Draft ETSI EN 302 550-1-1 V1.1.0 (2009-07)

European Standard (Telecommunications series)

Satellite Earth Stations and Systems (SES);
Satellite Digital Radio (SDR) Systems;
Part 1: Physical Layer of the Radio Interface;
Sub-part 1: Outer Physical Layer

Reference

DEN/SES-00312-1-1

Keywords

digital, layer 1, radio, satellite

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

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://webapp.etsi.org/IPR/home.asp).

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 European Standard (Telecommunications series) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES), and is now submitted for the Public Enquiry phase of the ETSI standards Two-step Approval Procedure.

The present document is part 1, sub-part 1 of a multi-part deliverable covering Satellite Digital Radio (SDR), as identified below:

Part 1: "Physical Layer of the Radio Interface";

Sub-part 1: "Outer Physical Layer";

Sub-part 2: "Inner Physical Layer Single Carrier Modulation";

Sub-part 3: "Inner Physical Layer Multi Carrier Modulation".

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Introduction

TC SES is producing standards and other deliverables for Satellite Digital Radio (SDR) systems. An SDR system enables broadcast to fixed and mobile receivers through satellites and complementary terrestrial transmitters. Functionalities, architecture and technologies of such systems are described in TR 102 525 [i.1].

Several existing and planned ETSI standards specify parts of the SDR system, with the aim of interoperable implementations. The physical layer of the radio interface (air interface) is divided up into the outer physical layer, the inner physical layer with a single carrier modulation, and the inner physical layer with multi carrier modulation. These parts can be used all together in SDR compliant equipment, or in conjunction with other existing and future specifications.

The present document specifies the outer physical layer. The inner physical layer with single carrier modulation is specified in EN 302 550-1-2 [i.2], and with multi carrier modulation in EN 302 550-1-3 [i.3]. Guidelines for using the physical layer standard can be found in TR 102 604 [i.4].

The physical layer specifications have previously been published as "Technical Specification (TS)" type ETSI deliverables. The present document supersedes TS 102 550 [i.5] and is recommended for new implementations.

1 Scope

The present document concerns the radio interface of SDR broadcast receivers. It specifies the functionality of the outer physical layer. It allows implementing this part of the system in an interoperable way.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
    - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
    - for informative references.

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 indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

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

[2] ISO/IEC 11172-1: "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: Systems".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area**.** For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1] ETSI TR 102 525: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) service; Functionalities, architecture and technologies".

[i.2] ETSI EN 302 550-1-2: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Part 1: Physical Layer of the Radio Interface; Sub-part 2: Inner Physical Layer Single Carrier Modulation".

[i.3] ETSI EN 302 550-1-3: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Part 1: Physical Layer of the Radio Interface; Sub-part 3: Inner Physical Layer Multi Carrier Modulation".

[i.4] ETSI TR 102 604: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Guidelines for the Use of the Physical Layer Standards".

[i.5] ETSI TS 102 550 (V1.3.1): "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Outer Physical Layer of the Radio Interface".

3 Symbols and abbreviations

3.1 Symbols

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

<R>	Code rate

3.2 Abbreviations

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

AWGN	Additive White Gaussian Noise
BCH	Bose, Ray-Chaudhuri, Hocquenghem code
CRC	Cyclic Redundancy Checksum
C-TS	Channel-Transport Stream
CU	Capacity Unit
FEC	Forward Error Correction
ID	IDentifier
IP	Internet Protocol
IPL	Inner Physical Layer
IU	Interleaving Unit
LSB	Least Significant Bit
MPEG-TS	MPEG Transport Stream
MSB	Most Significant Bit
MTU	Maximum Transfer Unit
OPL	Outer Physical Layer
PF	Physical layer FEC
PFIW	Physical layer FEC Info Word
PL	Physical Layer
QoS	Quality of Service
RFU	Reserved for Future Use
SL	Service Layer
SOF	Start Of Frame
S-TS	Service-Transport Stream
VBR	Variable Bit Rate
WER	Word Error Rate

4 Outer physical layer

Refer to annex A for number format definitions.

4.1 Overview

Figure 1 displays the position and the interfaces of the Outer Physical Layer (in the following denoted by OPL) inside a complete broadcast transmission chain. The OPL connects to the Service Layer, where the interface is Service Transport Streams (S-TS) on the one side, and on the other side to the Inner Physical Layer (IPL - described in EN 302 550-1-2 [i.2] and EN 302 550-1-3 [i.3]), where the interfaces are Channel Transport Streams (C-TS).

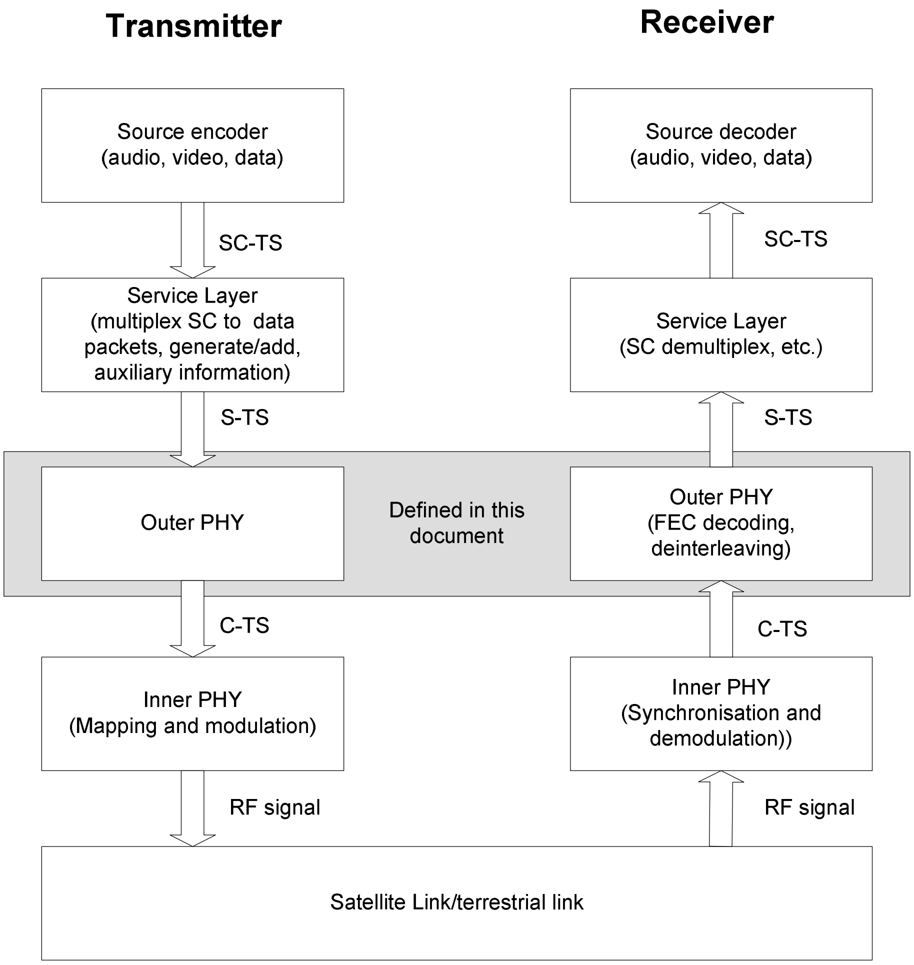

Transmitter
Receiver
Source encoder (audio, video, data)
Source decoder (audio, video, data)
SC-TS
SC-TS
Service Layer (multiplex SC to data packets, generate/add, auxiliary information)
Service Layer (SC demultiplex, etc.)
S-TS
S-TS
Outer PHY
Defined in this document
Outer PHY (FEC decoding, deinterleaving)
C-TS
C-TS
Inner PHY (Mapping and modulation)
Inner PHY (Synchronisation and demodulation))
RF signal
RF signal
Satellite Link/terrestrial link

Figure 1: Position and interfaces of the OPL inside the transmission chain

The following table gives an overview about the terminology used for the data streaming through the system.

Description		Comments
SC	Service component	E.g. source encoded audio or video or other data
SC-TS	Service component transport stream	
ES	Elementary Stream	ES: Elementary Stream, a generic term for one of the coded video, coded audio or other coded data bitstreams, cf. MPEG-1 standard ISO/IEC 11172-1 [2].
Program	A program is a collection of program elements. Program elements may be elementary streams (ES, SC-TS).	In line with the definition used for MPEG standard ISO/IEC 13818-1 [1].
Service	Set of programs and related auxiliary information	
S-TS	Service transport stream	Generalized term for transport stream. MPEG-TS is one example for a service transport stream.
MPEG-TS	Transport stream compliant to MPEG standard ISO/IEC 13818-1 [1]	
C-TS	Channel transport stream	Data stream (bit stream) representing the input to the modulator = data stream including all redundancy added by the FEC encoder - possibly with time-interleaving - and carrying configuration signalling information for the receiver. The content of the C-TS is referred to as a C-TS multiplex (a multiplex of encoded and interleaved S-TS plus signalling information). A bouquet of programs is carried by one or more C-TS multiplexes.

Description		Comments
Channel	RF resource	The meaning "RF resource" is aligned with the terminology used for DVB.

The functionality of the Outer Physical Layer is to provide Forward Error Correction and time interleaving for resistance against a variety of transmission channel conditions. Different transport channels are used in the OPL to offer the requested performance for different types of services. These transport channels are called pipes in the scope of the present document. The OPL is configurable in terms of error protection, outage mitigation in case of signal losses, end-to-end delay, zapping time, payload throughput and receiver complexity.

