Harmonic's answer to DVB TM-T-JSI Single Illumination Call for Technologies (CfT)

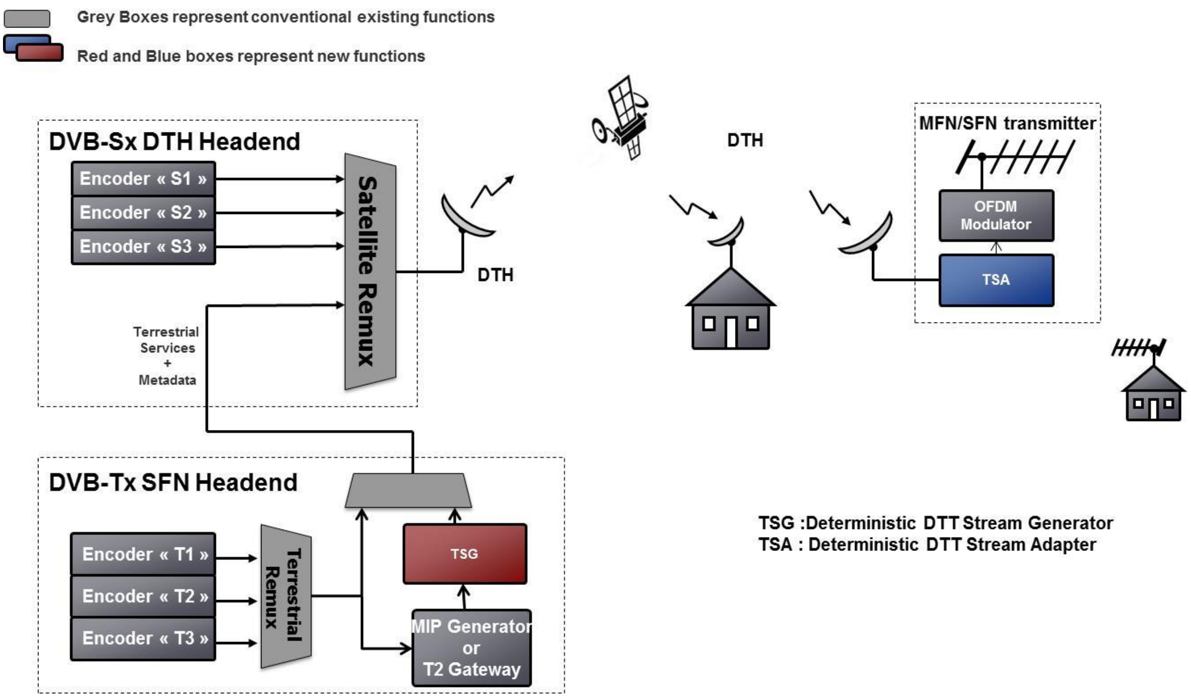
Version 1, June 17, 2016

Scope

This document is the Harmonic's answer to ''Single Illumination'' Call for Technologies (CfT) issued by DVB (TM-T-JSI group) on the 20th of May 2016 under the reference TM-T0034r1.

For a better understanding of the following parts of this document, a high level architecture diagram is given below. It highlights the location and connections to the rest of the system of the two new entities needed to satisfy the expressed commercial requirements.

High level architecture /Definition:



- The Deterministic DTT Stream Generator (TSG) is a new function or building block located in the Head-End that prepares the Metadata to be added in the DVB-Sy signal.
- The Deterministic DTT Stream Adapter (TSA) is a new function or building block that converts a DVB-Sy signal to a DVB-Ty signal by deterministic operation.

The following parts will explain the various operations made on the TSA function as well as the definition of the different metadata (using DVB technologies to transport them) produced by the TSG function and used by the TSA in order to ultimately provide a fully compliant stream to the terrestrial modulator.
The solution presented below relies on well proven ISO and DVB standardized technologies (DVB T2-MI, generic mechanism for table construction based on sections, …) and suggest some extensions on top of them to fulfil the commercial requirements.

I. Deterministic generation of re-multiplexed TSs from input TSs

A. Forewords

Re-multiplexing in the TSA is based on a reference stream of timestamped null TS-packets along with TS-packet election instructions deduced from configuration. The reference stream is locally built in the TSA from incoming metadata. Using the instructions, null packets of the reference stream are substituted with TS-packets from the DVB-Sy signal (or other input network). Input TSs packets can be used “as is”, PID-remapped, or processed.

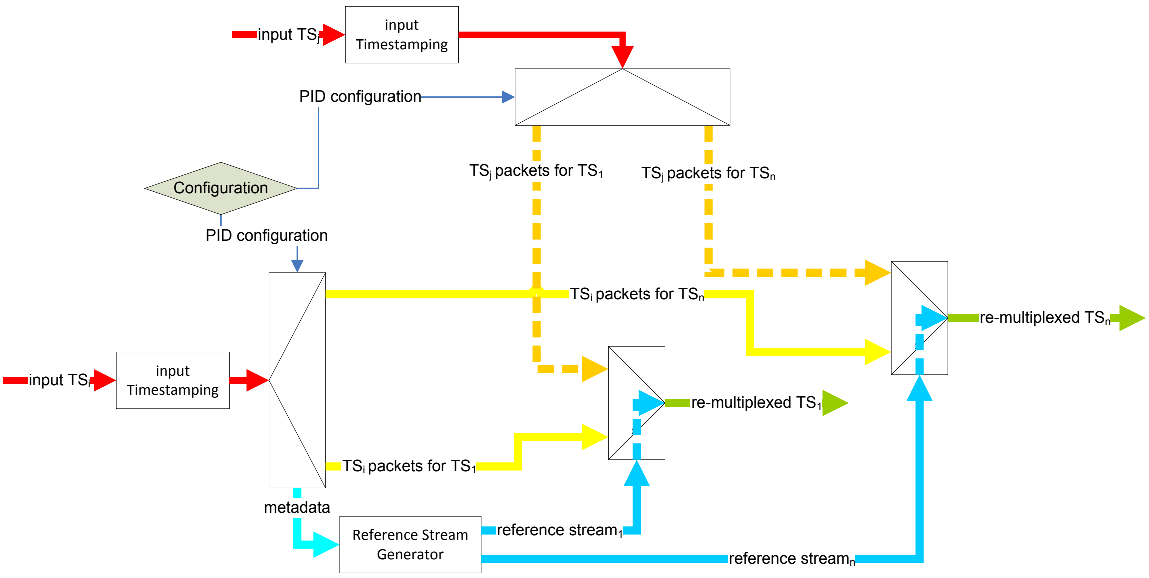


Figure 1 - re-mux TS from input TS

All input TSs arise from TSG(s) and reach the TSA through DVB-Sy network or other kind of transport network.

B. The Reference Stream(s)

The reference streams are generated in the TSA according to received metadata (see part III.A & III.B). In DVB-T or DVB-T2 input mode ‘A’ (single PLP), a single reference stream is required. In mode ‘B’ a reference stream per PLP is generated. A reference stream is a sequence of null transport stream packets at the re-multiplexed TS rate. Each packet of a reference stream holds a timestamp computed from metadata. These timestamps are used as “packet departure times” in the generation of output re-multiplexed TS.

The system clock used for reference stream timestamping is also used to timestamp packets from input TSs (see part I.C below). The delivery of these timestamps is described in parts III.A and III.B.

C. Input TS packet timestamping

Each input TS from TSG holds a P[illegible]omponent that carries PCR. This component, called *jsi_system_reference*, is generated and [illegible]rted by the TSG. The PCR_field follows the standard **[2]**, as a counter from a 27MHz clock. In the TSG, this 27MHz reference is synchronized on PPS. At 01/01/2000 00H00 GMT, the counter value was zero (as a convention), so that any GPS synchronized device can generate the same value at the same time. This ensures that input TSs from multiple TSGs hold a common clock. This clock is called *jsi_system_clock*

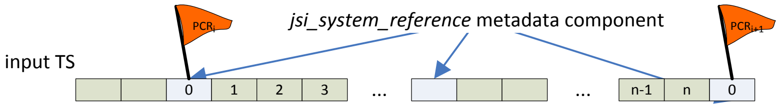


Figure 2 - input TS timestamping

Input TS timestamping consists in performing arithmetic operation to provide a timestamp for every packet: assuming two PCR values are received, The division $\frac{PCR_{i+1}-PCR_i}{n}$ and its remainder, are computed to provide an increment Δpcr_i^{i+1} that is used for extrapolation of PCR value for packet number 1..n, i.e. between two packet with PCR.

