bouquet identifiers to output ones. In the case of regeneration this XML type assigns an integer multiplier M_actual of the different repetition periods of SDT_actual and SDT_other's as well as assigning further parameters making up the regenerated SDTs and BATs.

5.7.3.9.2 XML schema representation

Table 13 below outlines the XML schema structure of the type_sdt_bat_processing_mode:

Table 13: XML schema syntax of the type_sdt_bat_processing_mode

```
<!-- type_sdt_bat_processing_mode -->
   <xs:complexType name="type_sdt_bat_processing_mode">
      <xs:choice>
         <xs:element name="sdt_bat_passthrough" type="sdt_bat_passthrough_processing"/>
         <xs:element name="sdt_bat_patching" type="sdt_bat_patching_processing"/>
         <xs:element name="sdt_bat_regeneration" type="sdt_bat_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>

<!-- sdt_bat_passthrough_processing -->
   <xs:simpleType name="sdt_bat_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- sdt_bat_patching_processing -->
   <xs:complexType name="sdt_bat_patching_processing">
      <xs:sequence>
         <xs:element name="sdt_crossreferencing_flag" type="xs:boolean"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
```

```
        <xs:element maxOccurs="unbounded" minOccurs="0" name="bouquet" type="type_bouquet"/>
      </xs:sequence>
   </xs:complexType>

<!-- sdt_bat_regeneration_processing -->
   <xs:complexType name="sdt_bat_regeneration_processing">
      <xs:sequence>
         <xs:element name="sdt_actual_period" type="xs:int"/>
         <xs:element name="M_actual" type="xs:int"/>
         <xs:element name="offset" type="xs:int"/>
         <xs:element name="sdt_actual_version_number" type="xs:int"/>
         <xs:element name="sdt_crossreferencing_flag" type="xs:boolean"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="bouquet" type="type_bouquet"/>
      </xs:sequence>
   </xs:complexType>

<!-- other_mux: number of instances equals number of defined other muxes -->
   <xs:complexType name="type_other_mux">
      <xs:sequence>
         <xs:element name="other_DSA_group_id" type="xs:int"/>
         <xs:element name="DSACI_PID" type="type_pid"/>
         <xs:element name="source_id" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_bouquet -->
   <xs:complexType name="type_bouquet">
      <xs:sequence>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="input_bouquet_id" type="xs:int"/>
         <xs:element name="terrestrial_bouquet_id" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.9.3 Definition of XML elements and XML types

type_sdt_bat_processing_mode: Indicates the mode of the SDT processing as follows:

sdt_bat_passthrough_processing: The incoming SDT and BAT tables are unchanged. Only PID remapping may be applied if required by pid_processing element.

sdt_bat_patching_processing: The relevant attributes for SDT and BAT tables as described in table 16 shall be modified according to values provided by others DSACI configuration elements. SDT actual sections not relevant for the current output TS are converted to others or converted to stuffing section according to sdt_crossreferencing_flag.

sdt_bat_regeneration_processing: The SDTs to be output at DSA's interface L are created based on DSA Control Information and SDTs received from parent side, which are addressing consumer receivers directly.

sdt_crossreferencing_flag: Indicates, if cross-referencing other Transport Streams of the same DVB-T2 multiplex shall be done or not. Not applicable to the DVB-T case and to be ignored in that case. When sdt_crossreferencing_flag equals '0', cross-referencing shall not be used, when sdt_crossreferencing_flag equals '1', cross-referencing shall be used.

sdt_actual_period: Indicates the repetition period of SDT_actual as the number of 90 kHz ticks.

M_actual: Indicates an integer that represents a multiple of SDT_actual and all SDT_other repetition periods.

offset: Defines an offset relative to the beginning of the insertion window for the insertion of the first related table within this insertion window. See clause 6.4.4.4.1 for the insertion window mechanism.

sdt_actual_version_number: Sets the version_number of the SDTactual at the DSA output, see EN 300 468 [6]. The version numbers of the SDTother tables is equivalent to the sdt_actual_version_number in the related other DSACI sets identified by the corresponding other_DSA_group_id's.

sdt_crossreferencing_flag: See above.

other_DSA_group_id: Identifier of another set of DSA Configuration Information.

DSACI_PID: Packet Identifier of the TS packets carrying the DSA Configuration Information identified by other_DSA_group_id above.

source_id: Identifies the related Transport Stream uniquely and replaces the TS_Id/ON_Id double. That TS carries the DSA Configuration Information identified by other_DSA_group_id above.

source_id: See above.

input_bouquet_id: Identifier of the related bouquet at the input of the DSA.

terrestrial_bouquet_id: Identifier of the related terrestrial bouquet at the output of the DSA.

5.7.3.10 EIT processing

5.7.3.10.1 Overview

The XML type_eit_processing_mode covers pass-through, patching and regeneration of Event Information Tables – see EN 300 468 [6].

5.7.3.10.2 XML schema representation

Table 14 below outlines the XML schema structure of the type_eit_processing_mode:

Table 14: XML schema syntax of the type_eit_processing_mode

```
<!-- type_eit_processing_mode -->
   <xs:complexType name="type_eit_processing_mode">
      <xs:choice>
         <xs:element name="eit_passthrough" type="eit_passthrough_processing"/>
         <xs:element name="eit_patching" type="eit_patching_processing"/>
         <xs:element name="eit_regeneration" type="eit_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>

<!-- eit_passthrough_processing -->
   <xs:simpleType name="eit_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- eit_patching_processing -->
   <xs:complexType name="eit_patching_processing">
      <xs:sequence>
```

```
        <xs:element name="eit_crossreferencing_flag" type="xs:boolean"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
      </xs:sequence>
   </xs:complexType>

<!-- eit_regeneration_processing -->
   <xs:complexType name="eit_regeneration_processing">
      <xs:sequence>
         <xs:element name="eit_pf_actual_period" type="xs:int"/>
         <xs:element name="eit_pf_other_period" type="xs:int"/>
         <xs:element name="eit_sch_1stday_actual_period" type="xs:int"/>
         <xs:element name="eit_sch_1stday_other_period" type="xs:int"/>
         <xs:element name="eit_sch_2nd_8th_actual_period" type="xs:int"/>
         <xs:element name="eit_sch_2nd_8th_other_period" type="xs:int"/>
         <xs:element name="eit_sch_aft_8th_actual_period" type="xs:int"/>
         <xs:element name="eit_sch_aft_8th_other_period" type="xs:int"/>
         <xs:element name="eit_insertion_window_duration" type="xs:int"/>
         <xs:element name="eit_crossreferencing_flag" type="xs:boolean"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="DTT_only_service"
type="type_DTT_only_service"/>
      </xs:sequence>
   </xs:complexType>

<!-- other_mux: number_of_instances equals number of defined other muxes -->
   <xs:complexType name="type_other_mux">
      <xs:sequence>
         <xs:element name="other_DSA_group_id" type="xs:int"/>
         <xs:element name="DSACI_PID" type="type_pid"/>
         <xs:element name="source_id" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>

<!-- DTT_only_service: number of instances equals number of defined DTT-only services -->
   <xs:complexType name="type_DTT_only_service">
      <xs:sequence>
         <xs:element name="DTT_only_service_id" type="xs:int"/>
         <xs:element name="input_EIT_PID" type="type_pid"/>
         <xs:element name="source_id" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.10.3 Definition of XML elements and XML types

type_eit_processing_mode: Indicates the mode of the EIT processing as follows:

eit_passthrough_processing: The incoming tables are unchanged. Only PID remapping may be applied if required by the pid_processing sub-section.

eit_patching_processing: The relevant attributes for EIT tables as described in table 16 shall be modified according to values provided by others DSACI configuration elements. EIT actual sections not relevant for the current output TS are converted to others, or converted to stuffing section according to eit_crossreferencing_flag.

eit_regeneration_processing: The EITs to be output at DSA's interface L are created based on EITs received from parent sidethat are addressing consumer receivers directly.

eit_crossreferencing_flag: Indicates, if cross-referencing other Transport Streams of the same DVB-T2 multiplex shall be done for EITs or not. Not applicable to the DVB-T case and to be ignored in that case.

When eit_crossreferencing_flag equals ‘0’, cross-referencing shall not be used, when eit_crossreferencing_flag equals ‘1’, cross-referencing shall be used.

eit_pf_actual_period: Repetition period of EIT Present/Following Table for the actual multiplex expressed in 90 kHz ticks.

eit_pf_other_period: Repetition period of EIT Present/Following Tables for other TSs expressed in 90 kHz ticks.

eit_sch_1stday_actual_period: Repetition period of EIT Schedule table for the first full day for the actual multiplex expressed in 90 kHz ticks.

eit_sch_1stday_other_period: Repetition period of EIT Schedule table for the first full day for other TSs expressed in 90 kHz ticks.

eit_sch_2nd_8th_actual_period: Repetition period of EIT Schedule table for the second full day to the 8Th full day for the actual multiplex expressed in 90 kHz ticks.

eit_sch_2nd_8th_other_period: Repetition period of EIT Schedule table for the second full day to the 8Th full day for the actual multiplex expressed in 90 kHz ticks.

eit_sch_aft_8th_actual_period: Repetition period of EIT Schedule table for further than 8 days for the actual multiplex expressed in 90 kHz ticks.

eit_sch_aft_8th_other_period: Repetition period of EIT Schedule table for further than 8 days for the actual multiplex expressed in 90 kHz ticks.

eit_insertion_window_duration: Indicates the length in time of the EIT insertion window, see clause 6.4.4.4.1 for the insertion window mechanism.

eit_crossreferencing_flag: See above.

other_DSA_group_id: Identifier of another set of DSA Configuration Information.

DSACI_PID: Packet Identifier of the TS packets carrying the DSA Configuration Information identified by DSACI_id above.

source_id: Identifies the related Transport Stream uniquely and replaces the TS_Id/ON_Id double.

DTT_only_service_id: Service identifier of a service not addressing consumer receivers directly.

input_EIT_PID: Packet Identifier of an EIT at a DSA input that will be used for EIT regeneration.

source_id: See above.