Multiple pipes can be used as described above. Each of them contains FEC, Mixer and Disperser. One special pipe exists whose functionality is to transmit all relevant parameters to decode the other pipes. The so-called signalling pipe is always transmitted at the lowest coderate which is 1/5. The modulation of the signalling pipe is equal to the modulation of the data pipes.

The general block diagram of the OPL functionality is given in figure 2.

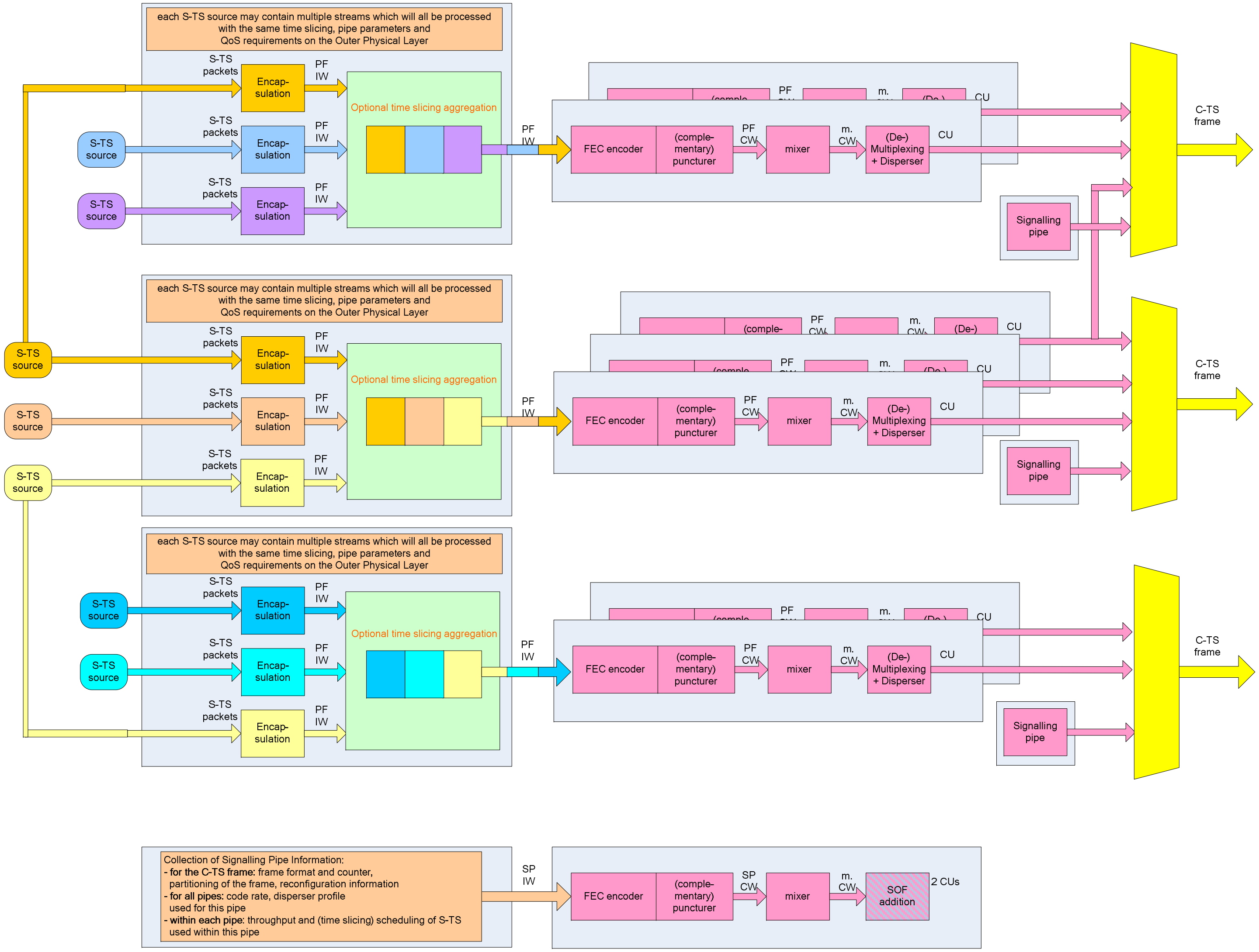

each S-TS source may contain multiple streams which will all be processed with the same time slicing, pipe parameters and QoS requirements on the Outer Physical Layer
S-TS packets
S-TS source
Encap-sulation
PF IW
Optional time slicing aggregation
FEC encoder
(comple-mentary) puncturer
PF CW
mixer
m. CW
(De-) Multiplexing + Disperser
CU
Signalling pipe
C-TS frame
Collection of Signalling Pipe Information:
- for the C-TS frame: frame format and counter, partitioning of the frame, reconfiguration information
- for all pipes: code rate, disperser profile used for this pipe
- within each pipe: throughput and (time slicing) scheduling of S-TS used within this pipe
SP IW
SP CW
SOF addition
2 CUs

Figure 2: General overview of the OPL functionality

The processing, multiplexing and demultiplexing of the data in the OPL is displayed in figure 3. An S-TS scheduler multiplexes together all S-TS contained in the pipe. The scheduler is controlled by an S-TS schedule, which determines the number of words taken from one S-TS before the multiplexer selects the next S-TS of the pipe. After an encapsulation, FEC encoding and mixing, the codewords (segmented into interleaver units) are demultiplexed codeword-wise to the slots of the considered pipe, each of the slots possessing its individual disperser. After demultiplexing a codeword to a slot, i.e. to the input of its disperser, the slot demultiplexer selects the next slot/disperser. At the outputs of the dispersers, the dispersed codewords are multiplexed together again by the collector to form one pipe. The slot demultiplexer and the collector always select synchronously the same slot/disperser.

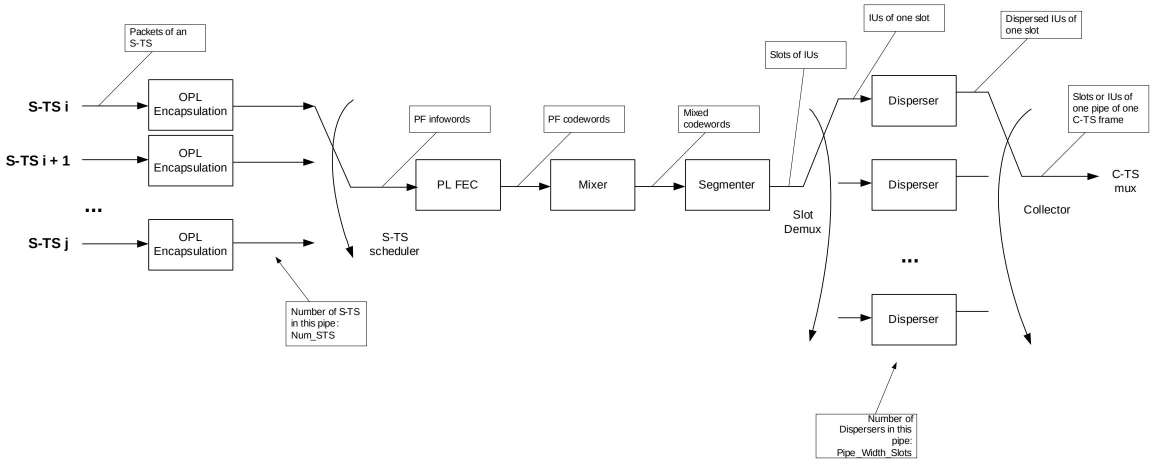


Figure 3: Definition of the different blocks involved in the OPL processing

4.2 Interfacing to Service Layer (SL)

The interface to the service layer is the so-called Service-Transport Stream (S-TS). For the OPL, each S-TS source is the smallest granularity which can be processed independently.

The interface may work synchronously or asynchronously. In the case of asynchronous interface, the PL must be able to accept at least the average data rate that is provided by the SL. Any data buffering shall be done inside the SL, such that no data from the S-TS is lost at this interface. When the PL requests new data for transmission, the SL can either provide the requested data to the PL or it can signal that no data is currently available. If no data is available for transmission, the PL instead transmits dummy data that is discarded in the receiver.

Inside an S-TS, multiplexing and de-multiplexing of information shall be carried out by the service layer.

Each pipe provides a different set of transmission parameters (e.g. FEC code rate and disperser profile), and achieves a different QoS in terms of protection against transmission errors and end-to-end delay. One pipe of the OPL may carry several S-TS, all with the same QoS parameters.

If PL time slicing is used, each time slice is associated with one S-TS. The scheduling of the S-TS, i.e. their start instants and lengths, inside a pipe can be adapted frequently (once per schedule/time slicing period). This opens the possibility of handling Variable Bit Rate (VBR) transmission.

The maximum allowed payload throughput per S-TS is 3,2 Mbit/s (this corresponds to approximately 8 to 10 video services inside one S-TS). This is the throughput that the processing chain inside the receiver (e.g. the turbo decoder) must be able to handle at least.

4.3 S-TS to OPL adaptation layer: S-TS encapsulation

The OPL is prepared to transport different types of S-TS, and a mixture of different S-TS types may be transported simultaneously over one C-TS multiplex.