The resulting timestamping is used as "packet arrival time" for an input TS packet that is involved in a DVB-Ty re-multiplexed TS. It is assumed that the same deterministic operation is made on every TSA of the same SFN region, using the same input TS PCR.

D. TS-packet election

Based on input PID and received configuration (see part III.D), an input TS packet may be discarded or involved in a re-multiplexed TS. A packet may take part of the re-multiplexed TS "as is", or can have its PID remapped, or can have its content patched according to the component type and the configuration of the TSA.

All selected packets are kept with their output PID and the input timestamp in chronological order.

E. Deterministic scheduling

Deterministic scheduling makes decisions based on timestamp value related to packets. When involved, an input TS packet substitutes the first null packet of the target reference stream that has a timestamp greater or equal to its timestamp. If the input TS packet holds a PCR, its value is corrected by arithmetic operation, like in any classical multiplexing operation:

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

$packet\_arrival\_time$ is the input TS timestamp computed as described in part I.C while $packet\_departure\_time$ is timestamp of the substituted null packet in the reference stream (see part I.B).

Transport packets from any input stream are involved in the scheduling one after the other by their "arrival time" order.

In the case where several input TS packets - potentially coming from different TS having been transported on different networks - have timestamps such that they should replace the same null packet in the reference stream, then the input packet having the lowest timestamp will replace the above mentioned null packet and the other input TS packet will replace the following null packet in the reference stream.

If different input TS packets have exactly the same input timestamp, then the selection criteria to allocate only one input TS packet to one null packet is based on the output PID value: packets are sorted by their output PID as key, and one extremum (greatest or smallest) is chosen according to the computed even parity bit of the timestamp.

F. PSI/SI Processing

The TSA does not perform deterministic re-packetizing of PSI/SI table sections. Nevertheless, tables may be modified through PID remapping, and content patching inside TS-payload. Three strategies can be used by the TSG for PSI/SI management.

#1. The first assumption is that when a signalling component is identical in its description of input TS (DVB-Sy or any other network) elementary content versus DVB-Ty, it is transmitted "as is".

#2. Else, for tables located in a transport stream and that references this same transport stream, a version suitable for target terrestrial multiplex is inserted by the TSG on a dedicated ghost PID. According to its configuration, the TSA will only remap this PID to the final one.

#3. Additionally, table sections inside TS-packet may be invalidated using stuffing section and some attributes may be modified – see following Table.

Table 1 - list of alterable attributes and their location

attribute	table
table_id (actual->other)	SDT, NIT, EIT
service_id	EIT, PMT, PAT
transport_stream_id	SDT, NIT, EIT, PAT
original_network_id	SDT, NIT, EIT
program_map_PID	PAT
network_id	NIT
running_status	SDT, EIT
free_ca_mode	SDT, EIT
EIT_schedule_flag	SDT
EIT_present_following_flag	SDT
bouquet_id	BAT
elementary_PID	PMT
PCR_PID	PMT

Even if strategies 1 plus 2 can handle any PSI/SI update needed in DVB-Ty multiplex, it is TSG duty to select the 3rd strategy in order to save input TS bandwidth. The decision is taken at TSG level and the TSA performs the necessary steps, to follow the decision available through the configuration metadata.

Strategies are not mutually exclusive and apply equally to each PID table. A synthesis of SI/PSI processing by PID is provided in Annex C.

Multiple input networks management:

- Since TSA does not perform section rebuilding and re-packetization, only PMT and AIT tables are able to be managed according to strategy #1, #2 or #3.
- All other tables are systematically managed according to strategy #2, each table being transmitted in only one input TS. These tables are:
  - tables related to a whole DVB-Ty (e.g. PAT, SDT);

- tables whose sub-tables instances are related to items (services) transmitted with the same PID by different input TS (e.g. EIT). For each DVB-Ty, the approach consists in transmitting as ghost in one input TS all the sub-tables related to the items transmitted in the different input TS.

- Configuration metadata provides each TSA with the input TS that carry sub-tables, possible PID remapping information, and other possible needed processing at TSA side (attributes modification) – see § III.D.1 below.

G. Scrambling

Scrambling feature concerns:

- PMT and CAT: CA Descriptor management;
- ECM and EMM broadcasting.

Configuration metadata (for input TS packets selection) provides the set of ECM and EMM PID transported in input TS to be transmitted in DVB-Ty.

Case #1: a service is scrambled by the same way in the input TS and in the DVB-Ty stream (i.e. no different set of CAS_id in input TS and DVB-Ty):

- PMT related to the input TS service is used "as is" for the DVB-Ty stream (as soon as there is no other PMT change, according to the configuration) – cf. strategy #1 described in F above.
- CAT related to the input TS service is also used "as is" for the DVB-Ty stream.
- All ECM and EMM packets at inserted in the output TS.

Case #2: a service is scrambled with at least one different CAS in the input TS and in the DVB-Ty stream: according to TSG strategy:

- PMT related to the input TS service is also used for the DVB-Ty stream after CA Descriptor invalidating by TSA. *Precision*: CA Descriptor invalidation is performed by using Stuffing Descriptor – cf. strategy #3 described in F above.

  Or: a specific terrestrial PMT is inserted in the input TS on a dedicated ghost PID and remapped by TSA – cf. strategy #2 described in F above.
- CAT from the input TS is also used for the DVB-Ty stream after CA Descriptor invalidating by TSA.

  Or: a specific terrestrial CAT is inserted in the input TS on a dedicated ghost PID and remapped by TSA.
- Only ECM and EMM packets corresponding to the kept CA types are inserted in the output TS.

Case #3: a service is scrambled in the input TS and in the clear in the DVB-Ty stream.

- This can be handled if TSA hosts the descrambling feature (TS packets carrying components are descrambled, PMT are unchanged).
- Terrestrial PMT and CAT are managed as in case #2.
- No ECM and EMM packets is transmitted in the output TS.

II. Deterministic generation of output T2-MIs/TMFs from re-multiplexed TSs and additional metadata

As seen in part I.B & I.C, the TSG insert a *jsi_system_reference* component for deterministic timestamping. This same TS component is also used to provide the structure of the framing structure of the DVB-Ty modulation.

A. Generation of TMFs

In DVB-T, the reference streams (see part I.B) are built from MIP packets inserted by the TSG. The number or packets per mega-frame is computed from MIP packets and timestamps are computed the same way as I.C, from additional metadata that give timestamp for the first packet of the mega-frame (see description in part III.A).

The reference stream is then populated with the incoming MIP packets at the position consistent with "pointer" attribute, and then input TS packets are scheduled. This deterministic generation of re-multiplexed TS (see part I) fully describes the TMF when multiplexing is done in such reference frame.

B. Generation of T2-MI

1. BB-frame generation

With the exception of BB-frames, all T2-MI packets types that are suitable for the T2 gateway configuration, are generated in the TSG. Then they are transmitted (on the satellite link or aside network) as part of the metadata on the *jsi_system_reference* TS component.

Instead of BB-frame a new type of T2-MI packet called here "BB-reference" is inserted by the TSG (see part III.B). The BB-reference provides mode adaptation instruction for deterministically building BBHEADERs with the help of L1-current signalling.

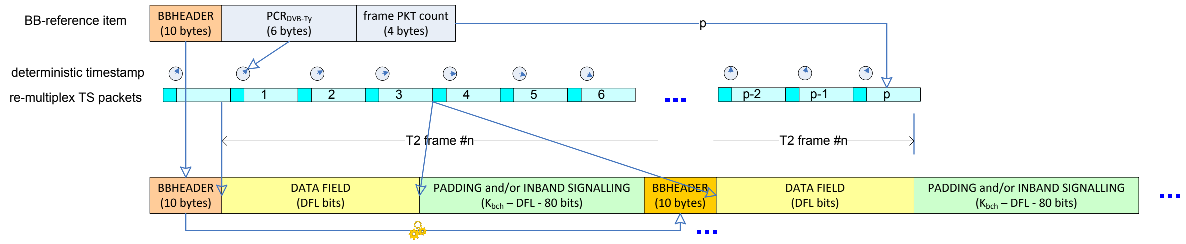


Figure 3 - BB-reference expansion

Using timestamping information available in the BB-reference, each re-multiplexed stream from input TS is split in packet chunks. For each target PLP, these chunks are mapped to the DATA FIELD of the BB-frames with same timestamp. BB-frame padding if any is deduced from rules (see part IV).