5.7.3.11 Output configuration

5.7.3.11.1 Overview

The XML type_output_processing configures the output TS(s) of the DSA as either a DVB-T-mega-frame stream (DVB-T case) or encapsulates the T2-MI packets at the output of the DSA into TS packets (interface L in figure 1):

5.7.3.11.2 XML schema representation

Table 15 below outlines the XML schema structure of the type_output_processing:

Table 15: XML syntax of the type_output_processing

```
<!-- type_output_processing -->
   <xs:complexType name="type_output_processing">
      <xs:sequence>
         <xs:element name="terrestrial_standard_generation"
type="type_terrestrial_standard_generation" />
      </xs:sequence>
   </xs:complexType>

<!-- type_terrestrial_standard_generation -->
   <xs:complexType name="type_terrestrial_standard_generation">
      <xs:choice>
         <xs:element name="dvb_t" type="dvb_t_standard"/>
         <xs:element name="dvb_t2" type="dvb_t2_standard"/>
      </xs:choice>
   </xs:complexType>

<!-- dvb_t_standard -->
   <xs:simpleType name="dvb_t_standard">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- dvb_t2_standard -->
   <xs:complexType name="dvb_t2_standard">
      <xs:sequence>
         <xs:element name="output_T2_MI_PID" type="type_pid"/>
         <xs:element name="output_T2_MI_stream_id" type="xs:int"/>
         <xs:element name="output_rate" type="xs:int"/>
         <xs:element maxOccurs="1" minOccurs="0" name="L2_signalling" type="type_L2_signalling"/>
      </xs:sequence>
   </xs:complexType>

<!-- L2_signalling -->
   <xs:complexType name="type_L2_signalling">
      <xs:sequence>
         <xs:element name="output_TS_id" type="xs:int"/>
         <xs:element name="output_ON_id" type="xs:int"/>
         <xs:element name="output_L2_service_id" type="xs:int"/>
         <xs:element name="output_L2_service_provider_name" type="xs:string"/>
         <xs:element name="output_L2_service_name" type="xs:string"/>
         <xs:element name="output_L2_PCR_PID" type="type_pid"/>
         <xs:element name="output_L2_PMT_PID" type="type_pid"/>
      </xs:sequence>
   </xs:complexType>
```

5.7.3.11.3 Definition of XML elements and XML types

terrestrial_standard_generation: Indicates the generation of the terrestrial physical layer to be used for transmission on daughter site as follows:

dvb_t: DVB-T, i.e. first generation

dvb_t2: DVB-T2, i.e. second generation

output_T2_MI_PID: Packet Identifier of the TS packets, into which the T2-MI stream is encapsulated.

output_T2_MI_stream_id: Identifier of the T2-MI stream at the output of the DSA, i.e. interface L. The format of this identifier is defined by TS 102 773 [1].

output_rate: Bitrate of the output T2-MI oder T-mega-frame stream.

output_TS_id: Identifies the output Transport Stream the loop instance is covering.

output_ON_id: Identifies the Original Network the aforementioned output TS stems from. Together with the output_TS_id it identifies the TS uniquely.

output_L2_service_id: Service identifier of the service output by the DSA at interface L. The format of DVB-T-mega-frame or DVB-T2-MI stream is a Transport Stream. The format of this service identifier is identical to the service identifier format defined by EN 300 468 [6].

output_L2_service_provider_name: Service provider name of the single service output at DSA interface L.

output_L2_ service_name: Service name of the single service output at DSA interface L.

output_L2_PCR_PID: Packet Identifier of those TS packets carrying PCR in the TS output at interface L of the DSA.

output_L2_PMT_PID: Packet Identifier of those TS packets – part of the DVB-T-mega-frame or datapiped DVB-T2-MI stream - at the DSA output interface L carrying PMT sections.

5.7.4 DSACI schema

The DSACI schema, i.e. the mandatory XML syntax of the Configuration Information for the Daughter Site Adapter is defined in Annex A of this document.

5.7.5 In-band carriage of the DSACI XML file

The DSACI XML file built on the basis of the schema described in clause 5.7.4 above is gzipped prior to provision to the DSAs – in compliance with RFC1952 [12].

That gzipped file is provided to the DSA making use of the private_section syntax defined in ISO/IEC 13818-1 [5] and output by the CSG in a carousel manner – at least once per minute. The DSACI section created this way is illustrated with table 16 below:

Table 16: DSACI section based on private_section syntax

Syntax	Number of bits	Identifier
DSACI_section		
table_id	*8*	*uimsbf*
section_syntax_indicator	*1*	*bslbf*
reserved_future_use	*1*	*bslbf*
reserved	*2*	*bslbf*
section_length	*12*	*uimsbf*
current_DSA_group_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<N1;i++){		
DSACI_file_data	8	uimsbf
}		
CRC_32	32	uimsbf

current_DSA_group_id: Identifier of a group of DSAs that are configured the same way and are hence producing the same terrestrial signal.

DSACI_file_data: A single byte of the gzipped DSACI XML file.

For the definition of all other parameters being part of table 16 above see ISO/IEC 13818-1 [5].

5.8 Carriage of signal components within the parent Transport Streams

Service components and PSI/SI intended for direct reception of the Parent Signal by commercial receivers shall be carried using the normal DVB mechanisms as specified in DVB-S, EN 300 421 [8], DVB-S2/S2X, EN 302 307-1/EN 302 307-2 [9]/[10], DVB-T, EN 300 744 [3], DVB-T2, EN 302 755 [2] and DVB-SI, EN 300 468 [6].

For the SIS metadata carried in the Parent Signal the following applies:

- Each CSG shall produce a single SIS Service of service_type = $0C_{16}$ (data broadcast service, see EN 300 468 [6]) that contributes to the Parent Signal.
- The PMT for the SIS Service shall indicate the PCR_PID making reference to the component that carries the PCR_{abs} (see section 5.3 of this document).
- When present in the Parent Signal, DSACI is declared as a component of stream_type 06_{16} inside the SIS Service (see section 5.7 of this document). Additionally, a data_broadcast_id_descriptor shall be used with the following parameters:
    - data_broadcast_id = 101_{16}
    - id_selector_byte = 02_{16}
- When present in the Parent Signal, F&TI is declared as a component of stream_type 06_{16} inside the SIS Service (see section 5.6 of this document). Additionally, a data_broadcast_id_descriptor shall be used with the following parameters:
    - data_broadcast_id = 101_{16}
    - id_selector_byte = 01_{16}

- In DVB-T2 use case, T2-MI packets containing L1-signalling (sub-clause 5.4.1) and Framing & Timing Information (encapsulated in T2-MI packets of type $F0_{16}$, see sub-clause 5.6.2) as well as other permitted T2-MI packets in general (clause 5.5) shall be carried via Data Piping as defined in TS 102 773 [1], on a single component (with stream_type = 06_{16}, see ISO/IEC 13818-1 [5]) of the SIS Service.

- In the DVB-T use case, a single component of stream_type = 06_{16}, declared in the SIS Service, shall carry the F&TI packets, see clause 5.6.3.

- When required by the DSACI, additional terrestrial PSI/SI tables belonging to a hybrid service are added to the SIS Service as components of stream_type 06_{16}. A data_broadcast_id_descriptor shall be used with the following parameters:

    - data_broadcat_id = 101_{16}
    - id_selector_byte = 03_{16}

6 Daughter site processing

6.1 Overview (DVB-T2 and -T cases)

The Daughter Site Adapter (DSA) consists of the following stages that perform the listed operations (DVB-T2-specific operations appear in **bold characters**, DVB-T-specific operations in *Italic characters*):

- Bootstrapping (see clause 6.2)
    - Locate DSACI in-band or out-of-band (manual configuration)
- Input processing (see clause 6.3)
    - Identification of the Transport Streams at the input stage of the DSA services and service components are to be extracted from for daughter site re-transmission (see …)
    - Identification of the PIDs of the PMTs belonging to the SIS Services (one per Parent Signal) (see …)
    - Identification of the single Primary SIS Service beneath all SIS Services being part of the Parent Signals (see 5.7.y)
    - (Virtual) arrival timestamping (see 6.3.x)
- TS remultiplexing (see clause 6.4)
    - Assigning output Transport Streams to Physical Layer Pipes – in the case of DVB-T2 (see …)
    - Generation of Reference Transport Stream(s)
    - Selection of relevant content from input TSs (clause 6.4.3)
    - PSI/SI selection and insertion (optionally also generation of PSI/SI tables/sections)
    - Deterministic re-multiplexing
- *DVB-T mega-framing*
    - *Remove megaframe_timestamping function (see clause 6.4.2.3) from MIP packets*
- **DVB-T2 BaseBand Framing**
    - **TS splitting (optional)**
    - **Terrestrial Framing**
    - **Null Packet Deletion (optional)**
    - **BaseBand Frame generation**
- **T2-MI packet handling**
- **Output processing**

The aforementioned stages and tasks are illustrated with figures 4 and 6 below for the DVB-T2 and -T cases respectively and are explained in the following sub-clauses. Input processing and TS re-multiplexing are widely common for both broadcast systems – T2 and T. Framing, T2-MI packet handling and output processing, in turn, are T2- and T-specific. The structure of clause 6 reflects these circumstances.

NOTE: The following clauses describe conceptual operations. A real DSA may combine multiple steps into one operation.

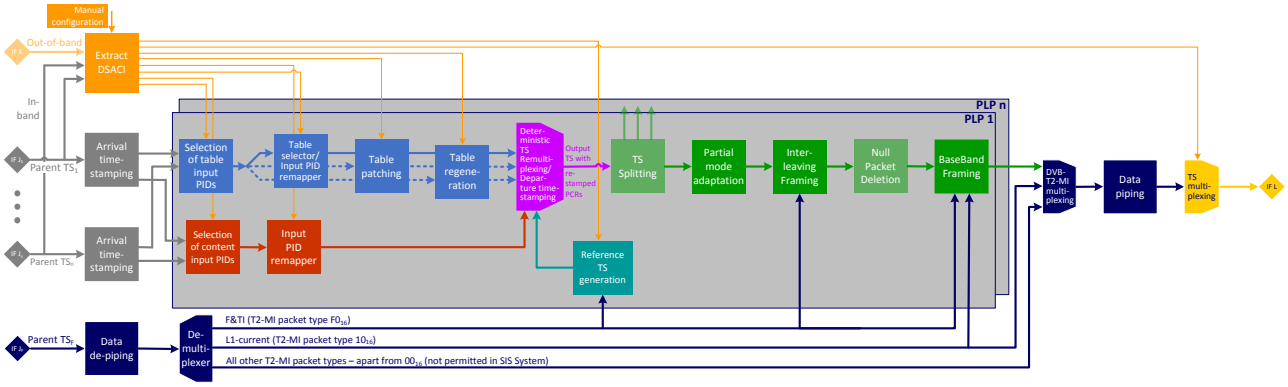

Figure 5: DSA processing (DVB-T2 case)

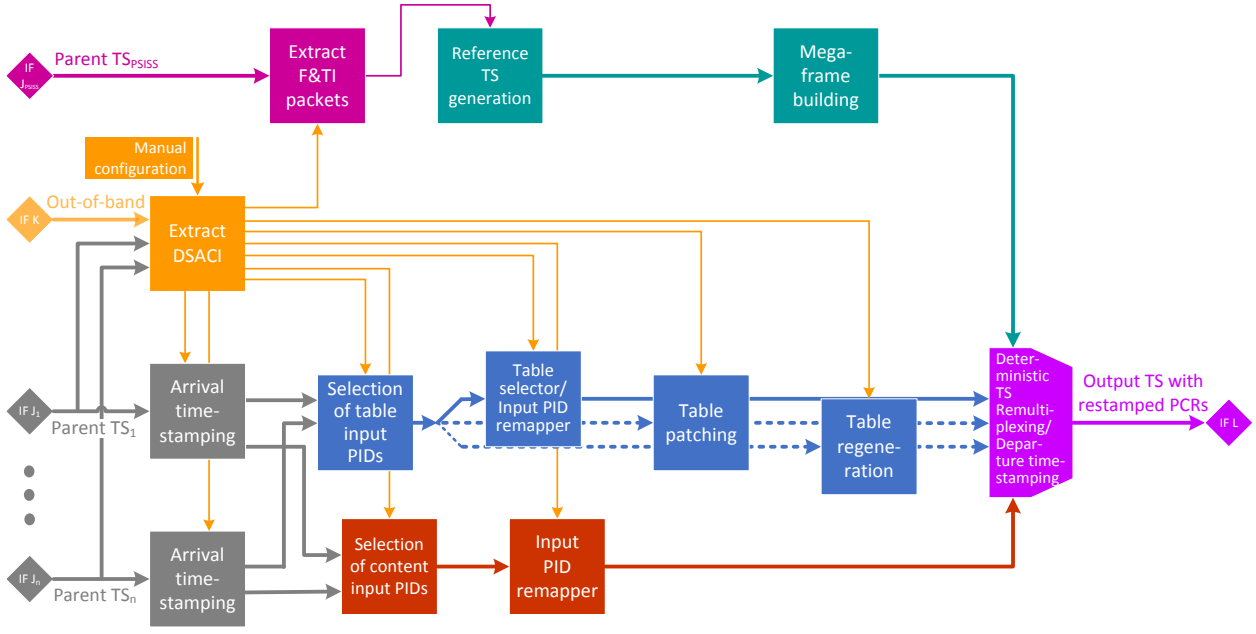

Figure 6: DSA processing (DVB-T case)

6.2 Bootstrapping

The configuration of the receivers providing the Parent Signals to the DSA is out of scope of this document. The described operations either for bootstrapping or any other later processing is based on the assumption that the different involved Parent Signals (Transport Streams) have been made available to the DSA.