The following parameters have to be determined for each S-TS (for parameters, refer to signalling pipe in clause 4.10.1):

- S-TS ID: identifier for the transported S-TS, that is unique for each network operator (i.e. for each Operator_ID); observe that one S-TS may be transported over multiple instances of the PL and still have a single unique S-TS ID; this helps, for example, for diversity combining of one S-TS transmitted over satellite and simultaneously over terrestrial repeaters. Several rules apply for the S-TS:
  - S-TS ID 0 plays a special role: this is the Service Layer configuration S-TS (the SL can signal its own configuration via this S-TS).
  - An S-TS may be fed to several C-TS multiplexes. The S-TS IDs in all of these C-TS multiplexes are identical.
  - An S-TS may not be fed to several pipes inside the same C-TS multiplex, and an S-TS may not be fed several times to the same pipe inside one C-TS multiplex either.
  - S-TS IDs must be unique over the complete network of one operator except for S-TS ID `0` which is allowed on every C-TS multiplex.
  - S-TS with an identical `Operator_ID` and S-TS ID can always be diversity combined (except for S-TS ID 0).
  - The length of an S-TS can be configured in a granularity of one PL infoword per C-TS frame.
- Pipe number that this S-TS is transported over.

Moreover, for the ensemble of S-TS contained inside a complete C-TS multiplex, the following parameters have to be fixed (for parameters, refer also to signalling pipe in clause 4.10.1):

- Operator_ ID: unique identifier for the network operator.
- Partitioning of the C-TS multiplex into pipes and scheduling of the S-TS inside the pipes, i.e. what is the data rate of one S-TS and when are the bursts of one S-TS transported.

Each S-TS is partitioned into packets to match the length of the PL FEC information word (PF infoword). The packet size is individual for each type of S-TS. The OPL encapsulation inside the S-TS to OPL adaptation layer adapts the length of the S-TS packets to the PF infoword length by appending a suffix to the S-TS packet. Table 1 defines the S-TS packet length and the suffix length for different S-TS types.

Table 1: Defined S-TS type IDs

S-TS Type	S-TS Type ID	S-TS payload packet Size in bytes	Suffix length in bits	Comment
Dummy packet	0	0	26	used for asynchronous sl/pl interface. is discarded in receiver.
Transparent	1	1 532	26	sl has to decide what to do with this data.
MPEG-TS	2	1 504	250	payload packet is 8 mpeg packets of 188 bytes each; additionally, a bch code of 196 bits is applied.
IP stream	3	1 504	250	mtu of ip = 4 095 bytes with 2 bytes additional header per packet.
RFU	4 to 7			reserved for future s-ts types.

The detailed format for the different types of S-TS is given in the following clauses. The Cyclic Redundancy Check (CRC) polynomial, which appears in the following clauses, is $x^8 + x^5 + x^3 + x^2 + x + 1$ for all S-TS stream types. The calculation of the CRC is described in annex B.

4.3.1 PF infoword format for S-TS stream type 0 (dummy packet)

The format of the dummy packet is given in table 2. The insertion of a dummy packet is performed if no data was available at the instant of processing the actual packet in the OPL.

Table 2: PF infoword format for S-TS stream type 0 (dummy packet)

Start bit index	Parameter	Description	Wordsize (bits)	Format	Comment
0	Dummy data	To be filled with zeros	12 256	1 532×U8 (1 532 bytes)	
12 256	RFU	4 bits reserved for future use	4	U4	helps to bit-align the payload to byte boundaries.
12 260	STS_ID	S-TS ID	8	U8	can be chosen arbitrarily.
12 268	STS_Stream_Type_ID	S-TS stream type identifier	3	U3	fixed to 0 for dummy packets.
12 271	Encap_Ver	Version number of the OPL encapsulation format	3	U3	fixed to 0.
12 274	HeaderCRC	CRC over the 18 relevant bits of the header	8	U8	the light grey marked bits are included in the header.
		Total length of PFIW	**12 282**		

4.3.2 PF infoword format for S-TS stream type 1 (transparent)

The format of the transparent mode is given in table 3. It provides a transparent transmission of whatever payload. The throughput capability of the transparent stream type is 1 532 bytes per PF infoword. No additional error correction or detection except the turbo code is used; therefore, data integrity and flow control needs to be performed by the link layer. The definition of such protocol is not included in the present document.

Table 3: PF infoword format for S-TS stream type 1 (transparent)

Start bit index	Parameter	Description	Wordsize (bits)	Format	Comment
0	Payload_Packet	Transparent payload packet	12 256	1 532×U8 (1 532 bytes)	May include counters, error correction and error detection.
12 256	RFU	4 bits reserved for future use	4	U4	Helps to bit-align the payload to byte boundaries.
12 260	STS_ID	S-TS ID	8	U8	
12 268	STS_Stream_Type_ID	S-TS stream type identifier	3	U3	Fixed to 1 for transparent packets.
12 271	Encap_Ver	Version number of the OPL encapsulation format	3	U3	Fixed to 0.
12 274	HeaderCRC	CRC over the 18 relevant bits of the header	8	U8	The light grey marked bits are included in the CRC check.
		Total length of PFIW	**12 282**		

4.3.3 PF infoword format for S-TS stream type 2 (MPEG-TS)

The format of the MPEG-TS stream mode is given in table 4. It provides a transparent transmission of up to 8 MPEG-TS packets according to ISO/IEC 13818-1 [1], each having a size of 188 bytes. If less than 8 packets are available for transport, the missing packets are filled by MPEG-TS null packets. Additional error correction and detection is performed by using one shortened BCH (3 057, 3 008) code each 2 MPEG-TS packets. Therefore, each PF infoword contains 4 sections of BCH parity of 49 bits each.

As this BCH-code is a systematic code, the parity may be discarded in the receiver if this additional parity check is not desired; however, performance is supposed to degrade in this case. On the contrary, it is a mandatory requirement on the transmitter side to include this parity.

The error correction code (overall minimum distance d_{min} = 10) is actually an outer BCH(3056, 3008, 9) code (with minimum distance d_{min} = 9) concatenated by an inner single-parity check code (3057,3056,1). The BCH code is gained by shortening a narrow-sense binary BCH(4095,4047,9)-Code. Concatenated encoding of (payload) message bits $\boldsymbol{m} = (m_{3007}, m_{3006}, ..., m_1, m_0)$ onto an (overall) codeword

c = $(m_{3007}, m_{3006}, ..., m_1, m_0, d_{47}, d_{46}, ..., d_1, d_0, p_0)$ is achieved as follows:

- The message bit m_{3007} is gained from the temporally first bit (MSB of the temporally first byte) of the temporally earlier MPEG-TS packet. The next message bit m_{3006} corresponds to the temporally second bit (Bit 6 of the temporally first byte) of the temporally earlier MPEG-TS packet, and so on until bit m_{1504}, which is the temporally last bit (LSB of the temporally last byte) of the temporally earlier MPEG-TS packet. The following 1 504 bits m_{1503} to m_0 are taken in the same manner from the temporally later MPEG-TS packet.

- Multiply the message polynomial $m(x)$ = $m_{3007}x^{3007} + m_{3006}x^{3006} + ... + m_1 x + m_0$ by x^{48} (the coefficients of $m(x)$ for exponents > 3 007 are all set to zero in order to shorten this BCH code; note that this corresponds to temporally preceding the message by 1 039 zeros).

- Divide $x^{48}\,m(x)$ by the BCH generator polynomial

$$g(x) = x^{48} + x^{44} + x^{41} + x^{37} + x^{36} + x^{34} + x^{32} + x^{29} + x^{27} + x^{26} + x^{21} + x^{17} + x^{16} + x^{13} + x^{7} + x^{5} + x^{3} + x + 1$$

  Let $d(x) = d_{47}x^{47} + ... + d_1 x + d_0$ be the remainder.

- Set the outer (i.e. BCH) codeword polynomial $c_o(x) = x^{48}m(x) + d(x)$.

- Calculate the single-parity check bit $p_0 = c_o(x = 1)$ and set the overall codeword polynomial to $c(x) = c_o(x) \cdot x + p_0 = x^{49}m(x) + x \cdot d(x) + p_0$.

Observe that the temporal transmission order of the bits of the codeword c is $(m_{3007}, m_{3006}, ..., m_1, m_0)$ for the message part and $(d_{47}, d_{46}, ..., d_1, d_0, p_0)$ for the parity part, i.e. the order is temporally descending for this BCH codeword-specific indexing. Note that by contrast the indexing of the bits inside the PF infoword is in temporally ascending order, i.e. bit 0 to bit 12 281. Inside the Payload_Packet field of the PF infoword, there are four such message parts (each representing 2 MPEG-TS packets or 376 bytes); the four associated parity parts are transmitted in the field Parity_Parts in the same order.

Table 4: PF infoword format for S-TS stream type 2 (MPEG-TS)

Start bit index	Parameter	Description	Wordsize (bits)	Format	Comments
0	Payload_Packet	Payload packet	12 032	1 504×U8 (1 504 bytes)	
12 032	Parity_Parts	Parity bits for Error Detection or Outer Error Correction Code	196	4×U49	Four times the 49 parity bits of a shortened BCH(3 057,3 008)-code, which each protects 2 MPEG-TS packets.
12 228	RFU	32 bits reserved for future use	32	U32	Helps to bit-align the payload to byte boundaries.
12 260	STS_ID	S-TS ID	8	U8	
12 268	STS_Stream_Type_ID	S-TS stream type identifier	3	U3	Fixed to 2 for MPEG-TS.
12 271	Encap_Ver	Version number of the OPL encapsulation format	3	U3	Fixed to 0.
12 274	CRC_Bits	CRC over the 46 relevant bits of the header	8	U8	The light grey marked bits are included in the CRC check.
		Total length of PFIW	**12 282**		

4.3.4 PF infoword format for S-TS stream type 3 (IP stream)

The format of the IP stream mode is given in table 5. It provides a transparent transmission of IP packets, each having a maximum size (MTU) of 4 095 bytes. Each IP packet to be transmitted is preceded by a header of 2 bytes that is defined in table 6 and contains information about the IP packet format and length.