In the case where multiple input TS is used to build an output TS to be mapped in a BB-frames, only one input TS (built on the Master TSG) carries the BB-reference T2-MI packets encapsulated in MPEG-2 TS.

2. T2-MI packet transmission

The TSG implements a modified T2 gateway that produces all but T2-MI BB-frame packets. It is configured as required by the final DVB-T2 signal. BB-frames structures are encoded in T2-MI BB-reference packets and transmitted in place of T2-MI BB-frame packets.

The T2-MI packet type order built in the TSG is kept in TSA processing. The T2-MI BB-reference packets are removed, and expanded T2-MI BB-frame packets take their place. The whole T2-MI packet sequence has its packet_count renumbered according to **[1]**. This numbering made on the various TSA on the field does not have to be deterministic as this piece of information is not over-air DVB-T2 signal. Non-deterministic data piping is used to re-encapsulate T2-MI packets in MPEG-2 TS

III. Metadata enabling the two aforementioned steps

Several kinds of metadata are described in this chapter. Some will provide the framing structure of the DVB-Ty modulation with additional timestamping information, some will provide the configuration of the TSA and the management of SI/PSI. The figure below depicts the metadata path for single DVB-Ty modulation.

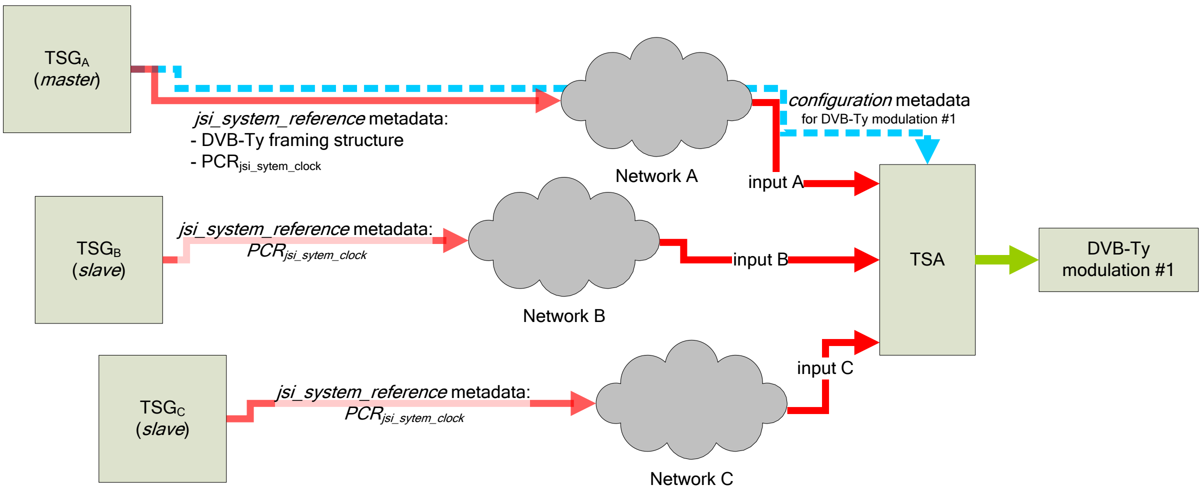


Figure 4 - metadata in JSI system

Note that the multiple TSGs do not play the exact same role. One is called “master” (for modulation #1) as it provides along with the services:

- DVB-Ty framing structure inside the *jsi_system_reference* component.
- TSA configuration, in (in-band) configuration metadata.

The “slave” TSGs provide additional services and the deterministic timestamping metadata inside the *jsi_system_reference* component.

A. Metadata for DVB-T reference streams reconstruction

1. Metadata syntax and semantics

In DVB-T, the TSG computes the MIP packets of the target DVB-Ty streams. The computed MIP packets fully define the number of packets per mega-frame. Additionally, the timestamp of first packet of each mega-frame is added to provide deterministic packet departure time for re-multiplexing. For this purpose a new function (chosen in “future_use” function_tag value range of **[3]**) is added in the function loop of the MIP packet.

Table 2: DVB-T jsi reference stream timestamping function

Syntax	Number of bits	Identifier
jsi_reference_stream_function() {		
function_tag	8	uimsbf
function_length	8	uimsbf
mega_frame_clock_reference_base	33	uimsbf
reserved	6	bslbf
mega_frame_clock_reference_extension	9	uimsbf

}		

mega_frame_clock_reference_base: PCR_base value that represents the reference timestamp of the first packet of the succeeding mega-frame.

mega_frame_clock_reference_extension: PCR_extension value that represents the reference timestamp of the first packet of the succeeding mega-frame.

2. Metadata delivery method

The TSG acts as a MIP inserter for each DVB-T modulation. It adds a jsi_reference_stream_function inside each computed MIP packet that are afterwards added to the DVB-Sy multiplex as *jsi_system_reference* metadata TS component.

B. Metadata for DVB-T2 reference streams reconstruction

1. Metadata syntax and semantics

As mentioned in part II.B.1 the BB-reference is used for BB-frame generation and reference stream building inside the TSA. A BB-reference describes each PLP of a T2-frame. A BB-reference item consists in the BBHEADER of first BB-frame followed by the PCR of the first transport stream packet that starts in the T2 frame. A packet count per frame is added.

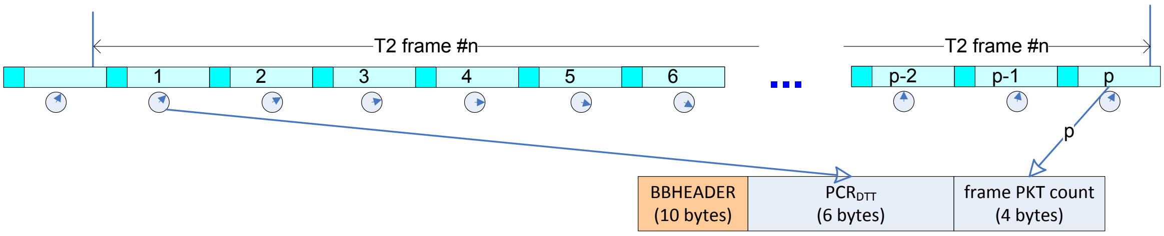


Figure 5 – BB-reference item for one PLP.

BBHEADER (80 bits): copy of the BBHEADER of the first baseband frame that appears for this PLP in the T2 frame (see **[4]**)

PCR_{DTT} (48 bits): PCR_field (see **[2]**) timestamp value in the jsi_system_clock for the first TS packet that starts in the T2 frame.

frame_pkt_count (32 bits): number of TS packet that start in the T2 frame.

For each frame, the items are grouped so that all PLPs in the T2-frame are described.

frame_idx (1 byte)	plp_id (1 byte)	lf_start & rfu (1 byte)	BBHEADER (10 bytes)	PCR_{DTT} (6 bytes)	frame PKT count (4 bytes)

x N_{plp} ...

Figure 6 - BB-reference aggregate

frame_idx (8 bits): frame_idx as defined in **[1]**.

plp_id (8 bits): PLP_ID as defined in **[1]**.

intl_frame_start (1 bit): as defined in **[1]**. The fact that the BBHEADER is from the first baseband frame implies this value is 1.

rfu (7 bits): 0_2 bits to follow definition of baseband frame in **[1]**.

All PLP descriptions aggregated are mapped inside the payload of new type $A0_{16}$ (for BB-Reference) of T2-MI packet. The existing standard T2-MI packet structure is re-used (see **[1]**).

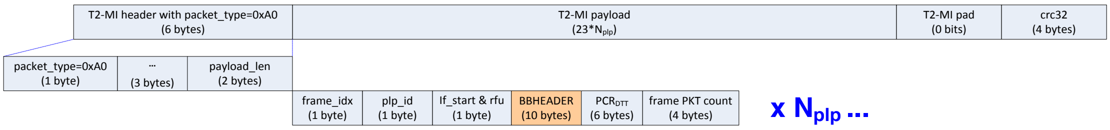


Figure 7 - T2-MI packet_type $A0_{16}$: Baseband reference

2. Metadata delivery method

This T2-MI packet is transported over MPEG-TS by data piping like any T2-MI packet (see **[1]**). The T2-MI packet order is unchanged compared to a standard T2 gateway, but BB-frames are replaced by this single T2-MI packet. The resulting component is added as *jsi_system_reference* TS component on dedicated PID by the TSG.