The bootstrapping operation for the DSA consists of providing to the DSA the path to the DSACI . Two cases are described - whether the DSACI is provided in-band (i.e. in one of the Parent Signals) or out-of-band:

6.2.1 Gaining access to DSACI provided in-band

The initial manual configuration of the DSA consists of providing to the DSA the following:

- The [TSid/ON ID/Program Id] triple corresponding to the Primary SIS Service in the relevant Parent Signal where the DSA will find the current DSACI component
- The current_DSACI_group_id identifying the current DSACI that this particular DSA shall use

6.2.2 Out-of-band DSACI provision

When DSACI is provided out-of-band, the initial manual configuration of the DSA consists of providing to the DSA the following:

- The path or location (eg URL, address, ..) where the DSA can find the current DSACI file
- The current_DSACI_group_id identifying the current DSACI that this particular DSA shall use

Note: When provided out-of-band, DSACI may be presented in another format compared to in-band distribution. This document defines in sub-clause 5.7.5 the DSACI distribution format in the case of in-band delivery. Out-of-band provision is not defined by this document.

6.3 Input processing

6.3.1 Reception of parent Transport Streams

The Daughter Site Adapter (DSA) receives one or several parent multiplex(es). Those are indicated by the DSA Configuration Information (DSACI) the DSA receives from the CSG (see sub-clause 5.7).

The Daughter Site Adapter receives one or several parent Transport Streams. These DVB TS packets may be received over any standard DVB Transport Stream interface, such as DVB-S, -S2 or S2X, ASI or encapsulated within IP packets in accordance with TS 102 034 [7].

6.3.1.1 Arrival timestamping

For each input TS on interface(s) H, packets shall have a timestamp ($packet\_arrival\_time$) computed according to the incoming PCR_{abs} values:

- For the packet of index n, holding the k[th] PCR_{abs} value in adaptation field :

$$packet\_arrival\_time(n) = PCR_{abs}(k)$$

- For packet of index p, holding the (k+1)[th] PCR_{abs} value in adaptation field :

$$packet\_arrival\_time(p) = PCR_{abs}(k+1)$$

- For packets, of same input TS, with index i, such that $n \leq i \leq p$:

$$packet\_arrival\_time(i) = PCR_{abs}(k) + \left\lfloor \frac{(i-n) \times \left(PCR_{abs}(k+1) - PCR_{abs}(k)\right)}{(p-n)} \right\rfloor$$

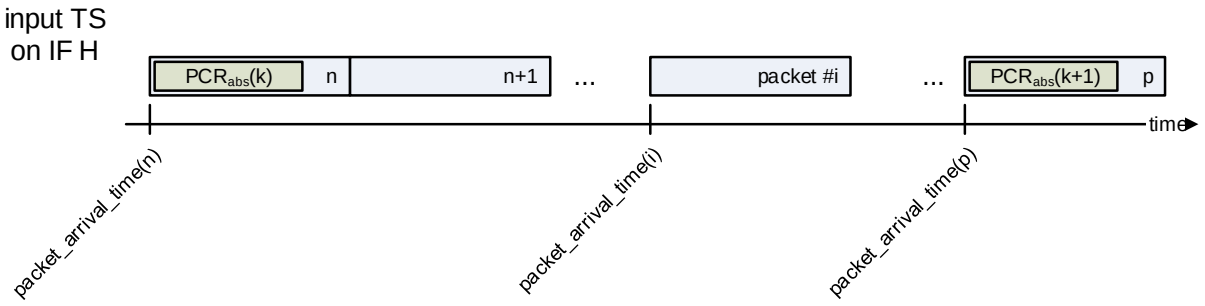

Figure 7: Arrival timestamping mechanism

6.3.1.2 Calculation of Virtual Arrival Timestamps (VATs)

6.3.1.2.1 Overview

The Virtual Arrival Timestamping method enables insertion of TS packets that are not present in the Parent Signal or of TS packets that are regenerated by the DSA. This may be done to save bandwith in the Parent Signal, for example by generating DTT PMT or EIT sections on daughter site as an alternative to receiving those from the CSG. The mechanism for the generation of a stream of TS packets ensures that multiple DSAs generate identical packets and an identical packet sequence.

Below, the method for insertion of TS packets generated with constant packet rate is defined. This method is intended for use for e.g. the insertion of regenerated PMTs.

6.3.1.2.2 CBR operation mode

The Virtual Arrival Timestamp of TS packets is calculated by the DSA based on a time interval indicated as part of the DSA Configuration Information. The time interval in question is equivalent to the indicated table_repetition_period. The DSA converts that table_repetition_period into a time interval for the insertion of TS packets carrying the corresponding PSI/SI table sections as follows:

$$packet\_period = \left\lfloor \frac{table\_repetition\_period}{N_{pkt}} \right\rfloor$$

Where N_{pkt} is the number of TS packets needed to packetize a whole table according to the packetization rules applicable to the sections of this table (see section 6.4.4.4), and table_repetition_period is the

repetition period of the table as a number of PCR_base clock ticks. The result is hence a duration as integer number of 90 kHz ticks.

The Virtual Arrival Timestamp VAT for the i[th] TS packet(i) transmitted since SIS epoch is calculated as follows:

$$VAT(i) = (packet\_period \times i) + offset$$

Where the number *i* is defined using the current time since SIS epoch (as number of PCR_base clock ticks):

$$i = \left\lfloor \frac{time\_since\_epoch}{packet\_period} \right\rfloor$$

offset is a PCR_base count provided as part of DSACI.

When expanding VAT to PCR base and extension format, the 9 extension bits are set to zero.

The Continuity Counter for the packet of index *i* is:

$$continuity\_counter\ (i) = i\ mod\ 16$$

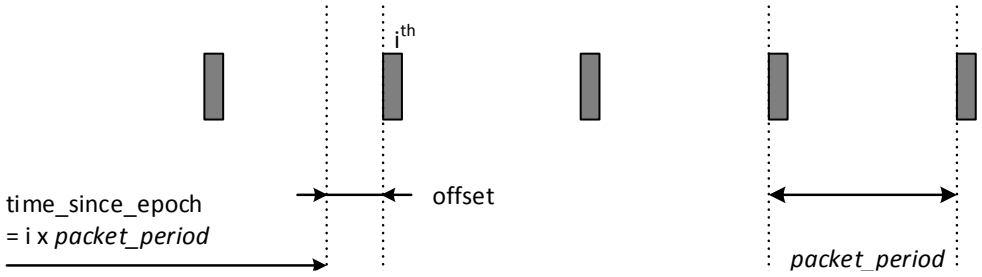

Figure 8: Generation of Virtual Arrival Timestamp for packets in CBR mode

When DSACI provides a new value for a table_repetition_period, the version number of the table that repetition period belongs to shall be incremented. At the indicated time instance of the change – global_application_time, also part of DSACI (see clause 5.7.3.1) – the DSA shall stop broadcasting packets of the old table version and introduce a single TS packet containing stuffing section with a VAT equal to the aforementioned time instance indicated as part of DSACI.

Then the process of Virtual Arrival Timestamping restarts as decribed above with the new repetition rate parameter setting. The stuffing packet is used to trigger a discontinuity (with discontinuity_indicator in its adaptation field) and provides the new value of the continuity_counter. This new value is computed by the DSA to match the following packet continuity_counter which contains the start of the section.

6.4 TS re-multiplexing

6.4.1 Overview

The TSs for each PLP shall be created by first generating a Reference TS (see clause 6.4.2 below), and then replacing null packets with selected packets from the incoming parent TSs, with packets stemming originally from the incoming parent TSs but were modified, and with packets generated locally by the DSA. Figure 6 above illustrates the TS remultiplexing in detail, because for the DVB-T case that represents the major part of the DSA processing.

6.4.2 Generation of Reference Transport Streams

6.4.2.1 Overview

A Reference Transport Stream (RTS) is a Transport Stream made up entirely of null packets (ISO/IEC 13818-1 [5]) but with a defined timing in terms of the SIS system clock. Each packet is connected to an associated PCR_{abs} value.

The Reference Transport Stream acts as a placeholder with correct time positions for the null packets, which are subsequently replaced by TS packets carrying service component content or PSI/SI metadata to form the output Transport Stream.

6.4.2.2 DVB-T2 case

There shall be one RTS for each data PLP, but the RTSs for the PLPs within the same PLP_GROUP shall be identical in case a common PLP is present, see EN 302 755 [2].

The RTS shall include packets with PCR_{abs} values equal to values of the PCR_{FTSP} signalled in the F&TI signalling packets for the relevant PLP. Between these packets there shall be further Null packets with equally spaced PCR_{abs} values (as defined below), such that the total number of packets including the packet with a given PCR_{FTSP} value up to but excluding the packet with the subsequent PCR_{FTSP} value shall be equal to the frame_packet_count in the T2-MI packet carrying F&TI containing the first PCR_{FTSP} value.

The PCR_{abs} values for the packet of index $i, 0 \leq i < frame\_packet\_count(n)$ shall be given by

$$PCR_{abs}(i) = PCR_{FTSP(n)} + \left\lfloor \frac{i \times (DPCR_{IF})}{frame\_packet\_count(n)} \right\rfloor$$

NOTE: $DPCR_{IF} = PCR_{FTSP}(n+1) - PCR_{FTSP}(n)$

6.4.2.3 DVB-T case

6.4.2.3.1 Overview

In the case of DVB-T, a mega-frame building block, see TS 101 191 [4], sits between the Reference Transport Stream generator and the TS remultiplexer, see figure 6 above. The latter remultiplexer is directly connected to interface L, see figure 6 above as well.

6.4.2.3.2 Extraction of F&TI packets

The F&TI packets are extracted by the DSA. The modulation parameters TPS are derived from the F&TI packets.

The DSA shall remove the megaframe_timestamping function when building the MIP packet for the DVB-T transmission and shall change the PID to the standardized PID 15_{16}.

PCR_{abs} is then assigned to the MIP packet.

The bitrate of the Reference Transport Stream, and therefore the number of packets in a mega-frame, N_{MF}, is defined by the modulation parameters TPS.

The PCR_{abs} value associated with the MIP packet for mega-frame n, denoted $PCR_{MF}(n)$, shall be equal to the value given by the PCR_ABS_BASE and PCR_ABS_EXT fields signalled in the megaframe_timestamping function of the MIP packet (see clause 5.6.3).

The PCR_{abs} value associated with the packet of index $i, 0 \leq i < N_{MF}$ shall be given by

$$PCR_{abs}(i) = PCR_{FTSP(n)} + \left\lfloor \frac{i \times (DPCR_{MF})}{N_{MF}} \right\rfloor$$

NOTE: $DPCR_{MF} = PCR_{MF}(n+1) - PCR_{MF}(n)$

6.4.3 Selection of relevant content from input TSs

The services and service components that shall be extracted for terrestrial re-transmission are determined by the DSA Configuration Information (DSACI). Within a Transport Stream, the TS packets are selected by PID.

The PID component selection shall also be used for remapping of PSI/SI sent on hidden PIDs as per clause 4.3.

6.4.4 Processing and generation of Layer 2 signalling (PSI/SI)

6.4.4.1 Background

Terrestrial PSI/SI is inserted into the output TS in a way that it enables the receiver/decoder to select and locate the broadcast services and the components of the services. PSI/SI is partly dynamic and time-accurate to support changes in the service status and changes in the service composition.