The payload size of one PF infoword is 1 504 bytes, but the amount of header information needs to be taken into account. Each header consumes 2 bytes of the total payload available.

The address of the first available header within one PF infoword is contained in the parameter `First_Header_Address`. Only this first header is announced; if more than one IP packets are present in one PF infoword, the address of the headers can be incrementally derived from the preceding ones.

If no header was available in this PF infoword, the value `0xFFF` is set to indicate the absence of any header.
See figure 4 for clarification.

If not enough payload is available for transport, the missing bytes are filled with `0xFF` bytes. Any `First_Header_Address` larger than 1 502 is not allowed as splitting of headers is not permitted. In this case, the last byte(s) of the payload packet is (are) padded with `0xFF` bytes.

Additional error correction and detection is performed by using one shortened BCH (3 057, 3 008) code each 376 bytes. Therefore, each PF infoword contains 4 sections of BCH parity of 49 bits each.

As this BCH-code is a systematic code, the parity may be discarded in the receiver if this additional parity check is not desired; however, performance is supposed to degrade in this case. On the contrary, it is a mandatory requirement on the transmitter side to include this code.

The generation of the BCH codeword and the bit format is described in clause 4.3.3.

Table 5: PF infoword format for S-TS stream type 3 (IP stream)

Start bit index	Parameter	Description	Wordsize (bits)	Format	Comment
0	Payload_Packet	Payload packet	12 032	1 504×U8 (1 504 bytes)	See table 6 for further details.
12 032	Parity_Parts	Parity bits for Error Detection or Outer Error Correction Code	196	4×U49	Four times the 49 parity bits of a shortened BCH(3 057,3 008)-code, which each protects 376 payload bytes.
12 228	RFU	20 bits reserved for future use	20	U20	Helps to bit-align the payload to byte boundaries.
12 248	First_Header_Address	Byte address where the first header of the first IP packet can be found; counting is zero-based	12	U12	This value gives the start address of the first header to be found. If no header is present, the address is set to 0xFFF. Any `First_Header_Address` larger than 1 502 needs to be discarded while 1 502 is still allowed.
12 260	STS_ID	S-TS ID	8	U8	
12 268	STS_Stream_Type_ID	S-TS stream type identifier	3	U3	Fixed to 3 for IP stream.
12 271	Encap_Ver	Version number of the OPL encapsulation format	3	U3	Fixed to 0.
12 274	CRC_Bits	CRC over the 46 relevant bits of the header	8	U8	The light grey marked bits are included in the CRC check.
		Total length of PFIW	**12 282**		

Table 6: IP Header definition for each IP packet processed by the OPL encapsulation

Start bit index	Parameter	Description	Wordsize (bits)	Format	Comment
0	IP_Packet_Type	Defines the type of the encapsulated packet	2	U2	The following definitions apply: **0**: reserved **1:** IPv4; **2:** IPv6; **3:** Padding/Stuffing.
2	IP_Packet_Error	Is set if the IP packet is erroneous	1	U1	0 if no error occurred.
3	IP_Packet_Length	Defines the length of the IP Packet (in bytes)	12	U12	This enables a maximum transfer unit (MTU) size of 4 095 bytes.
14	RFU	1 bit reserved for future use	1	U1	
		Total length of one header	**16**		

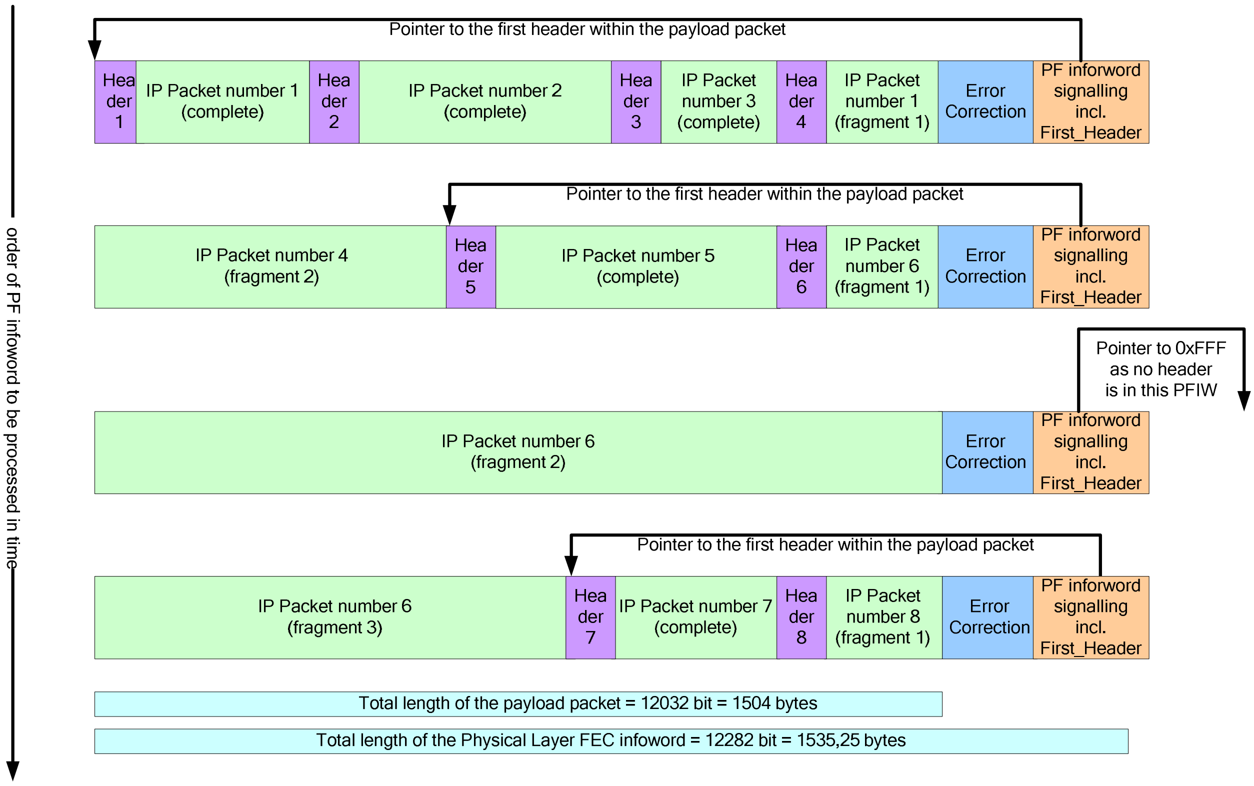

order of PF infoword to be processed in time
Pointer to the first header within the payload packet
Header 1
IP Packet number 1 (complete)
Header 2
IP Packet number 2 (complete)
Header 3
IP Packet number 3 (complete)
Header 4
IP Packet number 1 (fragment 1)
Error Correction
PF inforword signalling incl. First_Header
Pointer to the first header within the payload packet
IP Packet number 4 (fragment 2)
Header 5
IP Packet number 5 (complete)
Header 6
IP Packet number 6 (fragment 1)
Error Correction
PF inforword signalling incl. First_Header
Pointer to 0xFFF as no header is in this PFIW
IP Packet number 6 (fragment 2)
Error Correction
PF inforword signalling incl. First_Header
Pointer to the first header within the payload packet
IP Packet number 6 (fragment 3)
Header 7
IP Packet number 7 (complete)
Header 8
IP Packet number 8 (fragment 1)
Error Correction
PF inforword signalling incl. First_Header
Total length of the payload packet = 12032 bit = 1504 bytes
Total length of the Physical Layer FEC infoword = 12282 bit = 1535,25 bytes

Figure 4: Description of IP packet encapsulation

4.4 PL FEC: turbo code

As PL FEC scheme, the Turbo Code as standardized by the 3GPP2 organization has been chosen.

4.4.1 Interface to OPL encapsulation

The turbo encoder encodes blocks of 12 282 bits, which are referred to as PL FEC information words (PF infoword), for the payload transmission.

For each S-TS, these PF infowords are sequentially input to the turbo encoder after OPL encapsulation.

4.4.2 Turbo encoder

Besides the PF infowords for the S-TS payload of length 12 282 bits, the turbo encoder is also able to encode blocks of 762 bits for the signalling pipe. During encoding, an encoder output tail sequence is added. N_{turbo} is the total number of data excluding the tail bits. The turbo encoder generates N_{turbo}/R encoded data output symbols followed by 6/R tail output symbols, where R is the code rate.

The turbo encoder employs two systematic, recursive, convolutional encoders connected in parallel, with an interleaver, the turbo interleaver, preceding the second recursive convolutional encoder. The two recursive convolutional codes are called the constituent codes of the turbo code. The outputs of the constituent encoders are punctured to achieve the (N_{turbo} + 6)/R output symbols.

A common constituent code is used for all turbo code rates. The transfer function for the constituent code is:

$$G(D) = \begin{bmatrix} 1 & \frac{n_0(D)}{d(D)} & \frac{n_1(D)}{d(D)} \end{bmatrix}$$

where $d(D) = 1 + D^2 + D^3$, $n_0(D) = 1 + D + D^3$, and $n_1(D) = 1 + D + D^2 + D^3$.