C. Metadata for input TSs timestamping

Input timestamping needs a dedicated PCR_PID per input TS (the *jsi_system_reference* TS component). As seen in I.C above, the TSGs provides coherent values among several TSA inputs in order to provide a consistent “packet_arrival_time”.

1. Metadata syntax and semantics

program_clock_reference_base; program_clock_reference_extension – the PCR field used for input timestamping follows the syntax from **[2]**. The system clock used for PCR encoded shall be synchronized to the PPS from a GPS receiver. The initial value is set to $PCR\_base = 0$, $PCR\_ext = 0$ at 2000-01-01T00:00:00Z. (see jsi_system_clock definition).

2. Metadata delivery method

The PCR_field is hold by the *jsi_system_reference* TS component, i.e:

- For DVB-T2, the PCR_field is delivered on the component that is used to transport the T2-MI packets over MPEG-2 TS.
- For DVB-T, the PCR_field is delivered in the DVB-Sy stream on the MIP component. The PCR_field shall be removed by the TSA before using MIP packet in DVB-T context.

Regarding bandwidth requirement for *jsi_system_reference* TS component (including DVB-Ty structure and *jsi_system_clock* delivery), and assuming standard PCR repetition rate, the result is typically less than 40 kbps whatever the terrestrial standard is used.

D. Configuration metadata, for input TS packet selection and SI/PSI processing.

The configuration metadata following description applies to a single DVB-Ty modulation system. When more than one DVB-Ty modulation is involved in the TSA processing, several configuration metadata set shall be provided to the TSA.

The configuration of the TSA can be made partially through metadata and through local/remote configuration GUI. If a part of the configuration is made by an operator is made, then it should be consistent with the configuration made on the TSGs. The consistency of the global TSA configuration when made by both metadata and User configuration is out of the scope of the proposal.

The following proposal and description focusses on the configuration metadata (i.e. configuration metadata built by a TSG and sent though any network to the TSA).

1. Metadata syntax and semantics

The main purpose of this configuration metadata item is the mapping of input TS (DVB-Sy components or other TS coming from other networks) to output TS that will be used to transmit the DVB-Ty services. It provides information to enable, within the multiplex, internal configuration of the TSA. It defines thus which service(s) and component(s) belong to which output TS.
Input TS level provides the list of input(s) used to build output TS. Output TS level provides the list of contributing PID (component and PSI/SI) and services. Service level configuration provides the current list of input services transmitted in an output TS, and the list of the services transmitted in all other output TS. PID level configuration provides the type of processing:

1. "as is": component that is involved in the output TS without content modification nor PID remapping.
2. "remapped component": component that contributes to the output TS after PID remapping.

The configuration also provides the type of processing to perform on PSI/SI related tables.
It also points out (gives its input TS/PID) the *jsi_system_reference* TS component and defines L2 signaling configuration details not available elsewhere for DVB-T2.

Syntax	Nb. of bits	Format	Comment
protocol_version_number	8	uimsbf	
terrestrial_standard_version_number	8	uimsbf	
if (**terrestrial_standard_version_number** == 2) {			DVB-T2 parameters
output_L2_T2_MI_PID	16	uimsbf	
output_L2_T2_MI_stream_id	8	uimsbf	
network_delay	32	uimsbf	
output_rate	32	uimsbf	
L2_signaling_generation	8	uimsbf	
if (**L2_signaling_generation**) {			
output_L2_service_id	16	uimsbf	
output_L2_service_provider_name_length	8	uimsbf	
for (i=0; i<N; i++) {			
char	8	uimsbf	
}			
output_L2_ service_name_length	8	uimsbf	
for (i=0; i<N; i++) {			
char	8	uimsbf	
}			
output_L2_PCR_PID	16	uimsbf	
output_L2_PMT_PID	16	uimsbf	

}			
}			
number_of_input_TS	8	uimsbf	
for (i=0; i<number_of_input_TS; i++) {			
input_TS_id	16	uimsbf	
input_ON_id	16	uimsbf	
jsi_system_reference_PID	16	uimsbf	
number_of_output_TS	8	uimsbf	
for (i=0; i<number_of_output_TS; i++) {			
output_TS_id	16	uimsbf	
output_ON_id	16	uimsbf	
network_id	16	uimsbf	
if (**terrestrial_standard_version_number** == 2) {			DVB-T2 parameter
PLP_id	16	uimsbf	
}			
number_of_PID	16	uimsbf	
for (i=0; i<number_of_PID; i++) {			
input_PID	16	uimsbf	
PID_processing_type	8	uimsbf	
if (PID_processing_type & 0x01) {			PID remapping
terrestrial_PID	16	uimsbf	
}			
if (PID_processing_type & 0x40) {			BAT processing
number_of_bouquet	8	uimsbf	
for (i=0; i<number_of_bouquet; i++) {			
input_bouquet_id	16	uimsbf	
terrestrial_bouquet_id	16	uimsbf	
}			
}			
}			
number_of_service	16	uimsbf	
for (i=0; i<number_of_service; i++) {			
input_service_id	16	uimsbf	
service_processing_type	8	uimsbf	
if (service_processing_type & 0x02) {			service_id patching
terrestrial_service_id	16	uimsbf	
}			
if (service_processing_type & 0x0C) {			EIT flags and/or running_status and/or free_CA_mode patching
terrestrial_status_and_flags	8	uimsbf	
}			
number_of_CAS_id	8	uimsbf	
for (i=0; i<number_of_CAS_id; i++) {			
CAS_id	16	uimsbf	CA Descr filtering in PMT (ECM stream invalidation)
}			
}			
number_of_CAS_id	8	uimsbf	
for (i=0; i<number_of_CAS_id; i++) {			
CAS_id	16	uimsbf	CA Descr filtering in CAT

			(EMM stream invalidation)
}			
}			
}			

Figure 8 – In-band configuration structure

protocol_version_number (8 bits) specifies the version number of the protocol used in the current sub_table. It shall be incremented by "1" when a change in the following description occurs.

terrestrial_standard_version_number (8 bits) specifies the version of the DVB-Ty (1 for DVB-T, 2 for DVB-T2).

output_L2_T2_MI_PID (16 bits) [DVB-T2] indicates the PID of the T2-MI packets which carries T2 data at the output of the TSA.

output_L2_MI_stream_id (8 bits) [DVB-T2] identifies the different T2-MI streams in a unique way; shall have the same value as the t2mi_stream_id field in the header of the T2-MI packets and is part of the T2-MI descriptor of the PMT table at the output of the TSA.

network_delay (32 bits) [DVB-T2] specifies an estimation of maximum delay of input TSs inside networks from TSGs to TSA.

output_rate (32 bits) [DVB-T2] is the output rate of the DVB/MPEG Transport Stream that carries T2-MI packets at the output of the TSA.

L2_signaling_generation (8 bits) [DVB-T2] when set to "1": indicates that the output L2 signaling shall be generated. When set to "0": indicates that the output L2 signaling shall NOT be generated.

output_L2_service_id (16 bits) [DVB-T2] gives the value of the service_id used in TSA output L2 signaling.

output_L2_service_provider_name_length (8 bits) [DVB-T2] specifies the number of bytes that follow this field for describing characters of the name of the service provider (included in the service descriptor of the SDT table at the output of the TSA).

output_L2_service_name_length (8 bits) [DVB-T2] specifies the number of bytes that follow this field for describing characters of the name of the service (included in the service descriptor of the SDT table at the output of the TSA).

char (8 bits) [DVB-T2] string of char fields that specifies the name of the service provider or service; text information is coded using the character sets and methods described in annex A of document [7].

output_L2_PCR_PID (16 bits) [DVB-T2] indicates the PID which carries PCR at the output of the TSA.

output_L2_PMT_PID (16 bits) [DVB-T2] indicates the PID which carries the PSI PMT table at the output of the TSA.

number_of_input_TS (8 bits) specifies the number of input TS (DVB-Sy or other input network) described in the following loop and that contributes to the DVB-Ty stream generation.

input_TS_id (16 bits) gives the input TS identification, from any other multiplex within the delivery system.

input_ON_id (16 bits) gives the identification of the input TS network of the originating delivery system.

jsi_system_reference_PID (16 bits) gives the input PID of the component that carries jsi_system_reference metadata.

number_of_output_TS (8 bits) specifies the number of output TS described in the following loop and that composes the T2-MI stream.