PSI/SI for the terrestrial multiplex also assists the viewer in selecting services and the related events.

The following clauses detail the four methods available for insertion of terrestrial PSI/SI by the DSAs – depending on instructions being part of DSACI.

6.4.4.2 Pass-through remultiplexing of selected parent tables/sections

When a Parent Signal component is also applicable to a DTT multiplex, it may be passed to the output without content modification. When the CSG builds a DSACI that complies with this situation (see section

5.7), the DSA shall pass-through all packets of the target PSI/SI component. According to the DSACI, the output PID may be modified to match the DTT PSI/SI consistency.

The CSG may also use this method to send DTT PSI/SI sections in Parent Signal on non-default PIDs and build the DSACI so that the DSA will remap the PIDs to match DTT PIDs. These sections are part of an SIS Service, see clause 5.3 for details.

In this method, the $PCR_{abs}(i)$ computed at input timestamping remains valid for packet scheduling purposes (see section 6.4.5).

NOTE: This method may be applicable for example to PMT table.

6.4.4.3 Conversion of parent table/sections on the fly (patching)

In addition to the method described above, the conversion mechanism may be used to modify table attributes according to the DSACI inserted by CSG (see section 5.7). This method is applicable when the modified table does not need re-packetization: Only CRC updates and section invalidation using stuffing sections are allowed. To ensure Continuity Counter consistency, no incoming packet of the same component shall be discarded. Depending on DSACI, the output PIDs may be modified.

Table 17: PSI/SI table attributes that can be converted based on DSACI

Table	Attribute	DSACI element	DSACI type
PAT	transport_stream_id	output_TS_id	type_output_TS
	program_number	output_service_id	type_service
	program_map_PID	output_PMT_PID	type_service
PMT	program_number	output_service_id	type_service
	PCR_PID	output_PID	type_pid_proc
	elementary_PID	output_PID	type_pid_proc
	CA_PID (of CA_descriptor)	output_ECM_PID	type_ECM
SDT	table_id (actual →other)*		
	service_id	output_service_id	type_service
	transport_stream_id	output_TS_id	type_output_TS
	original_network_id,	output_ON_id	type_output_TS
	running_status	running_status	type_service
	free_ca_mode	free_ca_mode	type_service
BAT	bouquet_id	terrestrial_bouquet_id	type_bouquet
CAT	CA_PID (of CA_descriptor)	output_EMM_PID	type_ca_provider
EIT	table_id (actual →other)**		
	service_id	output_service_id	type_service
	transport_stream_id	output_TS_id	type_output_TS

	original_network_id, running_status free_ca_mode	output_ON_id running_status free_ca_mode	type_output_TS type_service type_service

Notes:

According to the transport_stream_id, original_network_id, service_id(s) in the incoming SDT section.

According to the transport_stream_id, original_network_id, service_id in the incoming EIT section.

In this method, the $PCR_{abs}(i)$ computed at input timestamping remains valid for packet scheduling purposes (see section 6.4.5).

6.4.4.4 Static table regeneration

- PAT, PMT , SDT, CAT regeneration
    - The PAT table is created in compliance with ISO/IEC 13818-1 [5], clause 2.4.4.3. The DSACI configures the TS_id, version_number and lists the ouput program_number and the associated PMT_PIDs. The services are asserted in ascending order based on the output program_number.
    - The CAT table is created in compliance with ISO/IEC 13818-1 [5], clause 2.4.4.6. DSACI configures the version number and the list of CA_system_id /EMM_PID pairs. The CA_descriptors are inserted in ascending order of CA_system_id and ascending order of EMM_PID.
    - The PMT skeletons for services not making use of remapping or patching original PMTs their PMTs are created in compliance with ISO/IEC 13818-1 [5], clause 2.4.4.8. The fields program_number, PCR_PID, the list of elementary streams are provided as part of DSACI. DSACI also indicates if conditional access information is to be removed. The following processing shall be implemented:
        - Extract the incoming PMTs identified by their related PMT_PID provided as part of DSACI
        - Create the new PMT using information given by DSACI up to the program_info loop
        - Compute the program_info loop, including a copy of the relevant descriptors in the same order as in the incoming table. When ca_provider's are removed, the matching descriptors shall be removed from the loop too. The ones that are kept shall follow the same order as the corresponding input CA_descriptors.
    - Compute the elementary stream loop: Copy the required elementary streams in the same order as in the incoming table, while also keeping the desired descriptor in their original

input order. When some of the ca_provider's are removed, the retained CA_descriptors shall appear in the same order as in the input PMT.

- Version Number of regenerated tables:
    - The Version Number of such a table is either part of the DSACI, i.e. provided by parent site, or it is identical to the Version Number of the incoming table
    - The DSACI consists of an indication regarding the source of the Version Number
- Insertion of tables into the output TS:
    - For SI specified within the present document the minimum time interval between the arrival of the last byte of a section to the first byte of the next transmitted section with the same PID, table_id and table_id_extension and with the same or different section_number shall be 25 ms. This limit applies for TSs with a total data rate of up to 100 Mbit/s.
- TS packetization:
    - The regenerated tables are composed of sections and the sections are packetized in TS packets according to the document 13818-1 [5], clause 2.4.4.
    - TS packetization shall be restricted to set the pointer_field to zero. In order to fill a TS packet after the end of a section , stuffing bytes 0xFF shall be added.
    - Padding based on using the adaptation_field shall not be used.

6.4.4.4.1 SDT and BAT regeneration.

The SDT table is created in compliance with EN 300 468 [6], clause 5.2.3. DSACI supplies the transport_stream_id, the version_number, the ONID, the list of input service to select. Each service_id_is associated with the output service_id with EIT_schedule_flag, EIT_present_following_flag, running status, free_CA_mode_ and the output provider name. The following processing is implemented:

- Extract the SDT tables from the different inputs used to generate the output stream.
- Create the new SDT using information given by DSACI
- For each required service copy from the incoming table, the parametres linked to the input service, update the information using the DSACI
- Insert in ascending order based on the output service_id the computed parameters

The BAT table is created in compliance with EN 300 468 [6], clause 5.2.2.

DSACI lists the different bouquets to manage. The following processing is implemented:

- Extract the BAT tables from the different inputs used to generate the output stream.
- For each extracted BAT, update if necessary the bouquet_id.

The SDT actual, SDT others and BAT generated sections are then played out using an insertion window. This insertion windows is defined using the restriction that all SDT others and BAT repetition periods are the same (given by DSACI, as 90 kHz tick count). Additionally, the SDT actual repetition period shall be a multiple value of the SDT other repetition period (integer M_actual given by DSACI).

Thus, the insertion window is defined as the duration of the SDT other repetition period. The very first window started at SIS epoch + offset (given by DSACI).

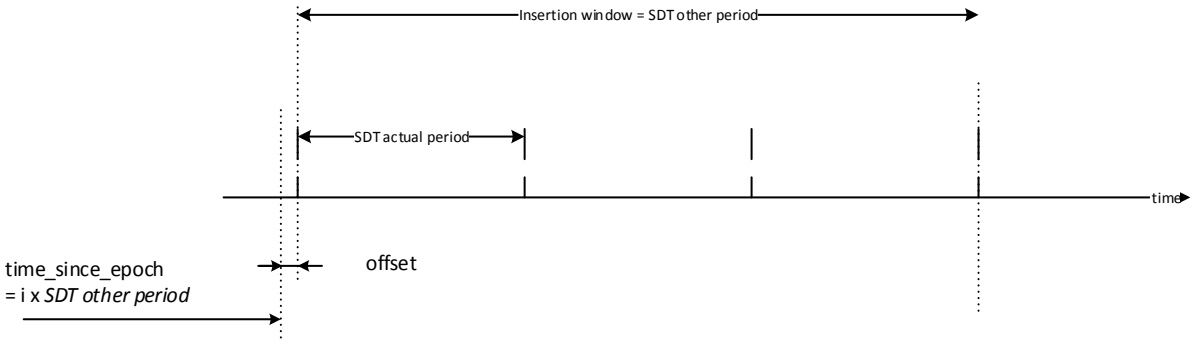

Figure 9: Definition of an SDT insertion window

In this windows the priority of insertion content is SDT actual, SDT others from smallest to greatest transport stream id, and BAT from smallest to largest bouquet_id.

All sections are packetized as described in clause 6.4.4.4 above. Stuffing sections are added so that the resulting number of packets is a multiple of 16. This way, packets of PID 11_{16} have a continuity counter equal to zero at the beginning of each insertion window.

Parameters are:

- n_{other}: Number of other Transport Streams
- n_{bat}: Number of BAT tables transmitted
- n_{stuff}: Number of stuffing TS packets should stuffing section content be needed (for continuity_counter wrapping).

As there is *M_actual* starts of SDT actual tables, we define *P* opportunities to start the transmission of a section in each SDT actual period.

$$P = 1 + \left\lceil \frac{n_{other} + n_{bat} + n_{stuff}}{M\_actual} \right\rceil$$

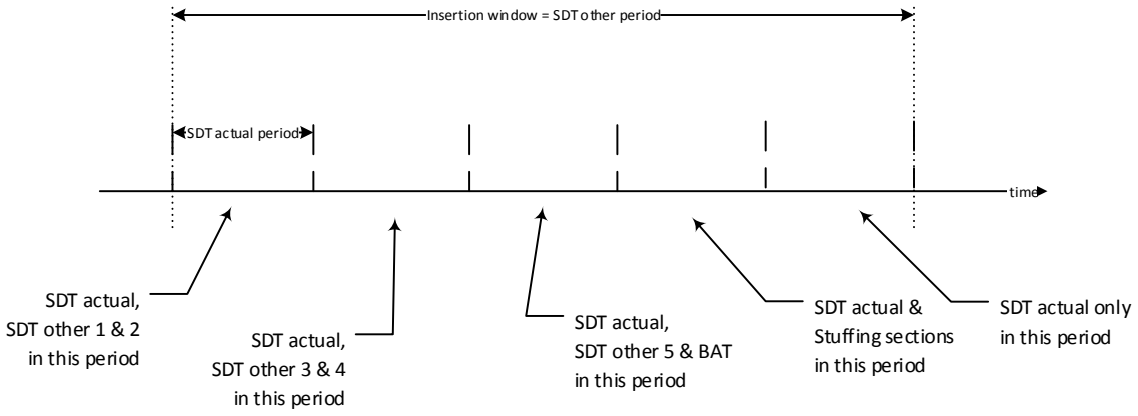

Figure 10: Example of repartitioning content within SDT actual interval, with P = 3 (M_actual = 5, n_{oth} = 5, n_{bat} = 1, n_{stuff} = 1)

At each opportunity an SDT or BAT or stuffing instance transmission starts according to the defined priority until all are transmitted.

During a SDT actual period, a constant packet period is used to define a VAT for each packet. It is computed by integer divison SDT actual period in 90 kHz ticks by the number of packets involved in the target section.