The turbo encoder generates an output symbol sequence that is identical to the one generated by the encoder shown in figure 5. Initially, the states of the constituent encoder registers in this figure are set to zero. Then, the constituent encoders are clocked with the switches in the positions noted.

Using the turbo encoder, the constituent encoder output symbols are generated by clocking the constituent encoders N_{turbo} times with the switches in the up positions and puncturing as specified in table 7. Within a puncturing pattern, a "0" means that the symbol shall be deleted and a "1" means that a symbol shall be passed. The puncturing patterns shall be read from left to right and continuously from one text line to the next one. The patterns are displayed with a partitioning into groups of 5 symbols. The 5 symbols of a group represent the outputs X Y_0 Y_1 Y'_0 Y'_1 of the encoder shown in figure 5, respectively. Each puncturing pattern consists of one such group or of a sequence of several groups. The displayed pattern is repeated cyclically, until 12 282 groups have been processed (one group per infoword bit). Hence, the last period of the pattern remains incomplete for some puncturing patterns.

According to table 7, some examples for puncturing are given.

The turbo encoder shall generate symbols for rate 1/2 turbo codes as follows:

- The symbols output by the encoder for even-indexed data bit periods shall be XY_0.
- The symbols output by the encoder for odd-indexed data bit periods shall be XY'_0.

The turbo encoder shall generate symbols for rate 1/3 turbo codes as follows:

- The symbols output by the encoder for all data bit periods shall be $XY_0Y'_0$.

The turbo encoder shall generate symbols for rate 1/4 turbo codes as follows:

- The symbols output by the encoder for even-indexed data bit periods shall be $XY_0Y_1Y'_1$.
- The symbols output by the encoder for odd-indexed data bit periods shall be $XY_0Y'_0Y'_1$.

The turbo encoder shall generate symbols for rate 1/5 turbo codes as follows:

- The symbols output by the encoder for all data bit periods shall be $XY_0Y_1Y'_0Y'_1$.

Symbol repetition is not used in generating the encoded data output symbols.

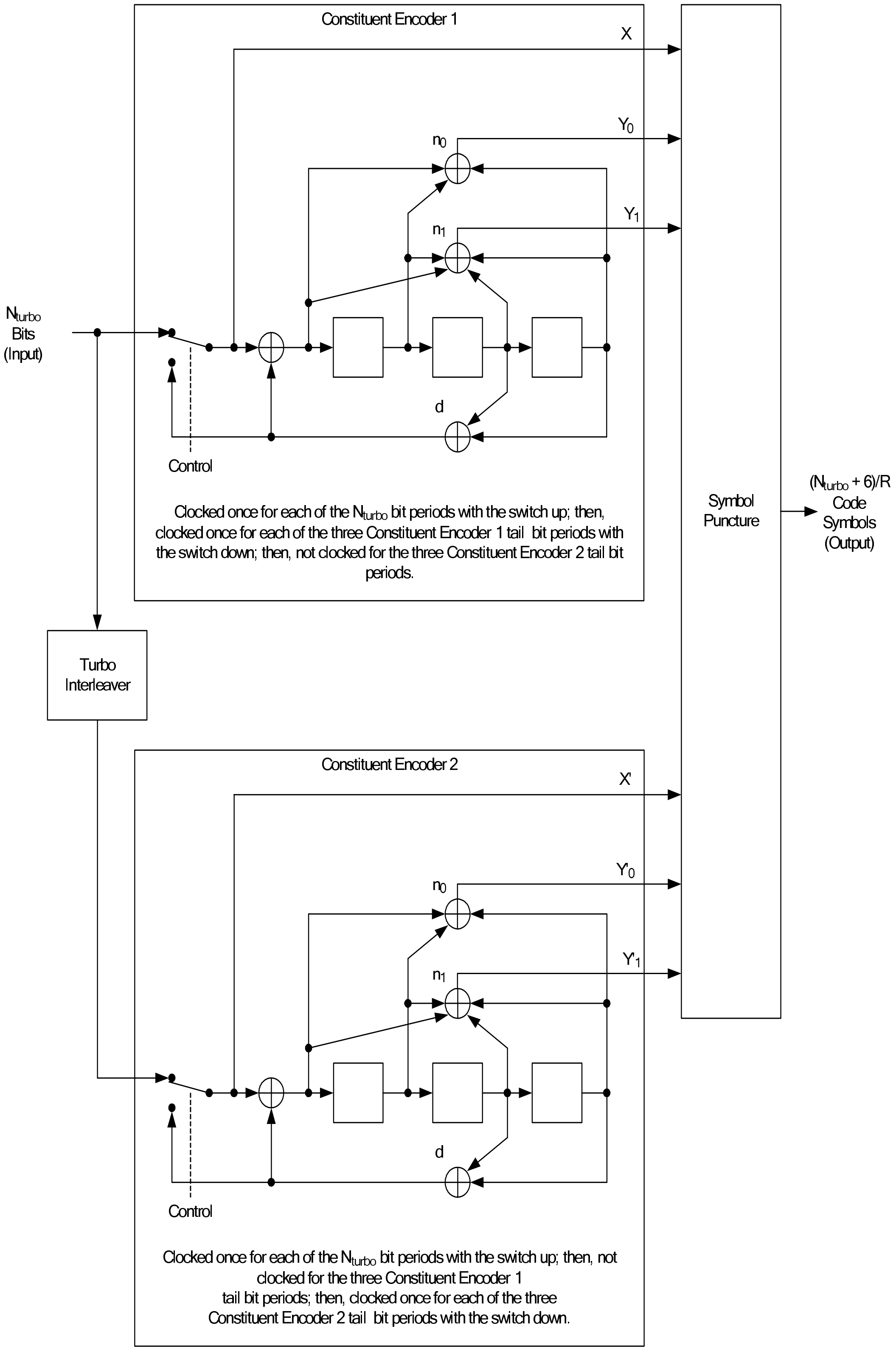

Constituent Encoder 1
X
Y_0
n_0
Y_1
n_1
N_{turbo} Bits (Input)
d
Control
Clocked once for each of the N_{turbo} bit periods with the switch up; then, clocked once for each of the three Constituent Encoder 1 tail bit periods with the switch down; then, not clocked for the three Constituent Encoder 2 tail bit periods.
Symbol Puncture
$(N_{turbo} + 6)/R$ Code Symbols (Output)
Turbo Interleaver
Constituent Encoder 2
X'
Y'$_0$
n_0
Y'$_1$
n_1
d
Control
Clocked once for each of the N_{turbo} bit periods with the switch up; then, not clocked for the three Constituent Encoder 1 tail bit periods; then, clocked once for each of the three Constituent Encoder 2 tail bit periods with the switch down.

Figure 5: Turbo encoder

Table 7: Puncturing patterns for the turbo encoder during the data bit periods

Punct_Pat_ID	Code Rate	Pattern Name	Puncturing Pattern (X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; etc.)
0	1/5	Standard	1;1;1;1;1
1	2/9	Standard	1;0;1;1;1; 1;1;1;1;1; 1;1;1;0;1; 1;1;1;1;1
2	1/4	Standard	1;1;1;0;1; 1;1;0;1;1
3	2/7	Standard	1;0;1;0;1; 1;0;1;1;1; 1;0;1;0;1; 1;1;1;0;1
4	3/10	Standard	1;1;0;1;0; 1;1;0;1;0; 1;1;0;1;0; 1;1;0;1;0; 1;1;0;1;0; 1;1;1;1;1
5	1/3	Standard	1;1;0;1;0
6	1/3	Complementary1	1;0;1;0;1
7	3/8	Standard	0;1;0;1;0; 1;1;0;1;0; 1;1;0;1;0
8	3/8	Complementary1	1;0;1;0;1; 0;0;1;0;1; 1;0;1;0;1

9	2/5	Standard	1;0;0;0;0; 1;0;1;0;1; 0;0;1;0;1;
			1;0;1;0;1; 1;0;1;0;1; 0;0;1;0;1;
			1;0;1;0;1; 1;0;1;0;1; 0;0;1;0;1;
			1;0;1;0;1; 1;0;1;0;1; 0;0;1;0;1

10	2/5	Complementary1	1;1;0;1;0; 0;1;0;1;0; 1;1;0;1;0;
			1;1;0;1;0; 0;1;0;1;0; 1;0;0;0;0;
			1;1;0;1;0; 0;1;0;1;0; 1;1;0;1;0;
			1;1;0;1;0; 0;1;0;1;0; 1;1;0;1;0

11	3/7	Standard	1;0;0;0;0; 1;1;0;1;0; 0;1;0;1;0; 1;1;0;1;0; 0;1;0;1;0; 1;1;0;1;0
12	3/7	Complementary1	1;0;1;0;1; 0;0;1;0;1; 1;0;1;0;1; 1;0;0;0;0; 1;0;1;0;1; 0;0;1;0;1
13	1/2	Standard	1;1;0;0;0; 1;0;0;1;0
14	1/2	Complementary1	1;0;0;1;0; 1;1;0;0;0
15	1/2	Complementary2	1;0;1;0;0; 1;0;0;0;1
16	3/5	Standard	1;0;0;0;0; 1;0;0;1;0; 1;1;0;0;0
17	3/5	Complementary1	1;0;0;1;0; 1;1;0;0;0; 1;0;0;0;0
18	3/5	Complementary2	1;1;0;0;0; 1;0;0;0;0; 1;0;0;1;0
19	2/3	Standard	1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;1;0;1
20	2/3	Complementary1	1;0;0;0;0; 1;0;1;0;1; 1;0;0;0;0; 1;0;0;0;0
21	2/3	Complementary2	1;0;0;0;0; 1;0;0;0;0; 1;0;1;0;1; 1;0;0;0;0
22	3/4	Standard	1;0;0;0;0; 1;0;0;0;0; 1;1;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;1;0
23	3/4	Complementary1	1;0;0;0;0; 1;0;0;1;0; 1;0;0;0;0; 1;0;0;0;0; 1;1;0;0;0; 1;0;0;0;0
24	3/4	Complementary2	1;1;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;1;0; 1;0;0;0;0; 1;0;0;0;0
25	6/7	Standard	1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;1;0;0; 1;0;0;0;1

26	6/7	Complementary1	1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0;
			1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0;
			1;0;1;0;0; 1;0;0;0;1; 1;0;0;0;0;
			1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0

Punct_Pat_ID	Code Rate	Pattern Name	Puncturing Pattern (X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; etc.)