Note:

[DVB-T] set to 1.

[DVB-T2] matches the number of PLP.

output_TS_id (16 bits) identifies uniquely an output TS within the terrestrial system.

output_ON_id (16 bits) gives the identification of the output TS network of the originating delivery system.

network_id (16 bits) gives the identification of the output TS network of the delivery system.

PLP_id (16 bits) [DVB-T2] identifies uniquely an output TS within the T2 system identified by the T2_system_id.

number_of_PID (16 bits) gives the number of PID transmitted in the DVB-Ty stream (identified by its output_TS_id).

Note/reminder: this PID loop describes all elected PID, i.e. component PID or SI/PSI PID, possibly after remapping (in other words, all PID not present in the loop are discarded).

input_PID (16 bits) gives the PID that contributes to the current output TS, possibly before remapping (PID value in input TS).

PID_processing_type (8 bits) gives the processing to be performed on input component or table; bit field:

- bit0: PID remapping
- bit1: relevant for PSI/SI: "actual" table filtering (transmitted in input TS and not to be inserted in DVB-Ty; e.g NIT)
- bit2: relevant for PSI/SI: "other" table filtering (transmitted in input TS and not to be inserted in DVB-Ty; e.g NIT)
- bit3: relevant for PID 0x0012 (EIT): "schedule" table_id filtering
- bit4: relevant for PID 0x0012 (EIT): "p/f" table_id filtering
- bit5: relevant for PSI/SI: actual to other table_id patching (e.g. EIT: 0x4E → 0x4F, 0x5X → 0x6X)
- bit6: relevant for PID 0x0011: BAT flag (set to 1: BAT filtering or BAT patching according to bouquet_id loop)
- bit7: *RFU*

terrestrial_PID (16 bits) gives the PID that contributes to the current output TS, possibly after remapping.

number_of_bouquet (8 bits) gives the number of bouquets that contribute to the current output TS (and that are possibly to be patched: see input_bouquet_id and terrestrial_bouquet_id parameters below).

input_bouquet_id (16 bits) gives the bouquet_id in input TS.

terrestrial_bouquet_id (16 bits) gives the bouquet_id in DVB-Ty stream.

number_of_service (16 bits) gives the number of services that contribute to the current output TS.

Notes:

- This service loop describes all signalized services, i.e. as actual or other, possibly after service_id remapping or some parameter(s) patching.
- All sub-tables related to a service not present in the loop are to be invalidated by a Stuffing Section.

input_service_id (16 bits) gives the service_id in input TS.

service_processing_type (8 bits) gives the processing to be performed on table from input TS; bit field is:

- bit0: relevant for EIT processing: "other" service flag: if set to 1, related "actual" sub-tables to be patched to "other"
- bit1: service_id patching
- bit2: EIT_sheduling_flag and/or EIT_present_following_flag patching: values provided by terrestrial_status_and_flags bit field
- bit3: running_status patching and/or free_CA_mode patching: values provided by terrestrial_status_and_flags bit field (same values for all sub-tables that carry them) .
- bit5-7: *RFU*

Notes:

- No bit set means "no processing" (i.e. any sub-table related to the service is to be transmitted "as is").
- If any parameter of a sub-table is modified, CRC32 of the section is recalculated and patched.

terrestrial_service_id (16 bits) gives the service_id in current output TS.

terrestrial_status_and_flags (8 bits) gives the running_status and/or free_CA_mode and/or EIT_flags values of the service in the current output TS.

- bit0: EIT_schedule_flag; relevant if bit2 of service_processing_type set.
- bit1: EIT_present_following_flag; relevant if bit2 of service_processing_type set.
- bit2..4: running_status value; relevant if bit3 of service_processing_type set.
- bit5: free_CA_mode value; relevant if bit3 of service_processing_type set
- *bit6-7: RFU - set to 0*

number_of_CAS_id (8 bits) gives the number of CA_descriptor to be kept within PMT or CAT (a CA_descriptor being identified by the CAS_id it carries).

Note:

- This allows to perform PMT cleaning at TSA level to optimize overhead in SimulCrypt context, in case of CAS_id set in DVB-Ty is a subset of the CAS_id set in input TS (no specific terrestrial PMT generated and transmitted in input TS).
- Cleaning method: CA Descriptor invalidated by Stuffing Descriptor (Tag 0x42).

CAS_id (16 bits) gives the CAS_id to be kept.

Most common examples of configuration metadata related to each PID processing are provided in Annex C.

2. Metadata delivery method

The configuration metadata is carried out in a private data structure which conforms to the syntax of the generic private section defined in **[2]** ISO/IEC 13818-1 and that is used for mapping all MPEG-2 tables and SI tables into TS packets. It is conveyed as a private SI table using user defined table_id defined in **[5]**. A proposal is to use table_id 0xFE. A value could be reserved and standardized.
This table is transported over MPEG-TS in the same manner used for carrying PSI tables in **[2]**. The resulting component is added as ghost or signalized PID to the DVB-Sy multiplex.

Syntax	Nb. of bits	format	Comment
table_id	8	uimsbf	0xFE
section_syntax_indicator	1	bflbf	1
private_indicator	1	bflbf	0
reserved	2	bflbf	11
private_section_length	12	uimsbf	
table_id_extension	16	uimsbf	Reserved
reserved	2	bflbf	11
version_number	5	bflbf	
current_next_indicator	1	bflbf	1
section_number	8	uimsbf	0
last_section_number	8	uimsbf	0
private_data_byte	N*8	uimsbf	In-band configuration section
CRC32	32	uimsbf	

Figure 9 – In-band configuration table

table_id (8 bits) private table_id ‘0xFE’ is used to identify the in-band configuration table.
section_syntax_indicator (1 bit) shall be set to “1”.
private_indicator (1 bit) shall be set to “0”.
private_section_length (12 bits) specifies the number of bytes of the section, starting immediately following the private section length and including the CRC. The private section size shall not exceed 1021 so that the entire section has a maximum length of 1024 bytes.
table_id_extension (16 bits) Reserved (bits set to 1).
version_number (5 bits) the version number of the sub_table; shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0.
current_next_indicator (1 bit) set to “1” and indicates that the sub_table is the currently applicable sub_table.
section_number (8 bits) gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and table_id_extension.
last_section_number (8 bits) specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.
CRC_32 (32 bits) contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of ISO/IEC 13818-1 after processing the entire section.

Due to the large amount of use cases, system configuration and TSG strategies for table management, it is impossible to give one single figure about an estimated bitrate of the TS component carrying these configuration metadata.

In simple cases where only one input TS is used, an estimate of the bitrate for a 250ms repetition rate is about 6 kbps.

IV. Rules applicable to any of the entities in question – be it metadata or TSs or …

A. Rules for metadata delivery

1. Delivery of metadata for reference streams and input timestamping.

For a single modulation (DVB-T or DVB-T2, single or multi-plp), the reference stream reconstruction is computed from metadata inserted in a single input by a TSG. This TSG is "master" for this modulation (see Figure 4 - metadata in JSI system).

Each input of the TSA holds a PCR component computed from *jsi_system_clock* and inserted by a TSG, be it master or slave.

In order to maintain the consistency of $PCR_{jsi\_system\_clock}$, among the networks - especially in DVB-Sy modulators that may achieve rate adaptation and PCR correction -, it is recommended to add signaling for the *jsi_system_reference* TS component. A proposal is to define a new service type called "jsi_service_type" that references this PCR_PID. The "jsi_service_type" should rely in a current "reserved for future use" range to avoid disturbing DVB-Sy receivers.

2. Delivery of configuration metadata for PSI/SI processing.

In case of signaling configuration metadata, a proposal is to add a new extended descriptor in the NIT actual table called *jsi_delivery_system_descriptor* to signalize configuration metadata, and defined as follow:

Syntax	Number of bits	Identifier
jsi_delivery_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
jsi_system_id	16	uimsbf
reserved_future_use	3	bslbf
inband_configuration_metadata_PID	13	uimsbf
}		

jsi_system_id: identifies a JSI system in which TSA belongs to.

inband_configuration_metadata_PID: this 13-bit field indicates the PID of the transport stream packets which contains configuration metadata.