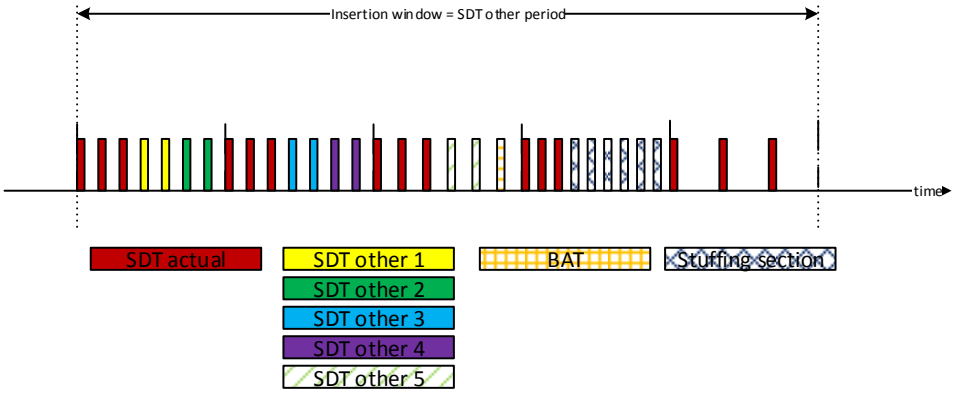

Figure 11: Example resulting packet distribution with 3 packets per SDT actual, 2 packets per SDT others, 1 packet per BAT, 6 stuffing sections of 1 packet

6.4.4.5 Dynamic table regeneration

The Event Information Tables (EITs) shall provide the output (DTT) transport streams with information about

- the Present/Following events for the Actual / Others DTT Transport Streams and
- the Scheduled events for the Actual / Others DTT Transport Streams.

The repetition rates of the EIT tables for DTT systems shall be reduced as DTT transmission offers a lower bandwidth than is provided by DTH systems.

An algorithm shall guarantee that all DSAs regenerate deterministically all the EIT tables in order to produce identical output (DTT) transmission, which is the condition to operate Single Frequency Network.

The algorithm described in this clause provide a method to guarantee that every DSA will perform the regeneration of the EIT tables similarly, in order to fulfil the SFN condition.

The algorithm shall be implemented in three phases :

- Phase (A) creation of a list of EIT_slots evenly spread over the EIT insertion window,
- Phase (B) creation of a database of EIT_sections extracted from the input (DTH) Transport Stream(s)
- Phase (C) playout of the EIT_slots list by mapping of the extracted EIT_sections_data in a series of TS packets to be inserted in the output (DTT) transport stream.

6.4.4.5.1 Phase (A): Creation of a list of EIT_slots

For each output (DTT) multiplex, a DSACI configuration file specifies the EIT_insertion_window duration, the repetition rate ***EIT_repetition_rate (EIT_category (i))*** required for all I categories of EIT and the list of services extracted from various input (DTH) multiplexes to build up the "actual" and the "others" output (DTT) multiplexes.

The Phase (A) shall be performed once each time a new configuration is implemented on either the "actual" or the "others" output (DTT) multiplexes.

The first step of the algoritm consists of computing the number of intervals for each EIT_category, using the following formulas:

number_of_EIT_interval (EIT_category (i)) = EIT_insertion_window_duration / EIT_repetition_rate (EIT_category (i))

The second step of the algorithm consists in computing the sub_interval between EIT_slots for each Service within a Category, using the following formula:

EIT_sub_interval (EIT_category (i)) = EIT_repetition_rate (EIT_category (i)) / number_of_service (EIT_category (i))

The third step of the algoritm consists of computing raw offsets for each Service within a EIT_category, using the following procedure:

```
EIT_slots_ptr = 0;
for (i=0; i < number_of_Category; i++) {
    for (j=0; j < number_of_EIT_interval (EIT_category (i)); j++) {
        for (k=0; k < number_of_Service_id(i); k++) {
            EIT_offset = ( j * EIT_repetition_rate (EIT_category (i)) ) + ( k * EIT_sub_interval (EIT_category (i)) )
            EIT_slots (EIT_slots_ptr) = Output_TS_id & Output_Service_id (k) & EIT_offset
            EIT_slots_ptr ++;
    }   }   }
```

The fourth step of the algoritm consists in performing a sort of the EIT_slots table in ascending order of the EIT_offset. When several EIT_slots have the same EIT_offset, then shall ordered in ascending order of the EIT_category.

The fifth step of the algorithm consists in smoothing the pattern of EIT_slots by computing their final offset in order to limit the jitter in the periodicity of EIT_slots.

The whole EIT_slots table is split in a number (EIT_Nb_insertion_step) of segments (EIT_Insertion_step), within each segments, the number (Nb_EIT_slots_to_spread) of EIT_slots to spread is computed and then, the final offset of each EIT_slot is computed as follows:

```
EIT_slots_size = 0 ;
for (i=0; i < number_of_Category; i++) {
    EIT_slots_size =  EIT_slots_size + ( number_of_Service_id(i) * number_of_EIT_interval (EIT_category (i)) ) ;
    }
EIT_Insertion_step = MIN ( EIT_sub_interval (EIT_category (i)) );
EIT_Nb_insertion_step = ( EIT_insertion_window_duration / EIT_Insertion_step );
for (l=0; l < EIT_Nb_insertion_step; l++) {
    Nb_EIT_slots_to_spread = 0;
    for (m=0; m < EIT_slots_size; m++) {
        if ( (l * EIT_insertion_step)  =< EIT_offset(EIT_slots(m))  < ((l+1)* EIT_insertion_step) ) {
            Nb_EIT_slots_to_spread ++;
        }
    }
    for (n=0; n < Nb_EIT_slots_to_spread; n++) {
        for (m=0; m < EIT_slots_size; m++) {
            if ( (l * EIT_insertion_step)  =< EIT_offset(EIT_slots(m))  < ((l+1)* EIT_insertion_step) ) {
                EIT_offset(EIT_slots(m))  = (l* EIT_insertion_step )+ ((EIT_insertion_step / Nb_EIT_slots_to_spread) * n);
            }
        }
    }
}
```

The five steps of the described algorithm create an ordered list of EIT_slots which will be filled with data (EIT_sections_group) extracted from various input (DTH) Transport Stream(s).

6.4.4.5.2 Phase (B): Creation of the EIT_sections database

The EIT_sections database shall be populated by ***Nb_records*** records as outline with figure 12, the EIT_sections_data being elements extracted from the input (DTH) Transport Stream(s).

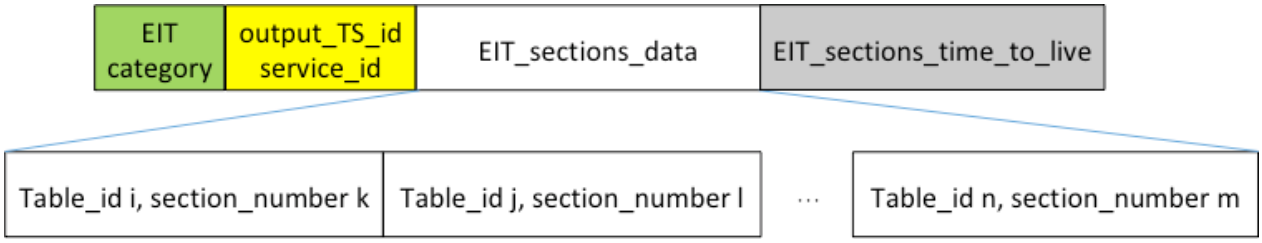

Figure 12: EIT_sections database entry

Where Nb_records is defined as follows:

```
Nb_records = 0;
for (i=0; i < number_of_Category; i++) {
    for (j=0; j < number_of_Service_id(i); j++) {
            Nb_records ++;
    }   }   }
```

According to the DVB standard, EIT sections are carried in four types of tables: Actual (0x4E) / Others (0x4F) for the Present/Following events and Actual (0x50..0x5F) / Others (0x60..0x6F) for the Scheduled events with various future horizons (i.e. Next day, Next Week, Next Month).

Eight categories of EIT_sections (EIT_Category) are defined as outlined below with figure 18 below, and recorded in the EIT_sections database.

Table 18: Categories of EIT_sections

CATEGORY NAME	EIT_CATEGORY	FROM	TO
EIT_cat1	EIT_pf_actual	table_id 0x4E, section number 0x00	table_id 0x4E, section number 0xFF
EIT_cat2	EIT_pf_other	table_id 0x4F, section number 0x00	table_id 0x4F, section number 0xFF
EIT_cat3	EIT_sch_1stday_actual	table_id 0x50, section number 0x00	table_id 0x50, section number 0x3F
EIT_cat4	EIT_sch_1stday_other	table_id 0x60, section number 0x00	table_id 0x60, section number 0x3F
EIT_cat5	EIT_sch_2nd_8_actual	table_id 0x50, section number 0x40	table_id 0x51, section number 0xFF
EIT_cat6	EIT_sch_2nd_8_other	table_id 0x60, section number 0x40	table_id 0x61, section number 0xFF
EIT_cat7	EIT_sch_>_8_actual	table_id 0x52, section number 0x00	table_id 0x5F, section number 0xFF
EIT_cat8	EIT_sch_>_8_other	table_id 0x62, section number 0x00	table_id 0x6F, section number 0xFF

Processes performed during the extraction of EIT_sections

For each output (DTT) multiplex, a DSACI configuration file specifies the EIT_insertion_window duration and the list of services to be extracted from various input (DTH) multiplexes to build up the “actual” output (DTT) multiplex. The DSACI configuration file refers also to the services delivered through “other” output (DTT) multiplexes having another output_TS_id and eventualy, another Original Network Identifier (ONID).

For a given output TS, the DSACI defines the TS_id, the ONID and the services to be extracted.

Extraction Process related to the “Actual” output (DTT) multiplex

The services are selected using the input service_id and they are associated to an output service_id, an eit_pf_flag and an eit_sch_flag. The eit_pf_flag indicates if the incoming EIT_present has to be present in the output stream. The eit_sch_flag indicates if the incoming EIT_schedule has to be present in the output stream.

The DSA parses all the incoming streams, it filters the EIT sections referencing the service defined in DSACI using the input service_id, the eit_sch_flag and eit_pf_flag. The fields TS_id, ONID, service_id are updated accordingly.

After processing, for each service, the EIT sections are concatenated according their type (i.e. present/following or schedule) and when completed, the concatenated EIT sections are associated with the arrival of the last packet of the last section of the table, called Time-to-Live. The record thus formed shall be stored in the EIT_sections database of “odd” categories (i.e. EIT_cat1, EIT_cat3, EIT_cat5, EIT_cat7) related to services delivered through the “actual” output (DTT) multiplex ..

Extraction Process related to the “Others” output (DTT) multiplexes

The label **eit_crossreferencing** configures the cross_referencing management, in the case of DVB-T2. When the parameter **eit_crossreferencing** is set to 1, the EIT tables carried in each PLP shall describe the services present in the other TS stream carried in other PLPs. The DSA uses the description of the other transport stream contained in the DSACI to obtain the EIT referencing the other TS of the DVB-T2 configuration.

The DSA parses all the incoming streams, the DSA filters the EIT sections referencing the service defined in DSACI using the input service_id, the eit_sch_flag and eit_pf_flag. The DSA changes the table_id, table_id defining actual table are changed to table_id defining other table. The fields TS_id, ONID, service are updated if necessary.

The DTT network should contain several multiplexes and the referencing of the different multiplex should be required. The parameter DSACI_id, DSACI_PID and source_id allows to retrieve the DSACI which configures the multiplex. The processing is identical to eit_cross_referencing.

After processing, for each service, the EIT sections are concatenated according their type (i.e. present/following or schedule) and when completed, the concatenated EIT sections are then associated with the arrival of the last packet of the last section of the table, called Time-to-Live. The record thus formed shall be stored in the EIT_sections database of “even” categories (i.e. EIT_cat2, EIT_cat4, EIT_cat6, EIT_cat8) related to services delivered through the “others” output (DTT) multiplex.

Management of the EIT_sections database

Two mechanisms are implemented to manage the EIT_sections database, one for the EIT present/following tables (i.e. EIT_cat1 and EIT_cat2), the other to manage the EIT schedule tables (i.e. EIT_cat3 to EIT_cat8).