27	6/7	Complementary2	1;0;0;0;0; 1;0;0;0;0; 1;0;1;0;0;
			1;0;0;0;1; 1;0;0;0;0; 1;0;0;0;0;
			1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0;
			1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0

28 to 63	RFU		

4.4.3 Turbo code termination

The turbo encoder shall generate tail output symbols following the encoded data output symbols. This tail output symbol sequence shall be identical to the one generated by the encoder shown in figure 5. The tail output symbols are generated after the constituent encoders have been clocked N_{turbo} times with the switches in the up position. The first tail output symbols are generated by clocking Constituent Encoder 1 three times with its switch in the down position while Constituent Encoder 2 is not clocked and puncturing the resulting constituent encoder output symbols. The last tail output symbols are generated by clocking Constituent Encoder 2 three times with its switch in the down position while Constituent Encoder 1 is not clocked and puncturing the resulting constituent encoder output symbols. The constituent encoder outputs for each bit period shall be output in the sequence X, Y_0, Y_1, X′, Y'_0, Y'_1 with the X output first.

The tail output symbol puncturing shall be as specified in table 8. Within a puncturing pattern, a "0" means that the symbol shall be deleted and a "1" means that a symbol shall be passed. The tail puncturing patterns shall be read from left to right and continuously from one text line to the next one. The patterns are displayed with a partitioning into six groups of 5 symbols. The 5 symbols of a group represent either the outputs X; Y_0; Y_1; Y'_0; and Y'_1 of the encoder shown in figure 5 for the first three groups, or X'; Y_0; Y_1; Y'_0; and Y'_1 for the last three groups of the pattern, respectively.

A 2 or a 3 means that two or three copies of the symbol shall be passed. E.g. for rate 1/5 turbo codes, the tail output symbols for each of the first three tail bit periods shall be $XXXY_0Y_1$, and the tail output symbols for each of the last three tail bit periods shall be $X'X'X'Y'_0Y'_1$.

Table 8: Puncturing and symbol repetition patterns for the turbo encoders during the tail bit periods

Punct_Pat_ID	Code Rate	Pattern Name	Tail Puncturing Pattern (X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; Y_1; Y'_0; Y'_1; X'; Y_0; Y_1; Y'_0; Y'_1; X'; Y_0; Y_1; Y'_0; Y'_1; X'; Y_0; Y_1; Y'_0; Y'_1)
0	1/5	Standard	3;1;1;0;0; 3;1;1;0;0; 3;1;1;0;0; 3;0;0;1;1; 3;0;0;1;1; 3;0;0;1;1
1	2/9	Standard	3;1;1;0;0; 3;1;1;0;0; 2;1;1;0;0; 2;0;0;1;1; 2;0;0;1;1; 3;0;0;1;1
2	1/4	Standard	2;1;1;0;0; 2;1;1;0;0; 2;1;1;0;0; 2;0;0;1;1; 2;0;0;1;1; 2;0;0;1;1
3	2/7	Standard	1;1;1;0;0; 2;1;1;0;0; 2;1;1;0;0; 2;0;0;1;1; 1;0;0;1;1; 1;0;0;1;1
4	3/10	Standard	1;1;1;0;0; 1;1;1;0;0; 2;1;1;0;0; 1;0;0;1;1; 1;0;0;1;1; 2;0;0;1;1
5	1/3	Standard	2;1;0;0;0; 2;1;0;0;0; 2;1;0;0;0; 2;0;0;1;0; 2;0;0;1;0; 2;0;0;1;0
6	1/3	Complementary1	2;0;1;0;0; 2;0;1;0;0; 2;0;1;0;0; 2;0;0;0;1; 2;0;0;0;1; 2;0;0;0;1
7	3/8	Standard	1;1;0;0;0; 1;1;1;0;0; 1;1;1;0;0; 1;0;0;1;0; 1;0;0;1;1; 1;0;0;1;1

Punct_Pat_ID	Code Rate	Pattern Name	**Tail Puncturing Pattern** **(X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; Y_1; Y'_0; Y'_1; X; Y_0; Y_1; Y'_0; Y'_1;** **X'; Y_0; Y_1; Y'_0; Y'_1; X'; Y_0; Y_1; Y'_0; Y'_1; X'; Y_0; Y_1; Y'_0; Y'_1)**
8	3/8	Complementary1	1;0;1;0;0; 1;1;1;0;0; 1;1;1;0;0; 1;0;0;0;1; 1;0;0;1;1; 1;0;0;1;1
9	2/5	Standard	1;1;1;0;0; 1;1;1;0;0; 1;0;1;0;0; 1;0;0;1;1; 1;0;0;1;1; 1;0;0;0;1
10	2/5	Complementary1	1;1;1;0;0; 1;1;0;0;0; 1;1;1;0;0; 1;0;0;1;1; 1;0;0;1;0; 1;0;0;1;1
11	3/7	Standard	1;1;0;0;0; 1;1;0;0;0; 1;1;1;0;0; 1;0;0;1;0; 1;0;0;1;0; 1;0;0;1;1
12	3/7	Complementary1	1;0;1;0;0; 1;0;1;0;0; 1;1;1;0;0; 1;0;0;0;1; 1;0;0;0;1; 1;0;0;1;1
13	1/2	Standard	1;1;0;0;0; 1;1;0;0;0; 1;1;0;0;0; 1;0;0;1;0; 1;0;0;1;0; 1;0;0;1;0
14	1/2	Complementary1	1;0;1;0;0; 1;0;1;0;0; 1;0;1;0;0; 1;0;0;0;1; 1;0;0;0;1; 1;0;0;0;1
15	1/2	Complementary2	1;1;0;0;0; 1;1;0;0;0; 1;1;0;0;0; 1;0;0;1;0; 1;0;0;1;0; 1;0;0;1;0
16	3/5	Standard	1;0;1;0;0; 1;0;0;0;0; 1;1;0;0;0; 1;0;0;0;0; 1;0;0;1;0; 1;0;0;0;1
17	3/5	Complementary1	1;0;0;0;0; 1;1;0;0;0; 1;0;1;0;0; 1;0;0;1;0; 1;0;0;0;1; 1;0;0;0;0
18	3/5	Complementary2	1;1;0;0;0; 1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;1; 1;0;0;0;0; 1;0;0;1;0
19	2/3	Standard	1;0;0;0;0; 1;0;1;0;0; 1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;1; 1;0;0;0;1
20	2/3	Complementary1	1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;1; 1;0;0;0;0; 1;0;0;0;0
21	2/3	Complementary2	1;0;1;0;0; 1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;1; 1;0;0;0;1; 1;0;0;0;0
22	3/4	Standard	1;0;0;0;0; 1;0;0;0;0; 1;1;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;1;0
23	3/4	Complementary1	1;0;0;0;0; 1;1;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;1;0; 1;0;0;0;0
24	3/4	Complementary2	1;1;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;1;0; 1;0;0;0;0; 1;0;0;0;0
25	6/7	Standard	1;0;0;0;0; 1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;1
26	6/7	Complementary1	1;0;1;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;1; 1;0;0;0;0
27	6/7	Complementary2	1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0; 1;0;0;0;0
28 to 63	RFU		

4.4.4 Turbo Interleavers

The turbo interleaver, which is part of the turbo encoder, shall block interleave the N_{turbo} input bits.

The turbo interleaver shall be functionally equivalent to an approach where the entire sequence of turbo interleaver input bits are written sequentially into an array at a sequence of addresses, and then the entire sequence is read out from a sequence of addresses that are defined by the procedure described below.

Let the sequence of input addresses be from 0 to N_{turbo} - 1. Then, the sequence of interleaver output addresses shall be equivalent to those generated by the procedure illustrated in figure 6 and described below:

1) Determine the turbo interleaver parameter, n, where n is the smallest integer such that $N_{turbo} \le 2^{(n + 5)}$. Table 9 gives this parameter.

2) Initialize an (n + 5) - bit counter to 0.

3) Extract the n most significant bits (MSBs) from the counter and add one to form a new value. Then, discard all except the n least significant bits (LSBs) of this value.

4) Obtain the n-bit output of the table lookup defined in table 10 with a read address equal to the five LSBs of the counter. Note that this table depends upon the value of n.

5) Multiply the values obtained in Steps 3 and 4, and discard all except the n LSBs.

6) Bit-reverse the five LSBs of the counter.

7) Form a tentative output address that has its MSBs equal to the value obtained in Step 6 and its LSBs equal to the value obtained in Step 5.