Following the proposal, a TSA shall be configured with a jsi_system_id and has to parse NIT actual table on each input TS to find the matching jsi_delivery_descriptor. The full configuration related to one DVB-Ty output shall be rebuilt from a single inputs TS.

B. Rules for padding inside T2 BB-frame

The TSA maps a re-multiplexed TS onto the DATA FIELD of the baseband frame. Nevertheless SYNCD and DFL are transmitted only once per T2 frame. The TSA shall ensure a piecewise distribution of TS packets inside the cumulated DATA FIELD available room so that

1. the first baseband frame will follow the exact value available in the BBHEADER part of the BB-reference,
2. padding over T2 frame is divided (integer division) among other baseband frame,
3. the remaining of the division is used to allocate some more padding onto the second baseband frame of the T2 frame.

These rules allow deterministic computation of SYNCD and DFL for every baseband frame.

Note that when null-packet deletion is inactive, the number of user packet that starts in the interleaving frame is known from BB-reference (frame_pkt_count). When null-packet deletion is active, the frame_pkt_count (number of TS packet including null-packet) differs from actual number of user packet. This latter is only known at the end of interleaving frame, so that padding over the whole frame is also known at end.

Annex A. Definitions and Abbreviations

Terrestrial Stream Adapter (*TSA*): a function or building block that converts a DVB-Sy signal to a DVB-Ty signal by deterministic operation.

Metadata: the metadata that will enable deterministic DVB-Ty signal construction to take place at the edge within the "Terrestrial Stream Adapter".

Terrestrial Stream Generator (*TSG*): a function or building block located in the Head-End that prepares the Metadata to be added in the input signal (DVB-Sy or other network).

TMF: DVB-T MegaFrames.

Baseband reference (BB-ref or BB-reference): JSI metadata structure that describes Baseband frames for a whole T2-frame. Baseband reference is mapped to the payload of T2-MI packet_type $A0_{16}$.

PCR: program clock reference.

PPS: pulse per second signal from GPS receiver.

jsi_system_clock: single illumination system clock is a 27MHz absolute clock synchronized on PPS and with start time at 2000-01-01T00:00:00Z.

jsi_system_reference: is the name of the metadata TS component that holds DVB-Ty transmission system characteristics in payload and PCR_field of the jsi_system_clock in adaptation_field.

patching: (or remapping) is the process of changing attribute value without changing size of the attribute (examples: table_id, PID).

filtering: is the process of removing information by discarding (e.g. PID component) or invalidating (e.g. sub-table using Stuffing Section).

Annex B. References

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

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

[3] ETSI TS 101 191: DVB mega-frame for Single Frequency Network (SFN) synchronization.

[4] ETSI EN 302 755: Modulator Interface (T2-MI) for a second generation digital terrestrial television broadcasting system (DVB-T2).

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

Annex C. PSI/SI processing situations

A. Most common examples of processing by PID

Precision: cases #3…26 are related to Strategy #3: sub-table from input TS transmitted in DVB-Ty after processing (see I.F above). Processing could also be associated with the strategy #2 (PID remapping) if needed.

PID	PID type	PID remapping	Table processing	Sub-table processing	Comment	#
xxx	Component or any unchanged table (DVB-Ty vs. input TS)	no	N.A.	N.A.	strategy #1: PID for input TS transmitted "as is" in DVB-Ty (see I.F above)	1
xxx	Component or any DVB-Ty specific table	Yes	N.A.	N.A.	strategy #2: PID for input TS transmitted after remapping in DVB-Ty (see I.F above)	2
0	PAT	no	N.A.	patching	TS_id patching	3
		no	N.A.	patching	program_number (service_id) remapping	4
		no	N.A.	patching	program_map_PID remapping (linked to remapped PMT PID - case #2)	5
1	CAT	no	N.A.	patching	CAS_id filtering (CA Descr patched into Stuffing Descr)	6
16	NIT	no	N.A.	patching	network_id patching	7
		no	N.A.	patching	(TS_id, ON_id) patching	8
17	BAT	no	filtering	N.A.	sub-table filtering (table_id patched into Stuffing Section)	9
		no	filtering	filtering	sub-table filtering by by bouquet_id (table_id patched into Stuffing Section)	10
		no	N.A.	patching	bouquet_id patching	11
		no	N.A.	patching	(TS_id, ON_id) patching	12
	SDT	no	filtering	N.A.	sub-table filtering by table_id: actual, other (table_id patched into Stuffing Section)	13
		no	patching	N.A.	table_id patching: actual → other	14
		no	N.A.	patching	(TS_id, ON_id) patching	15
		no	N.A.	patching	service_id remapping	16
		no	N.A.	patching	(EIT_schedule_flag, EIT_pf_flag, running_status, free_CA_mode) patching	17

18	EIT	no	filtering	filtering	sub-table filtering by (service_id, TS_id, ON_id) (table_id patched into Stuffing Section)	18
		no	filtering	N.A.	sub-table filtering by table_id: actual, other - p/f, schedule (table_id patched into Stuffing Section)	19
		no	patching	N.A.	table_id patching: actual → other *(then last_table_id patching)*	20
		no	N.A.	patching	service_id remapping	21
		no	N.A.	patching	(running_status, free_CA_mode) patching	22
xxx	PMT	no	N.A.	patching	program_number (service_id) remapping	23
		no	N.A.	patching	PCR_PID remapping (linked to remapped component PID)	24
		no	N.A.	patching	elementary_PID (component PID) remapping	25

B. Configuration parameters related to examples above

Thereafter is the description of the setting of the configuration parameters specifically involved to handle processing related to each case listed above. Other configuration parameters (T or T2 modulation related parameters, input TS identification, output TS identification, etc.) are not described.

#1: Component or any unchanged table (DVB-Ty vs. input TS) - no PID remapping:

- PID present in configuration PID loop (input_PID parameter within output TS loop, for item related to currently processed DVB-Ty).
- No PID and table and sub-table processing:
  - PID_processing_type$_{config}$: none bit set.

    ⇒ PID transmitted "as is".

#2: Component or any DVB-Ty specific table - PID remapping:

- PID present in configuration PID loop (input_PID parameter).
- PID processing:
  - PID_processing_type$_{config}$: bit0 set (PID remapping).

    ⇒ PID remapped with terrestrial_PID$_{config}$.
- No table and sub-table processing.

#3: PAT, no remapping, TS_id patching:

- PID 0 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
  - PID_processing_type$_{config}$: none bit set.

$\Rightarrow$ transport_stream_id$_{PAT}$ is set to output_TS_id$_{config}$.

#4: PAT, no remapping, program_number (service_id) remapping:

- PID 0 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.
    - PAT program loop and configuration service loop are parsed:
      for each service whose:
      - input_service_id$_{config}$ = program_number$_{PAT}$, and
      - service_processing_type$_{config}$: bit1 set (service_id remapping):
      $\Rightarrow$ program_number$_{PAT}$ is set to terrestrial_service_id$_{config}$.

#5: PAT, no PID remapping, program_map_PID remapping:

- PID 0 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.
    - PAT program loop and configuration PID loop are parsed:
      for each PID in PID loop whose:
      - input_PID$_{config}$ = PMT_PID$_{PAT}$, and
      - PID_processing_type$_{config}$: bit0 set (PID remapping):
      $\Rightarrow$ PMT_PID$_{PAT}$ is set to terrestrial_PID$_{config}$.

#6: CAT, no PID remapping, CAS_id filtering:

- PID 1 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.
    - configuration CAS_id loop (related to CAT, i.e. outside the service loop) is parsed:
      $\Rightarrow$ CAS_id present in CAT but not present in CAS_id loop are invalidated (patched into a Stuffing Descriptor).

#7: NIT, no PID remapping, network_id patching:

- PID 16 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.
    - configuration is parsed:
      $\Rightarrow$ network_id$_{NIT}$ is set to network_id$_{config}$.

#8: NIT, no PID remapping, (TS_id, ON_id) patching:

- PID 16 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.

- configuration is parsed:
  if $(TS\_id, ON\_id)_{NIT}$ = $(input\_TS\_id, input\_ON\_id)_{config}$:
  ⇒ $(TS\_id, ON\_id)_{NIT}$ is set to $(output\_TS\_id, output\_ON\_id)_{config}$.