When, for a given service, a complete EIT present/following table is extracted from the incoming stream, all the concatened EIT_sections with the associated Time-to-Live are immedialety stored in the EIT_sections database according to their category and the service they refer to (i.e. service_id). It shall replace the one existing in the EIT database. The corresponding record is then updated.

When, for a given service, a complete EIT schedule table is extracted from the incoming stream, at the start of the next EIT_insertion_window, all the concatened EIT_sections with the associated Time-to-Live are splited in three EIT_categories (i.e. in categories EIT_cat3, EIT_cat5, EIT_cat7 for an EIT schedule actual and in categories EIT_cat4, EIT_cat6, EIT_cat8 for an EIT schedule other) and stored in the EIT_sections database according to their category and the service they refer to (i.e. service_id). They replace the one existing in the EIT database.

The Time-to-Live for a record in the database is equal to two repetition periods of the type whose the group_of_sections bellows. In other words, when the difference between the current PCRabs and the arrival time of the current record is greater than two repetition period of the type whose the record bellows, the record shall be removed from the database.

6.4.4.5.3 Phase (C): Playout of the EIT_slots list

All elements in these arrays are time-stamped with a "Arrival Time-Stamp". In the case of the filtered EIT_sections, the timestamp is a "Virtual Arrival Time-stamp" (VAT). The "VAT" computation of EIT TS packet is described bellow.

The VAT for each EIT_slot is computed using the EIT_offset defined for each EIT_slot and the number of time the insertion_window that has been repeated since the SIS epoch.

The first EIT_insertion_window starts at SIS epoch, when output_timestamp(n-1) < VAT(EIT_slot) < output_timestamp (n), there is an opportunity to output a record associated to the EIT_slot. When the record is present in the EIT_section database, all the sections of ***EIT_section_data*** of the record shall be tight-packed to a series_of_TSpackets and then, the VAT of this series shall be computed as follows:

$$VAT(packet(j)) = VAT(EIT\_slot\ (i)) + (((VAT(EIT\_slot(i+1)) - VAT(EIT\_slot\ (i))) / series\_of\_TSpackets) * j)$$

Where j = 0 to series_of_TSpackets-1

The first TS packet carrying a EIT pid of the EIT_insertion_window shall have a CC=0. The CC of the following EIT TS packet is incremented by one.

For each EIT_insertion_window, the DSA computes the number of EIT TS packets inserted, a multiple number of 16 TS packets shall be inserted during a EIT_insertion_window. Thus, at the last EIT_slot, the DSA computes the number of EIT TS packet of stuffing section to insert to reach a multiple number of 16 TS packets. The DSA computes the VAT of the EIT TS packet of stuffing as a standard EIT TS packet.

6.4.4.5.4 Parameter definitions:

EIT_category: An EIT_category regroups all the sections associated to a given type (actual or other) and a given horizon, the different EIT_categories are defined in figure 18.

EIT_slot: a time interval associated to a given service for a given EIT_category

EIT_sections: EIT table sections

EIT_insertion_window_duration: The duration of EIT insertion window

EIT_repetition_rate (EIT_category (i)): Repetition rate for EIT category i

number_of_EIT_interval (EIT_category (i)): Number of intervals for EIT_category i ,

number_of_service (EIT_category (i)): Number of filtered service for EIT_category i

EIT_sub_interval (EIT_category (i)): Time slot duration for the insertion of EIT table for a service for an EIT category

EIT_slots_size: Number of EIT_slot in the EIT_insertion_window

EIT_offset: Time offset from the beginning of the EIT insertion window

EIT_Insertion_step: The minimum duration of EIT_sub_interval

EIT_Nb_insertion_step: The number of segments (EIT_Insertion_step),

Nb_EIT_slots_to_spread: The number of EIT_slots to spread in a given segment

6.4.5 Placement of incoming packets in the outgoing TS

6.4.5.1 Overview

The deterministic scheduling of incoming packets is based on the arrival timestamps computed at the input, in conjunction with the departure timestamps from the Reference TS generation. Incoming packets - and locally DSA-generated packets (belonging to PSI/SI tables regenerated by the DSA), which are equipped with Virtual Arrival Timestamps - replace null packets inside the Reference TS.

6.4.5.2 Deterministic scheduling

Scheduling is based on a first-in first-out process, with additional rules to solve packet collisions. The following rules apply for selecting an input TS packet for insertion into an output TS performed by the deterministic TS remultiplexer:

- An input TS packet that has to be played out replaces the first null packet of the target Reference Transport Stream that has a departure timestamp greater or equal to arrival timestamp of the input TS packet.
- When multiple TS packets (from single or multiple inputs) target the same null packet in the Reference Transport Stream, the input TS packet with the earliest arrival timestamp is chosen.
- When multiple TS packets with the same arrival timestamp (from multiple inputs) target the same null packet in the Reference TS, the packet with the greatest output PID is selected first.

All incoming packets not selected for play-out during the scheduling round described above shall join the set of packets candidate to the following scheduling round. A packet is removed from such set in case one of the following situations occur:

1. The TS packet is selected for insertion into the Reference TS on the basis of the mechanism described for the previous scheduling round in a following scheduling round (successful operation).
2. The packet is not selected after being candidate for $N_{steps\_to\_live}$ scheduling steps (overflow condition).

NOTE: $N_{steps\_to\_live}$ scheduling is a value from the DSACI (see clause 5.7).

As a consequence of the scheduling operation, an input TS packet that holds a Program Clock Reference for a service, shall have its PCR value updated according to the following equation:

$$PCR_{output} = PCR_{input} + (packet\_departure\_time - packet\_arrival\_time)$$

6.5 Framing

6.5.1 DVB-T2 case

6.5.1.1 TS splitting (optional)

The PLP_GROUP_ID and PLP_TYPE fields of the configurable L1-post signalling – see EN 302 755 [2] – shall be used to determine whether a data PLP belongs to a PLP group. Each group of PLPs may contain one common PLP. If a common PLPs belongs to a PLP group, the TS Splitting model defined in Annex D of EN 302 755 [2] shall be applied to the PLPs of the group in order to generate a TSPS for each data PLP together with a TSPSC for the common PLP.

NOTE: Annex D of the DVB-T2 standard EN 302 755 [2] defines a prescriptive and deterministic process for splitting and merging.

6.5.1.2 Partial Mode Adaptation

Mode Adaptation shall be performed on the TS (or TSPS) for a given PLP that was generated as described in clause 6.4 (if common PLPs are used, see also clause 6.5.1.1 and EN 302 755 [2], Annex D). The partial Mode Adaptation process shall comprise:

- SYNC byte removal
- Dummy ISSY insertion (if applicable)
- Dummy DNP insertion (if applicable)
- Dummy CRC-8 insertion (if applicable)

The Null Packet Deletion process shall not be performed at this stage. The dummy ISSY, DNP and CRC-8 bytes are inserted to ensure correct allocation to the Interleaving Frames (see clause 6.5.1.3). If NPD is used, the actual value of these bits will be calculated later when the Null Packet Deletion is performed (see clause 6.5.1.4) and at this stage they may be set to zero. Similarly, ISSY will be generated once the framing has been finalised (see clause 6.5.1.5).

6.5.1.3 Allocation of TS bits to Interleaving Frames

The bits of the partially Mode-Adapted Transport Stream (see clause 6.5.1.2) shall be allocated to Interleaving Frames according to the signalling in the T2-MI packet of type $F0_{16}$ (F&TI) extracted in clause6.6.

If Null Packet Deletion (NPD) is not enabled, the bits shall be allocated such that the first complete packet in the first BBFRAME of the Interleaving Frame is the one whose PCR_{abs} value is equal to the value of PCR_{FTSP} for the relevant Interleaving Frame and PLP, and such that the value of SYNCD signalled in the first BBFRAME of the Interleaving Frame is equal to FIRST_SYNCD for the relevant Interleaving Frame and PLP.

If Null Packet Deletion is enabled, the FIRST_SYNCD and PCR_{FTSP} fields indicate the PCR_{abs} for the first complete packet and the SYNCD value for the first BBFRAME under the conditions that neither (a) the packet indicated by PCR_{FTSP} nor (b) the packet immediately preceding it are deleted by the NPD process. If condition (a) does not apply, then the first complete packet will have $PCR_{abs} > PCR_{FTSP}$; if condition (b) does not apply, then SYNCD will be zero.

The description above identifies the first bit allocated to a given Interleaving Frame. The last bit allocated to an Interleaving Frame shall simply be the bit preceding the first bit of the subsequent Interleaving Frame.

6.5.1.4 Null Packet Deletion

If enabled, Null Packet Deletion shall now be performed. The DSA shall perform the following process:

- The last complete packet of an Interleaving Frame shall always be transmitted, i.e. it shall not be deleted
- Any other null packet shall be deleted unless 255 previous null packets have already been deleted, in which case it shall be transmitted together with a DNP count of 255.
- The dummy DNP counts inserted in the Partial Mode Adaptation process (clause 6.5.1.2) shall be replaced by the correct DNP counts
- The dummy CRC-8 values inserted in the Partial Mode Adaptation process (clause 6.5.1.2) shall be replaced by the correct CRC-8 values.

6.5.1.5 ISSY generation

The three ISSY variables are inserted according to EN 302 755 [2].

When BUFS is inserted, it shall have the value indicated by BUFS in the relevant F&TI T2-MI packet (see clause 5.6.2) for the relevant PLP. When $ISCR_{long}$ is used, the third byte shall be coded as defined in EN 302 755 [2].

TTO is inserted once per Interleaving Frame and applies to the first complete packet carried in the Interleaving Frame. The TTO_E, TTO_M and (if present) TTO_L fields shall have the values indicated in the fields of the same name in the relevant F&T T2-MI packet applying to this Interleaving Frame and PLP (see clause 5.6.2).

When ISCR is sent, its value shall be calculated as follows:

$$ISCR(p) = FIRST\_ISCR(n) + \left\lfloor p \frac{DPCR_{IF}(n)}{f_{PCR} \times T \times frame\_packet\_count(n)} \right\rfloor$$

Where $FIRST\_ISCR(n)$, $DPCR_{IF}(n)$ and $frame\_packet\_count(n)$ are the values of FIRST_ISCR, $DPCR_{IF}$ and frame_packet_count respectively signalled in the F&TI T2-MI packet for this PLP and for Interleaving Frame n, and p is the index within the Interleaving Frame of the packet to which the ISCR applies, $f_{PCR} = 27\ MHz$ and T is the elementary period as defined in clause 9.5 of EN 302 755 [2].

The packet whose PCR_{abs} value is equal to the value of PCR_{FTSP} for Interleaving Frame n shall have $p = 0$, and p shall increase by 1 for each subsequent packet.

NOTE: T is defined in µs, see sub-clause 9.5 of EN 302 755 [2].

6.5.1.6 Generation of BBFRAMEs

6.5.1.6.1 Overview

The bits allocated to an Interleaving Frame shall be mapped onto the number of BBFRAMEs indicated in the dynamic L1 signalling for the relevant PLP and Interleaving Frame. The relevant L1 signalling will be contained in an L1-current T2-MI packet extracted as described in clause 6.6.

The number of bits allocated to an Interleaving Frame may be less than the total capacity of the BBFRAMES, therefore BBFRAME padding may be performed. Padding may also be used for In-Band Signalling.

6.5.1.6.2 Allocation of bits to the BBFRAMEs of the Interleaving Frame

In the case of not applying NPD, the number of payload bits in the Interleaving Frame number *n* is

$$N_{payloadIF}(n) = FIRST\_SYNCD(n) + frame\_pkt\_count(n) \times UPL - FIRST\_SYNCD(n+1)$$

FIRST_SYNCD(n) is signalled by the FIRST_SYNCD field of the Framing & Timing Information for Interleaving Frame n. FIRST_SYNCD(n+1) is identical to the NEXT_FIRST_SYNCD field for Interleaving Frame n.