8) Accept the tentative output address as an output address if it is less than N_{turbo}; otherwise, discard it.

9) Increment the counter and repeat Steps 3 through 8 until all N_{turbo} interleaver output addresses are obtained.

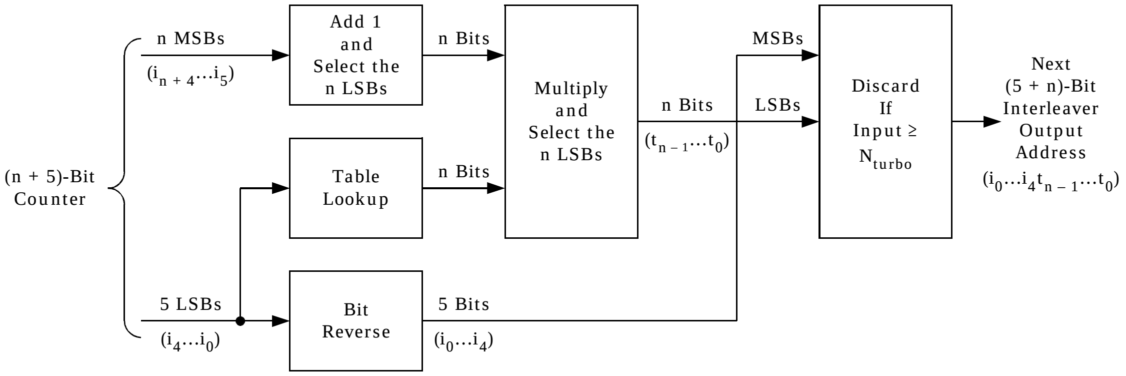

(n + 5)-Bit Counter
n MSBs
($i_{n+4}\ldots i_5$)
5 LSBs
($i_4\ldots i_0$)
Add 1 and Select the n LSBs
Table Lookup
Bit Reverse
n Bits
n Bits
5 Bits
($i_0\ldots i_4$)
Multiply and Select the n LSBs
n Bits
($t_{n-1}\ldots t_0$)
MSBs
LSBs
Discard If Input ≥ N_{turbo}
Next (5 + n)-Bit Interleaver Output Address
($i_0\ldots i_4 t_{n-1}\ldots t_0$)

Figure 6: Turbo interleaver output address calculation procedure

Table 9: Turbo interleaver parameters

Turbo Interleaver Block Size N_{turbo}	Turbo Interleaver Parameter n
762	5
12 282	9

Table 10: Turbo Interleaver lookup table definition

Table Index	n = 5 Entries	n = 9 Entries
0	27	13
1	3	335
2	1	87
3	15	15
4	13	15
5	17	1
6	23	333
7	13	11
8	9	13
9	3	1
10	15	121
11	3	155
12	13	1
13	1	175
14	13	421
15	29	5
16	21	509
17	19	215
18	1	47
19	3	425
20	29	295
21	17	229
22	25	427
23	29	83
24	9	409
25	13	387
26	23	193
27	13	57
28	13	501
29	1	313
30	13	489
31	13	391

4.4.5 Output of turbo encoder

For any S-TS, the encoded bits form a block of 12 288/R bits, where R is the selected code rate. This block is referred to as the PL FEC codeword (PF codeword). The output after encoding the signalling pipe form a block of 3 840 bits.

4.4.6 FEC Parameter signalling

The parameter `Punct_Pat_ID`, which specifies the chosen puncturing scheme (which implicitly also defines the code rate) is transmitted in the signalling pipe. Table 11 applies.

Table 11: Definition of turbocode code rate and puncturing pattern using `Punct_Pat_ID`

Punct_Pat_ID	Turbocode code rate	Puncturing pattern
0	1/5	standard
1	2/9	standard
2	1/4	standard
3	2/7	standard
4	3/10	standard
5	1/3	standard
6	1/3	complementary 1
7	3/8	standard
8	3/8	complementary 1
9	2/5	standard
10	2/5	complementary 1
11	3/7	standard
12	3/7	complementary 1
13	1/2	standard
14	1/2	complementary 1
15	1/2	complementary 2
16	3/5	standard
17	3/5	complementary 1
18	3/5	complementary 2
19	2/3	standard
20	2/3	complementary 1
21	2/3	complementary 2
22	3/4	standard
23	3/4	complementary 1
24	3/4	complementary 2
25	6/7	standard
26	6/7	complementary 1
27	6/7	complementary 2
28 to 63	RFU	RFU

4.4.7 Diversity combining

Table 11 displays for several code rates more than one puncturing pattern. The "standard" pattern should be used primarily for any code rate. The "complementary" patterns (number 1 or 2) can be used for diversity combining, whenever the same PF infoword shall be transmitted over more than one propagation channel to the same terminal. The combination of a standard pattern with one or more complementary patterns leads to a combined Turbo code of lower code rate and, moreover, a higher coding gain.

In principle, any two (or even more) of the puncturing patterns of table 11 can be combined with each other, whether they are standard, complementary 1 or complementary 2. However, the overlap of the selected patterns, i.e. the number of transmitted (non-punctured) symbols that are common to these patterns according to table 7, should be kept as low as possible.

4.4.8 FEC Parameters for the signalling pipe

The signalling pipe always uses the parameter Punct_Pat_ID = 0, i.e. code rate 1/5.

4.5 Mixer

The mixer is a block interleaver that works on a codeword basis. Its task is to re-order the codeword. This is especially helpful in scenarios where the reception suffers from bursty blockages, but also helps to achieve fast access times in case of good reception conditions. Any bursty loss of data (wanted or unwanted) is then spread on the whole code word equally instead of having bursty erasures which in turn helps the FEC decoder, as the losses can then be regarded as "random puncturing".

The input of the mixer is the output of the turbo encoder, i.e. a stream of PF codewords belonging to one S-TS. The output is referred to as mixed codewords.

The following formula has to be applied to the PF codewords where `a[i]` denotes the input of the mixer (equal to the output of the FEC), and `b[i]` denotes the output of the mixer, at the bit position `i`, respectively.

```
b[i] = a[ (CILM_Inc×i) mod Codeword_Len ];
```

with `CILM_Inc` denoting the mixer increment as defined in table 12, `mod` denoting the modulo operation, and `Codeword_Len` denoting the PL codeword length, also defined in table 12. The notation is 0-based, and the range of `i` is `[0; codewordLen-1]`.

As the mixer increment only depends on the PL codeword length and hence on the code rate only, it is not signalled additionally but has to be derived from the parameter `Punct_Pat_ID`.

Table 12: Mixer address increment definition

Code Rate	1/5	2/9	1/4	2/7	3/10	1/3	3/8	2/5	3/7	1/2	3/5	2/3	3/4	6/7
Codeword_Len	61 440	55 296	49 152	43 008	40 960	36 864	32 768	30 720	28 672	24 576	20 480	18 432	16 384	14 336
CILM_Inc	247	245	221	197	209	185	179	167	167	157	149	125	139	113

The signalling pipe always uses CILM_Inc = 61.

4.6 Segmenter and Slot demultiplexer

The segmenter's input is a stream of mixed codewords belonging to one S-TS. The segmenter has the following task: Chop the mixed codewords into "Interleaver Units" (IU) - each of length 512 bits - which are later processed in the dispersers.

Observe that for any configurable code rate, the PF codeword length is an integer multiple of 2 048 codebits. This granule is termed a "Capacity Unit" (CU). Therefore, a codeword can always be segmented into an integer multiple of 4 `IUs`.

For informative purpose, table 13 denotes the number of IUs and CUs per PF codeword (`IU_Per_CW` and `CU_Per_CW`).

Table 13: Number of CU and number of IU per code word

Code Rate	**Codeword length (in bits)**	**Number of CU per Codeword (`CU_Per_CW`)**	**Number of IU per Codeword (`IU_Per_CW`)**
1/5	61 440	30	120
2/9	55 296	27	108
1/4	49 152	24	96
2/7	43 008	21	84
3/10	40 960	20	80
1/3	36 864	18	72
3/8	32 768	16	64
2/5	30 720	15	60
3/7	28 672	14	56
1/2	24 576	12	48
3/5	20 480	10	40
2/3	18 432	9	36
3/4	16 384	8	32
6/7	14 336	7	28

After the chopping, a demultiplexer distributes these `IUs` codeword-wise to slots. A slot is a sub-stream of IUs that is processed by its individual disperser. The slots are sub-partitions inside a pipe of one frame, and the number of slots is hence determined by the width `Pipe_Width_CUs` of the pipe and the width `CU_Per_CW` of each slot (both in terms of CUs):

Pipe_Width_Slots = Pipe_Width_CUs / CU_Per_CW

Each slot is fed with all IUs of one codeword. The Pipe_Width_Slots slots of a pipe are cyclically fed with codewords from the S-TS of that pipe. The number of consecutive codewords of an S-TS that are assigned to the slots is denoted by STS_Width_Slots. This value is individual for every S-TS and may vary from S-TS to S-TS. The slot demultiplexer feeds exactly all IUs belonging to the same codeword to one slot, i.e. to the specific disperser for that slot. Note that neither the segmenter nor the demultiplexer change the order of the IUs inside one codeword/one slot. When all IUs of one codeword have been demultiplexed to one slot, the demultiplexer switches to the next slot and feeds it with the IUs of the next codeword. After Pipe_Width_Slots codewords have been demultiplexed to the Pipe_Width_Slots slots, the demultiplexer starts with the first slot again and feeds it with the Pipe_Width_Slots+1st codeword. This behaviour is repeated periodically. When STS_Width_Slots consecutive codewords of the S-TS have been fed to the slots, the slot demultiplexer continues with the codewords from the next S-TS in the pipe.