#9: BAT, no PID remapping, table_id patching: Stuffing Section to filter all sub-tables:

- PID 17 present in configuration PID loop.
- No PID and sub-table processing.
- Table processing:
  - $PID\_processing\_type_{config}$: bit1 (actual table filtering).
  - $PID\_processing\_type_{config}$: bit6 set (BAT processing).
  - $number\_of\_bouquet_{config}$ = 0.
    ⇒ sub-table is invalidated (table_id patched into Stuffing Section value).

#10: BAT, no PID remapping, sub-table filtering by bouquet_id:

- PID 17 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
  - $PID\_processing\_type_{config}$: bit1 and/or bit2 (actual and/or other table filtering).
  - $PID\_processing\_type_{config}$: bit6 set (BAT processing).
  - configuration bouquet loop related to current DVB-Ty is parsed:
    if no $input\_bouquet\_id_{config}$ = $bouquet\_id_{BAT}$:
    ⇒ sub-table is invalidated (table_id patched into Stuffing Section value).

#11: BAT, no PID remapping, bouquet_id patching:

- PID 17 present in PID loop.
- No PID and table processing.
- Sub-table processing:
  - PID_processing_type: none bit set.
  - configuration bouquet loop related to current DVB-Ty is parsed:
    if $bouquet\_id_{BAT}$ = $input\_bouquet\_id_{config}$:
    ⇒ $bouquet\_id_{BAT}$ is set to $terrestrial\_bouquet\_id_{config}$.

#12: BAT, no PID remapping, (TS_id, ON_id) patching:

- PID 17 present in PID loop.
- No PID and table processing.
- Sub-table processing:
  - PID_processing_type: none bit set.
  - configuration is parsed:
    if $(TS\_id, ON\_id)_{BAT}$ = $(input\_TS\_id, input\_ON\_id)_{config}$:
    ⇒ $(TS\_id, ON\_id)_{BAT}$ is set to $(output\_TS\_id, output\_ON\_id)_{config}$.

#13: SDT, no PID remapping, sub-table filtering by table_id (actual, respectively other):

- PID 17 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
  - $PID\_processing\_type_{config}$: bit1, resp. bit2, set (actual table, resp. other, table filtering).

- configuration bouquet loop related to current DVB-Ty is parsed:
  if $table\_id_{SDT}$: matches is 0x42, resp. 0x46:
  $\Rightarrow$ sub-table is invalidated (table_id patched into Stuffing Section value).

#14: SDT, no PID remapping, table_id patching: actual $\rightarrow$ other:

- PID 17 present in configuration PID loop.
- No PID and sub-table processing.
- Table processing:
  - $PID\_processing\_type_{config}$: bit5 set (actual to other table_id patching).
  - if $table\_id_{SDT}$ is 0x42:
    $\Rightarrow$ $table\_id_{SDT}$ is patched to 0x46.

#15: SDT, no PID remapping, (TS_id, ON_id) patching:

- PID 17 present in PID loop.
- No PID and table processing.
- Sub-table processing:
  - $PID\_processing\_type_{config}$: none bit set.
  - configuration is parsed:
    if $(TS\_id, ON\_id)_{SDT}$ = $(input\_TS\_id, input\_ON\_id)_{config}$:
    $\Rightarrow$ $(TS\_id, ON\_id)_{SDT}$ is set to $(output\_TS\_id, output\_ON\_id)_{config}$.

#16: SDT, no PID remapping, service_id remapping:

- PID 17 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
  - $PID\_processing\_type_{config}$: none bit set.
  - SDT program loop and configuration service loop are parsed:
    for each service whose $program\_number_{SDT}$ = $input\_service\_id_{config}$ and
    $service\_processing\_type_{config}$: bit1 set (service_id remapping):
    $\Rightarrow$ $service\_id_{SDT}$ is set to $terrestrial\_service\_id_{config}$.

#17: SDT, no PID remapping, (EIT_schede_flag, EIT_pf_flag, running_status, free_CA_mode) patching:

- PID 17 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
  - $PID\_processing\_type_{config}$: none bit set.
  - SDT program loop and configuration service loop are parsed:
    for each service whose:
    - $program\_number_{SDT}$ = $output\_service\_id_{config}$, and
    - $service\_processing\_type_{config}$: bit2 set (EIT_shed_flag and/or EIT_pf_flag patching) and/or bit5 set (running_status patching and/or free_CA_mode patching):
    $\Rightarrow$ $(EIT\_shed\_flag, EIT\_pf\_flag, running\_status, free\_CA\_mode)_{SDT}$ is set to $terrestrial\_status\_and\_flags_{config}$.
- *Precision*: this case #17 can be combined with previous case #16.

#18: EIT, no PID remapping, sub-table filtering by service_id:

- PID 18 present in configuration PID loop.

- No PID and sub-table processing.
- Table processing:
    - $PID\_processing\_type_{config}$: none bit set.
    - configuration output and service loops are parsed:
      if $service\_id_{EIT} \neq$ each $terrestrial\_service\_id_{config}$:
      $\Rightarrow$ sub-table is invalidated (table_id patched into Stuffing Section value).

#19: EIT, no PID remapping, sub-table filtering by table_id (actual, other - p/f, schedule):

- PID 18 present in configuration PID loop.
- No PID and sub-table processing.
- Table processing:
    - $PID\_processing\_type_{config}$: bit1 and/or bit2 set (actual and/or other table filtering).
    - $PID\_processing\_type_{config}$: bit3 and/or bit4 set (schedule and/or p/f table filtering).
    - if $table\_id_{EIT}$ = type identified by PID_processing_type bits set:
      $\Rightarrow$ sub-table is invalidated (table_id patched into Stuffing Section value).

#20: EIT, no PID remapping, table_id patching: actual $\rightarrow$ other:

- PID 18 present in configuration PID loop.
- No PID and sub-table processing.
- Table processing: actual tables processing:
    - configuration service loop is parsed:
      if $service\_id_{EIT} = terrestrial\_service\_id_{config}$, and
      $service\_processing\_type_{config}$: bit0 set (other service):
      $\Rightarrow$ $table\_id_{SDT}$ is patched (0x4E $\rightarrow$ 0x4F, or 0x5X $\rightarrow$ 0x6X).

#21: EIT, no PID remapping, (service_id, TS_id, ON_id) remapping:

- PID 18 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - $PID\_processing\_type_{config}$: none bit set.
    - configuration output and service loops are parsed: if:
      - $(service\_id, transport\_stream\_id, original\_stream\_id)_{EIT}$ =
        $(input\_service\_id, input\_TS\_id, input\_ON\_id)_{config}$, and
      - $service\_processing\_type_{config}$: bit1 (service_id):
      $\Rightarrow$ $(service\_id, transport\_stream\_id, original\_stream\_id)_{EIT}$ is set to
      $(terrestrial\_service\_id, output\_TS\_id, output\_ON\_id)_{config}$.

#22: EIT, no PID remapping, (running_status, free_CA_mode) patching:

- PID 18 present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - $PID\_processing\_type_{config}$: none bit set.
    - configuration output and service loops are parsed: if:
      - $(service\_id, transport\_stream\_id, original\_stream\_id)_{EIT}$ =
        $(output\_service\_id, output\_TS\_id, output\_ON\_id)_{config}$, and
      - $service\_processing\_type_{config}$: bit3 set (running_status and/or free_CA_mode

patching):

$\Rightarrow$ (running_status, free_CA_mode) $_{EIT}$ is set to terrestrial_status_and_flags$_{config}$.

- *Precision*: this case #22 can be combined with previous case #21.

#23: PMT, no PID remapping, program_number (service_id) remapping:

- PID present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - PID_processing_type$_{config}$: none bit set.
    - configuration output and service loops are parsed: if:

      - (program_number)$_{PMT}$ = (input_service_id, output_TS_id, output_ON_id)$_{config}$, and

      - service_processing_type$_{config}$: bit1 set (service_id patching):

      $\Rightarrow$ (program_number)$_{PMT}$ is set to output_service_id$_{config}$.

#24: PMT, no PID remapping, PCR_PID remapping (linked to remapped component PID):

- PID present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - configuration PID loop is parsed: if:

      - PID_processing_type$_{config}$: bit0 set (PID remapping), and

      - PCR_PID $_{PMT}$ = input_PID $_{config}$:

      $\Rightarrow$ PCR_PID $_{PMT}$ is set to terrestrial_PID $_{config}$.