If NPD is enabled, the number of bits that were allocated to the Interleaving Frame as described in clause 6.5.1.3 but which refers to the deleted null packets shall be subtracted from $N_{payloadIF}(n)$ before performing the following calculations. The resulting number of bits is given by:

$$N_{payloadIF}(n) = N_1 + frame\_pkt\_count(n) \times UPL - N_D \times UPL - N_2$$

where:

$N_1 = 0$ if the packet whose PCR_{abs} value is equal to the value of PCR_{FTSP} for Interleaving Frame n is deleted; otherwise $N_1 = FIRST\_SYNCD(n)$.

$N_2 = UPL$ if if the packet whose PCR_{abs} value is equal to the value of PCR_{FTSP} for Interleaving Frame *n*+1 is deleted; otherwise $N_1 = FIRST\_SYNCD(n+1)$.

The resulting number of bits shall be allocated to the BBFRAMES of the Interleaving Frame such that the resulting DFL values are as follows:

The DFL for the first BBFRAME of the Interleaving Frame shall be given by the FIRST_DFL field of the Framing & Timing Information.

NOTE: The value of FIRST_DFL may be used to allow additional space for insertion of in-band signalling by the modulator.

The number of bytes of payload to be spread over the remaining BBFRAMEs is therefore given by:

$$N_{bytesIF} = \frac{N_{payloadIF} - FIRST\_DFL}{8}$$

These bytes shall be distributed over the remaining BBFRAMEs such that the DFL for each BBFRAME is a multiple of 8, the DFLs between any two BBFRAMEs differ by at most 8 bits, and all the BBFRAMEs with the smaller value of DFL come at the beginning of the BBFRAME. The DFL value for BBFRAME i, where $0 \leq i < N_{blocks}$, is therefore given by:

$$DFL(i) = \begin{cases} FIRST\_DFL & i = 0 \\ \left\lfloor \frac{N_{bytesIF}}{(N_{blocks}-1)} \right\rfloor & 0 < i < N_{blocks} - \left(N_{bytesIF} \bmod (N_{blocks}-1)\right) \\ \left\lfloor \frac{N_{bytesIF}}{(N_{blocks}-1)} \right\rfloor + 1 & i \geq N_{blocks} - \left(N_{bytesIF} \bmod (N_{blocks}-1)\right) \end{cases}$$

6.5.1.6.3 Mapping of mode-adapted bits into the BBFRAMES

The bits of the mode-adapted TS shall be mapped into the Data Fields of the BBFRAMEs of the Interleaving Frame according to the DFL values of the BBHEADERs calculated as described in clause 6.5.1.6.2.

6.5.1.6.4 BBHEADER generation

BBHEADERs are part of each BBFRAME and are 10 bytes long. Depending on the mode used – Normal Mode (NM) or High Efficiency Mode (HEM), see EN 302 755 [2] – they consist of the parameters outlined below with figures 13 and 14, respectively. Those figures illustrate the two header formats. The related parameter definitions of EN 302 755 [2] apply for this specification in the same way – in addition to the definitions below.

MATYPE (2 Bytes)	UPL (2 Bytes)	DFL (2 Bytes)	SYNC (1 Byte)	SYNCD (2 Bytes)	CRC-8 MODE (1 Byte)

Figure 13: BBHEADER format (Normal Mode)

MATYPE (2 Bytes)	ISSY 2 MSBs (2 Bytes)	DFL (2 Bytes)	ISSY 1 LSB (1 Byte)	SYNCD (2 Bytes)	CRC-8 MODE (1 Byte)

Figure 14: BBHEADER format (High Efficiency Mode)

When Baseband Frames are built by the DSA, the BBHEADER parameters are derived from the sources indicated below. The indication of the mode itself, i.e. NM or HEM, is part of the Framing & Timing Information, see clause 5.6.

Common set of BBHEADER parameters:

MATYPE: Always derived from F&TI, see clause 5.6.2.

DFL: The DFL field shall contain the value of DFL for the corresponding BBFRAME as specified in clause 6.5.1.6.2.

SYNCD: The SYNCD value of the first BBFRAME of the Interleaving Frame for the PLP shall be given by the parameter FIRST_SYNCD provided with the F&T Information, see clause 5.6.2. The setting of SYNCD for all remaining BBFRAMEs of the same Interleaving Frame

shall be calculated according to the alignment of packets within the BBFRAME as specified in EN 302 755 [2].

NOTE: The SYNCD value for BBFRAME i will be given in terms of the previous SYNCD, previous DFL and UPL by the following formula:

$$\mathrm{SYNCD}(i) = \{\mathrm{SYNCD}(i-1) - \mathrm{DFL}(i-1)\} \bmod \mathrm{UPL}$$

CRC-8: To be calculated in the same way as defined by EN 302 755 [2].

MODE: Always derived from Framing & Timing Information, see clause 5.6.2.

BBHEADER parameters specific to Normal Mode:

UPL: Since this specification covers the Transport Stream as the only layer 2 protocol, UPL (the transmitted user packet length) shall initially be set to 187_{10} (the SYNC byte is removed from the TS packets and carried in the BBHEADER). Under the described conditions, the following modifications are applicable:

- If input stream synchronization is to be used: UPL increased by 16_{10} or 24_{10} bits according to ISSY field length; ISSY field appended after each UP. Either the short or long format of ISSY may be used.
- Optional null-packet deletion; DNP computation and storage after the next transmitted UP; UPL increased by 8_{10}.
- CRC-8 computation at UP level; CRC-8 storage after the UP; UPL increased by 8_{10}.

SYNC: Since this specification covers the Transport Stream as the only layer 2 protocol, this parameter is always set to 47_{16}. For further details see EN 302 755 [2].

BBHEADER parameters specific to High Efficiency Mode:

ISSY: Generated as defined in clause 6.5.1.5.

6.5.2 DVB-T case

As explained in clause 6.4.2.3 above, the DVB-T-related mega-framing takes place as part of the TS re-multiplexing stage – between the Reference Transport Stream generator and the TS re-multiplexer. Between the TS re-multiplexer and the DSA output interface L no further processing is applied to the stream of DVB-T mega-frames.

6.6 Extraction of T2-MI packets

T2-MI packets (see TS 102 773 [1]) are received by the DSA in encapsulated form, i.e. embedded in TS packets compliant to the data piping method specified in TS 102 773 [1]. All T2-MI packets are decapsulated from the aforementioned TS packets – should their PID identify a T2-MI stream whose content is required for configuring the T2-MI stream that the DSA provides at interface L. This step is

needed for the identification of the T2-MI packet types. Depending on the T2-MI packet type, the DSA applies a different processing as follows:

- T2-MI packets of type 00_{16} (Baseband Frame) are not permitted for SIS usage and are dropped by the DSA should they occur
- T2-MI packets of packet types not equal to 00_{16} (Baseband Frame), 10_{16} (L1-current) or $F0_{16}$ (F&TI) are simply piped through and are encapsulated again in TS packets as specified in TS 102 773 [1] for insertion into the output T2-MI stream
- T2-MI packets of packet types 10_{16} are used for configuring the DSA output stream and are in the end encapsulated again in TS packets as mentioned above for the previous mode of operation
- T2-MI packets of type $F0_{16}$ (F&TI) are also used for configuring the DSA output stream, but are discarded after processing and do not become part of the DSA output stream

6.7 Output processing

6.7.1 DVB-T2 case

6.7.1.1 T2-MI multiplexing

When addressing DVB-T2 modulators with a T2-MI stream, the DSA inserts the L1 signalling – received via one of its inputs – into the output T2-MI stream. All T2-MI packet types – apart from types 00_{16} and $F0_{16}$ – shall be inserted unmodified into the output T2-MI stream. The order of T2-MI packets at the related input of the DSA shall be kept at its output. The T2-MI packets of type 00_{16} – consisting of Baseband Frames produced by the DSA – are inserted in the order that TS 102 773 [1] defines. The related T2-frames and -super-frames are identified by header parameters of the T2-MI packets received, see TS 102 773 [1]. The T2-MI header packet_count and CRC shall be updated in order to comply with TS 102 773 [1].

6.7.1.2 Transport of T2-MI packets in MPEG-TS

The encapsulation of the T2-MI packets into TS packets shall follow the data piping mechanism described in TS 102 773 [1]. The overall TS bitrate, the PID value of the encapsulated T2-MI stream, and the presence of PCR may follow the related DSACI parameter settings, but other user-defined values may be used.

For carriage over a managed distribution network to the modulator, the PSI/SI parameters may take the values defined by DSACI or other alternative user defined values.

Annex A (normative):

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

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="DSACI">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="global_configuration" type="type_global_configuration" />
            <xs:element name="input_configuration" type="type_input_configuration" />
            <xs:element name="remultiplexing" type="type_remultiplexing"/>
            <xs:element name="output_processing" type="type_output_processing"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>


<!-- type_global_configuration -->
   <xs:complexType name="type_global_configuration">
      <xs:sequence>
         <xs:element name="current_DSA_group_id" type="xs:int"/>
         <xs:element name="global_version_number" type="type_version_number"/>
         <xs:element name="global_application_time" type="xs:int"/>
         <xs:element name="SIS_edition" type="type_SIS_edition"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_SIS_edition -->
   <xs:complexType name="type_SIS_edition">
      <xs:sequence>
         <xs:element name="major" type="xs:int"/>
         <xs:element name="middle" type="xs:int"/>
         <xs:element name="minor" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>
```

```
<!-- type_input_configuration -->
   <xs:complexType name="type_input_configuration">
      <xs:sequence>
         <xs:element maxOccurs="unbounded" minOccurs="1" name="input" type="type_input_parent" />
      </xs:sequence>
   </xs:complexType>

<!-- type_input_parent: number of instances equals number of defined inputs -->
   <xs:complexType name="type_input_parent">
      <xs:sequence>
         <xs:element name="input_TS_id" type="xs:int"/>
         <xs:element name="input_ON_id" type="xs:int"/>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="PMT_PID_SIS_service" type="type_pid"/>
         <xs:element name="Primary_SIS_Service_Flag" type="xs:boolean"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_remultiplexing -->
   <xs:complexType name="type_remultiplexing">
      <xs:sequence>
         <xs:element name="output_TS" type= "type_output_TS" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_output_TS: number of instances equals number of defined output TS -->
   <xs:complexType name="type_output_TS">
      <xs:sequence>
         <xs:element name="PLP_id" minOccurs="0" maxOccurs="1" type="xs:int"/>
         <xs:element name="output_TS_id" type="xs:int"/>
         <xs:element name="output_ON_id" type="xs:int"/>
         <xs:element name="pid_processing" type= "type_pid_processing"/>
         <xs:element name="service_pmt_processing" type="type_service_pmt_processing"/>
         <xs:element name="psisi_processing" type="type_psisi_processing_mode"/>
         <xs:element name="Nsteps_to_live" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_pid_processing -->
   <xs:complexType name="type_pid_processing">
      <xs:sequence>
```

```
      <xs:element name="pid" type="type_pid_proc" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

<!-- type_pid_proc: number of instances equals number of defined output pid -->
  <xs:complexType name="type_pid_proc">
    <xs:sequence>
      <xs:element name="source_id" type="xs:int"/>
      <xs:element name="input_PID" type="type_pid"/>
      <xs:element name="output_PID" type="type_pid"/>
    </xs:sequence>
  </xs:complexType>

<!-- type_service_pmt_processing -->
  <xs:complexType  name="type_service_pmt_processing">
    <xs:sequence>
      <xs:element name="service" type="type_service" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

<!-- type_service: number of instances equals number of defined service in an output TS -->
  <xs:complexType name="type_service">
    <xs:sequence>
      <xs:element name="source_id" type="xs:int"/>
      <xs:element name="input_service_id" type="xs:int"/>
      <xs:element name="output_service_id" type="xs:int"/>
      <xs:element name="output_service_name" type="xs:string"/>
      <xs:element name="output_provider_name" type="xs:string"/>
      <xs:element name="eit_schedule_flag" type="xs:boolean"/>
      <xs:element name="eit_present_following_flag" type="xs:boolean"/>
      <xs:element name="running_status" type="xs:int"/>
      <xs:element name="free_ca_mode" type="xs:int"/>
      <xs:element name="output_PMT_PID" type="type_pid"/>
      <xs:element name="pmt_processing_mode" type="type_pmt_processing_mode"/>
    </xs:sequence>
  </xs:complexType>

<!-- type_pmt_processing_mode -->
  <xs:complexType name="type_pmt_processing_mode">
    <xs:choice>
      <xs:element name="pmt_passthrough" type="pmt_passthrough_processing"/>
```

```
        <xs:element name="pmt_patching" type="pmt_patching_processing"/>
        <xs:element name="pmt_regeneration" type="pmt_regeneration_processing"/>
      </xs:choice>
    </xs:complexType>

<!-- pmt_passthrough_processing -->
    <xs:simpleType name="pmt_passthrough_processing">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>

<!-- pmt_patching_processing -->
    <xs:complexType name="pmt_patching_processing">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="ECM" type="type_ECM"/>
      </xs:sequence>
    </xs:complexType>

<!-- pmt_regeneration_processing -->
    <xs:complexType name="pmt_regeneration_processing">
      <xs:sequence>
        <xs:element name="table_repetition_period" type="xs:int"/>
        <xs:element name="offset" type="xs:int"/>
        <xs:element name="PCR_PID" type="type_pid"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="output_pid" type="type_pid"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="ECM" type="type_ECM"/>
      </xs:sequence>
    </xs:complexType>

<!-- type_ECM -->
    <xs:complexType name="type_ECM">
      <xs:sequence>
        <xs:element name="CAS_id" type="xs:int"/>
        <xs:element name="output_ECM_PID" type="type_pid"/>
      </xs:sequence>
    </xs:complexType>

<!-- type_psisi_processing_mode -->
    <xs:complexType name="type_psisi_processing_mode">
      <xs:sequence>
        <xs:element name="pat" type="type_pat_processing_mode"/>
        <xs:element name="cat" type="type_cat_processing_mode"/>
```

```
        <xs:element name="sdt_bat" type="type_sdt_bat_processing_mode"/>
        <xs:element name="eit" type="type_eit_processing_mode"/>
      </xs:sequence>
    </xs:complexType>

<!-- type_pat_processing_mode -->
    <xs:complexType name="type_pat_processing_mode">
      <xs:choice>
        <xs:element name="pat_passthrough" type="pat_passthrough_processing"/>
        <xs:element name="pat_patching" type="pat_patching_processing"/>
        <xs:element name="pat_regeneration" type="pat_regeneration_processing"/>
      </xs:choice>
    </xs:complexType>

<!-- pat_passthrough_processing -->
    <xs:simpleType name="pat_passthrough_processing">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>

<!-- pat_patching_processing -->
    <xs:simpleType name="pat_patching_processing">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>

<!-- pat_regeneration_processing -->
    <xs:complexType name="pat_regeneration_processing">
      <xs:sequence>
        <xs:element name="table_repetition_period" type="xs:int"/>
        <xs:element name="offset" type="xs:int"/>
        <xs:element name="PAT_version_number" type="type_version_number"/>
      </xs:sequence>
    </xs:complexType>

<!-- type_cat_processing_mode -->
    <xs:complexType name="type_cat_processing_mode">
      <xs:choice>
        <xs:element name="cat_stopping" type="cat_stopping_processing"/>
        <xs:element name="cat_passthrough" type="cat_passthrough_processing"/>
        <xs:element name="cat_patching" type="cat_patching_processing"/>
        <xs:element name="cat_regeneration" type="cat_regeneration_processing"/>
      </xs:choice>
```

```
  </xs:complexType>

<!-- cat_stopping_processing -->
  <xs:simpleType name="cat_stopping_processing">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

<!-- cat_passthrough_processing -->
  <xs:simpleType name="cat_passthrough_processing">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

<!-- cat_patching_processing -->
  <xs:complexType name="cat_patching_processing">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="ca_provider" type="type_ca_provider"/>
    </xs:sequence>
  </xs:complexType>

<!-- cat_regeneration_processing -->
  <xs:complexType name="cat_regeneration_processing">
    <xs:sequence>
      <xs:element name="table_repetition_period" type="xs:int"/>
      <xs:element name="offset" type="xs:int"/>
      <xs:element name="CAT_version_number" type="type_version_number"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="ca_provider" type="type_ca_provider"/>
    </xs:sequence>
  </xs:complexType>

<!-- ca_provider: number of instances equals number of output EMM_PID's -->
  <xs:complexType name="type_ca_provider">
    <xs:sequence>
      <xs:element name="CAS_id" type="xs:int"/>
      <xs:element name="output_EMM_PID" type="type_pid"/>
    </xs:sequence>
  </xs:complexType>

<!-- type_sdt_processing_mode -->
  <xs:complexType name="type_sdt_bat_processing_mode">
    <xs:choice>
      <xs:element name="sdt_passthrough" type="sdt_passthrough_processing"/>
```

```
      <xs:element name="sdt_patching" type="sdt_patching_processing"/>
      <xs:element name="sdt_regeneration" type="sdt_regeneration_processing"/>
    </xs:choice>
  </xs:complexType>

<!-- sdt_passthrough_processing -->
  <xs:simpleType name="sdt_passthrough_processing">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

<!-- sdt_patching_processing -->
  <xs:complexType name="sdt_patching_processing">
    <xs:sequence>
      <xs:element name="sdt_crossreferencing_flag" type="xs:boolean"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="bouquet" type="type_bouquet"/>
    </xs:sequence>
  </xs:complexType>

<!-- sdt_regeneration_processing -->
  <xs:complexType name="sdt_regeneration_processing">
    <xs:sequence>
      <xs:element name="sdt_actual_period" type="xs:int"/>
      <xs:element name="M_actual" type="xs:int"/>
      <xs:element name="offset" type="xs:int"/>
      <xs:element name="sdt_actual_version_number" type="xs:int"/>
      <xs:element name="sdt_crossreferencing_flag" type="xs:boolean"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="bouquet" type="type_bouquet"/>
    </xs:sequence>
  </xs:complexType>

<!-- other_mux: number of instances equals number of defined other muxes -->
  <xs:complexType name="type_other_mux">
    <xs:sequence>
      <xs:element name="other_DSA_group_id" type="xs:int"/>
      <xs:element name="DSACI_PID" type="type_pid"/>
      <xs:element name="source_id" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
```

```
<!-- type_bouquet -->
   <xs:complexType name="type_bouquet">
      <xs:sequence>
         <xs:element name="source_id" type="xs:int"/>
         <xs:element name="input_bouquet_id" type="xs:int"/>
         <xs:element name="terrestrial_bouquet_id" type="xs:int"/>
      </xs:sequence>
   </xs:complexType>

<!-- type_eit_processing_mode -->
   <xs:complexType name="type_eit_processing_mode">
      <xs:choice>
         <xs:element name="eit_passthrough" type="eit_passthrough_processing"/>
         <xs:element name="eit_patching" type="eit_patching_processing"/>
         <xs:element name="eit_regeneration" type="eit_regeneration_processing"/>
      </xs:choice>
   </xs:complexType>

<!-- eit_passthrough_processing -->
   <xs:simpleType name="eit_passthrough_processing">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>

<!-- eit_patching_processing -->
   <xs:complexType name="eit_patching_processing">
      <xs:sequence>
         <xs:element name="eit_crossreferencing_flag" type="xs:boolean"/>
         <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
      </xs:sequence>
   </xs:complexType>

<!-- eit_regeneration_processing -->
   <xs:complexType name="eit_regeneration_processing">
      <xs:sequence>
         <xs:element name="eit_pf_actual_period" type="xs:int"/>
         <xs:element name="eit_pf_other_period" type="xs:int"/>
         <xs:element name="eit_sch_1stday_actual_period" type="xs:int"/>
         <xs:element name="eit_sch_1stday_other_period" type="xs:int"/>
         <xs:element name="eit_sch_2nd_8th_actual_period" type="xs:int"/>
         <xs:element name="eit_sch_2nd_8th_other_period" type="xs:int"/>
         <xs:element name="eit_sch_sup_8th_actual_period" type="xs:int"/>
```

```
        <xs:element name="eit_sch_sup_8th_other_period" type="xs:int"/>
        <xs:element name="eit_insertion_window_duration" type="xs:int"/>
        <xs:element name="eit_cross_referencing_flag" type="xs:boolean"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="other_mux" type="type_other_mux"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="DTT_only_service" type="type_DTT_only_service"/>
      </xs:sequence>
    </xs:complexType>

<!-- DTT_only_service: number of instances equals number of defined DTT only service -->
    <xs:complexType name="type_DTT_only_service">
      <xs:sequence>
        <xs:element name="DTT_only_service_id" type="xs:int"/>
        <xs:element name="input_EIT_PID" type="type_pid"/>
        <xs:element name="source_id" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>

<!-- type_output_processing -->
    <xs:complexType name="type_output_processing">
      <xs:sequence>
        <xs:element name="terrestrial_standard_generation" type="type_terrestrial_standard_generation" />
      </xs:sequence>
    </xs:complexType>

<!-- type_terrestrial_standard -->
    <xs:complexType name="type_terrestrial_standard_generation">
      <xs:choice>
        <xs:element name="dvb_t" type="dvb_t_standard"/>
        <xs:element name="dvb_t2" type="dvb_t2_standard"/>
      </xs:choice>
    </xs:complexType>

<!-- dvb_t_standard -->
    <xs:simpleType name="dvb_t_standard">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>

<!-- dvb_t2_standard -->
    <xs:complexType name="dvb_t2_standard">
      <xs:sequence>
        <xs:element name="output_T2_MI_PID" type="type_pid"/>
```

```
      <xs:element name="output_T2_MI_stream_id" type="xs:int"/>
      <xs:element name="output_rate" type="xs:int"/>
      <xs:element maxOccurs="1" minOccurs="0" name="L2_signalling" type="type_L2_signalling"/>
    </xs:sequence>
  </xs:complexType>

<!-- L2_signalling -->
  <xs:complexType name="type_L2_signalling">
    <xs:sequence>
      <xs:element name="output_TS_id" type="xs:int"/>
      <xs:element name="output_ON_id" type="xs:int"/>
      <xs:element name="output_L2_service_id" type="xs:int"/>
      <xs:element name="output_L2_service_provider_name" type="xs:string"/>
      <xs:element name="output_L2_service_name" type="xs:string"/>
      <xs:element name="output_L2_PCR_PID" type="type_pid"/>
      <xs:element name="output_L2_PMT_PID" type="type_pid"/>
    </xs:sequence>
  </xs:complexType>

<!-- Other types -->
  <xs:simpleType name="type_version_number">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="31"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="type_pid">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="8191"/>
    </xs:restriction>
  </xs:simpleType>
  </xs:schema>
```

Bibliography

ETSI EN 102 831: "Digital Video Broadcasting (DVB); Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)".

ETSI TS 101 211: Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)

History

Document history		
V1.1.1	June 2018	Publication