#25: PMT, no PID remapping, elementary_PID (component PID) remapping:

- PID present in configuration PID loop.
- No PID and table processing.
- Sub-table processing:
    - configuration PID loop is parsed: if:

      - PID_processing_type$_{config}$: bit0 set (PID remapping), and

      - elementary_PID $_{PMT}$ = input_PID$_{config}$:

      $\Rightarrow$ elementary_PID$_{PMT}$ is set to terrestrial_PID $_{config}$.

harmonic

C. Extended SI Processing for multiple input networks

This chapter describes several use-cases leading to several processing whose complexity depends on the strategy of the TSA.

Case #1: DVB-T2, M-PLP (3 PLP):

- Configuration (e.g.):

	DTH1 (input1)	DTH2 (input2)	PLP1	PLP2	PLP3
TS_id	100	200	10	20	30
ON_id	101	201	11	21	31
Srv_id	111 222 333	111 444 555 666	111 [DTH1] 222	111 [DTH2] → **112** 333	444 555
Cmp PID	1111, 1112 2221, 2222 3331, 3332	1111, 1113 4441, 4442 5551, 5552 6661, 6662	1111, 1112 2221, 2222	1111 → **1121**, 1113 3331, 3332	4441, ~~**4442**~~, **3331** (1) 5551, 5552

(1) Cmp PID 3331: attachment of a component (shared component).

- PSI/SI:

Note about SI: in that use case, DVB-Tyi mux is to be signalized as other in others DVB-Ty mux.

	DTH1	DTH2	PLP1	PLP2	PLP3
PAT			specific	specific	specific
CAT			specific	specific	specific
PMT			111[DTH1] unchanged 222[DTH1] unchanged	111 specific 333[DTH1]unchanged	444 specific 555[DTH2] unchanged
TSDT			specific	specific	specific
NIT act			specific	specific	specific
NIT oth			filtered	filtered	filtered
SDT act			specific	specific	specific
SDT oth			#1: PLP2 SDT act, TId patched #2: PLP3 SDT act, TId patched	#1: PLP1 SDT act, TId patched #2: PLP3 SDT act, TId patched	#1: PLP1 SDT act, TId patched #2: PLP2 SDT act, TId patched
TDT			[DTH*i*] unchanged	[DTH*i*] unchanged	[DTH*i*] unchanged
TOT			specific	specific	specific
BAT			specific	specific	specific
RST			specific	specific	specific
EIT act	EIT act (100, 101, 111) EIT act (100, 101, 222) EIT act (100, 101, 333)	EIT act (200, 201, 111) EIT act (200, 201, 444) EIT act (200, 201, 555) EIT act (200, 201, 666)	EIT act (**10**, **11**, 111) EIT act (**10**, **11**, 222) EIT **oth** (**20**, **21**, 333) EIT **oth** (**20**, **21**, **112**) EIT **oth** (**30**, **31**, 444) EIT **oth** (**30**, **31**, 555) ~~EIT act (200, 201, 666)~~ (filtered)	EIT **oth** (**10**, **11**, 111) EIT **oth** (**10**, **11**, 222) EIT act (**20**, **21**, 333) EIT act (**20**, **21**, **112**) EIT **oth** (**30**, **31**, 444) EIT **oth** (**30**, **31**, 555) ~~EIT act (200, 201, 666)~~ (filtered)	EIT **oth** (**10**, **11**, 111) EIT **oth** (**10**, **11**, 222) EIT **oth** (**20**, **21**, 333) EIT **oth** (**20**, **21**, **112**) EIT act (**30**, **31**, 444) EIT act (**30**, **31**, 555) ~~EIT act (200, 201, 666)~~ (filtered)
EIT oth	EIT oth (xxx, xxx, xxx)	EIT oth (xxx, xxx, xxx)	filtered	filtered	filtered

Case #2: 3 DVB-Ty multiplexes (3 S-PLP in DVB-T2), each service of a multiplex signalized as other in the other multiplexes**:**

- Configuration (e.g.):

	DTH1 (input1)	DTH2 (input2)	TS1	TS2	TS3
TS_id	100	200	10	20	30
ON_id	101	201	11	21	31
Srv_id	111 222 333	444 555 666	111 222	333 444	555 666
Cmp PID	1111, 1112 2221, 2222 3331, 3332	4441, 4442 5551, 5552 6661, 6662	1111, 1112 2221, 2222	3331, 3332 4441, 4442	5551, 5552 6661, 6662

- PSI/SI:

Note about SI: in that use case, TSi multiplex is to be signalized as other in the other TS multiplexes.

	DTH1	DTH2	TS1	TS2	TS3
PAT			specific	specific	specific
CAT			specific	specific	specific
PMT			111[DTH1] unchanged 222[DTH1] unchanged	333[DTH1] unchanged 444[DTH2] unchanged	555[DTH2] unchanged 666[DTH2] unchanged
TSDT			specific	specific	specific
NIT act			specific	specific	specific
NIT oth			filtered	filtered	filtered
SDT act			specific	specific	specific
SDT oth			#1: TS2 SDT act, TId patched #2: TS3 SDT act, TId patched	#1: TS1 SDT act, TId patched #2: TS3 SDT act, TId patched	#1: TS1 SDT act, TId patched #2: TS2 SDT act, TId patched
TDT			[DTH*i*] unchanged	[DTH*i*] unchanged	[DTH*i*] unchanged
TOT			specific	specific	specific
BAT			specific	specific	specific
RST			specific	specific	specific
EIT act	EIT act (100, 101, 111) EIT act (100, 101, 222) EIT act (100, 101, 333)	EIT act (200, 201, 444) EIT act (200, 201, 555) EIT act (200, 201, 666)	EIT act (**10**, **11**, 111) EIT act (**10**, **11**, 222) EIT **oth** (**20**, **21**, 333) EIT **oth** (**20**, **21**, 444) EIT **oth** (**30**, **31**, 555) EIT **oth** (**30**, **31**, 666)	EIT **oth** (**10**, **11**, 111) EIT **oth** (**10**, **11**, 222) EIT act (**20**, **21**, 333) EIT act (**20**, **21**, 444) EIT **oth** (**30**, **31**, 555) EIT **oth** (**30**, **31**, 666)	EIT **oth** (**10**, **11**, 111) EIT **oth** (**10**, **11**, 222) EIT **oth** (**20**, **21**, 333) EIT **oth** (**20**, **21**, 444) EIT act (**30**, **31**, 555) EIT act (**30**, **31**, 666)
EIT oth	EIT oth (xxx, xxx, xxx)	EIT oth (xxx, xxx, xxx)	Filtered	Filtered	filtered

Case #3: 2 DVB-Ty multiplexes (M-PLP or not), **all related PSI/SI transmitted in one input TS:**

- Configuration (e.g.):

	DTH1 (input1)	DTH2 (input2)	DTH3 (input3)	TS1	TS2
TS_id	100	200	300	10	20
ON_id	101	201	301	11	21
Srv_id	111 222 333	444 555 666		111 222	333 444
Cmp PID	1111, 1112 2221, 2222 3331, 3332	4441, 4442 5551, 5552 6661, 6662	100 (PAT of TS1) 200 (PAT of TS2) 101 (CAT of TS1) 201 (CAT of TS2) 1111 (SDT of TS1) 1112 (SDT of TS2) 118 (EIT of TS1) 218 (EIT of TS2) 120 (TOT of TS1) 220 (TOT of TS2)	1111, 1112 2221, 2222	3331, 3332 4441, 4442

- PSI/SI:

	DTH1	DTH2	DTH3	TS1	TS2
PAT				100[DTH3] remapped	200[DTH3] remapped
CAT				101[DTH3] remapped	201[DTH3] remapped
PMT				111[DTH1] unchanged 222[DTH1] unchanged	333[DTH1] unchanged 444[DTH2] unchanged
TSDT					
NIT act					
NIT oth					
SDT act				1111[DTH3] unchang.	1112[DTH3] unchang.
SDT oth				1112[DTH3], TId patched	1111[DTH3], TId patched
TDT				[DTH*i*] unchanged	[DTH*i*] unchanged
TOT				120[DTH3] remapped	220[DTH3] remapped
BAT					
RST					
EIT act	EIT act (100, 101, 111) EIT act (100, 101, 222) EIT act (100, 101, 333)	EIT act (200, 201, 444) EIT act (200, 201, 555) EIT act (200, 201, 666)		118[DTH3] remapped	218[DTH3] remapped
EIT oth	EIT oth (xxx, xxx, xxx)	EIT oth (xxx, xxx, xxx)			