4.7 Disperser

For specifying memory requirements for standard-compliant receivers, examples of disperser profiles are given below that the terminals must be able to handle.

Standard-compliant receivers must be able to decode the payload in the following cases:

- S-TS payload throughput 3,2 Mbit/s (max. allowed S-TS throughput), Punct_Pat_ID = 2, i.e. code rate 1/4, uniform interleaving over 10 s, over static AWGN channel.

- S-TS payload throughput 1,6 Mbit/s (typical for a multiplex of 4 video services), Punct_Pat_ID = 0 i.e. code rate 1/5, uniform interleaving over 16 s, over static AWGN channel.

- S-TS payload throughput 0,4 Mbit/s (typical throughput for one video service), Punct_Pat_ID = 0 i.e. code rate 1/5, interleaving, where 50 % of the data is transmitted non-delayed and 50 % with a delay of 64 s, over static AWGN channel.

Performance requirements for the receivers for achieving successful decoding (i.e. WER < 10^{-3}) are stated in the Implementation Guidelines document.

The dispersers work on a slot basis, i.e. each slot has its individual disperser, such that there are Pipe_Width_Slots parallel dispersers for one pipe. The disperser distributes the IUs of one codeword over time by interleaving them inside one slot with IUs of other codewords, which are demultiplexed to the same slot.

The core function of the disperser is an irregular convolutional block interleaver, whose smallest granularity is one IU. Figure 7 displays the principle structure of the disperser. The number of tapped delay lines (referred to as taps) is IU_Per_CW, i.e. there is one tap for each IU of a codeword. In the sequel, Tap_Delay[i] denotes the delay (in terms of IUs) of tap i, with i from 0 to IU_Per_CW-1. The vector Tap_Delay[0] to Tap_Delay[IU_Per_CW-1] is referred to as the disperser profile. The disperser profile is configured over the signalling pipe as described in clause 4.10.1. It has the property that it is periodic with a period of 32.

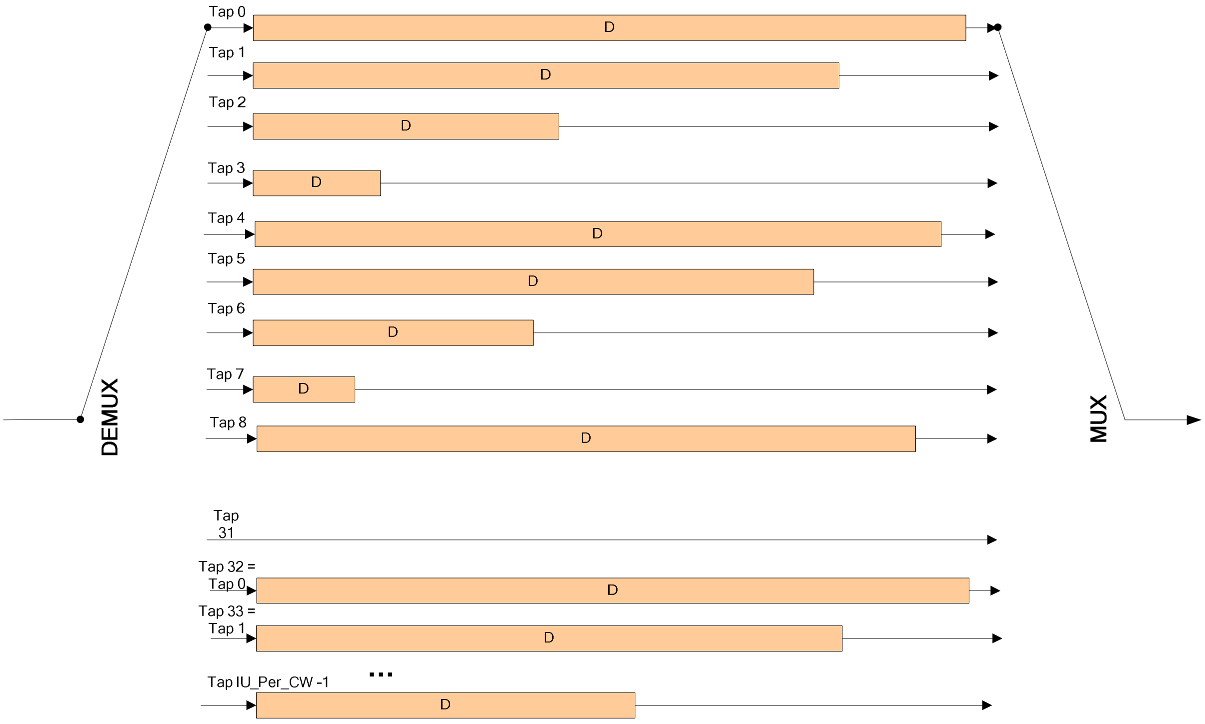


Figure 7: Example of a disperser profile

4.8 Collector

For each pipe of the current C-TS frame, the collector assembles its content by `IU`-wise reading the output of the `Pipe_Width_Slots` slots of the pipe, and multiplexing these `IUs` together.

To obtain the possibility of PL time slicing or a high degree of diversity even in case of high code rates and high pipe throughput, two modes are available to collect the `IUs` filling one pipe after the dispersers, depending on the parameter `Regrouping_Flag` transmitted inside the signalling pipe:

1) `Regrouping_Flag = 0` (Regrouping off): See figure 8. The collector multiplexes the disperser outputs with a granularity of one slot, i.e. it first reads all `IU_Per_CW IUs` from slot 0 (i.e. from the disperser associated with this slot), then it reads `IU_Per_CW IUs` from slot 1 etc. until slot number `Pipe_Width_Slots - 1`. Then the considered pipe inside the current C-TS frame is complete. This option hence preserves the slot structure of the dispersers in the transmitted pipe.

2) `Regrouping_Flag = 1` (Regrouping on): See figure 8. The collector multiplexes the disperser outputs with a granularity of one IU, i.e. it first reads the first `IU` from slot 0 (i.e. from the disperser associated with this slot), then it reads the first `IU` from slot 1 etc. until slot number `Pipe_Width_Slots - 1`. Then it continues with the second `IU` of all `Pipe_Width_Slots` slots in the same way etc. When the last `IU` (number `IU_Per_CW-1`) has been read from slot number `Pipe_Width_Slots - 1`, the considered pipe inside the current C-TS frame is complete. This option distributes the `IUs` of one slot maximally over the transmitted pipe. This option is therefore particularly suitable for enlarging the time diversity inside one transmitted codeword, when the disperser alone cannot provide a sufficiently high degree of time diversity (e.g. it disperses only over very few C-TS frames).

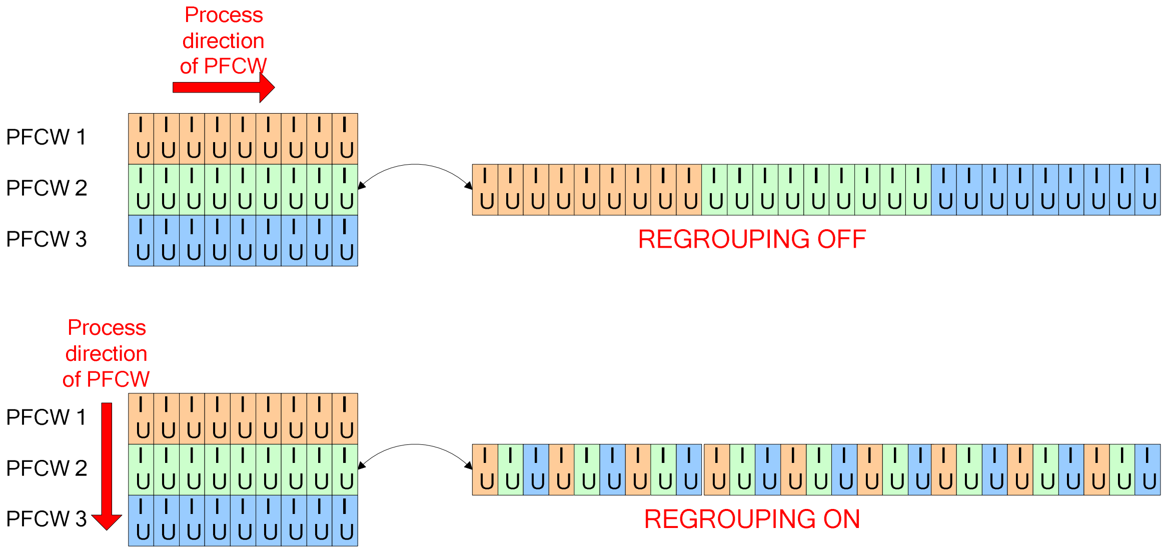


Figure 8: Demonstration of regrouping

4.9 C-TS multiplexer

The C-TS multiplex consists of Capacity Units (CU), where each of the CU consists of 2 048 `C_Chips`. The partitioning of all available CUs of a C-TS frame into pipes is described in clause 4.10.2.

The current C-TS frame is assembled by concatenating the SOF preamble, the signalling pipe, and all payload pipes in the order 0, 1, 2, etc. If this concatenation does not fill a complete C-TS frame, then the remaining CUs are filled by zero-padding.

Figure 9 shows the sub-partitioning of a C-TS frame into its smaller units and into logical content.

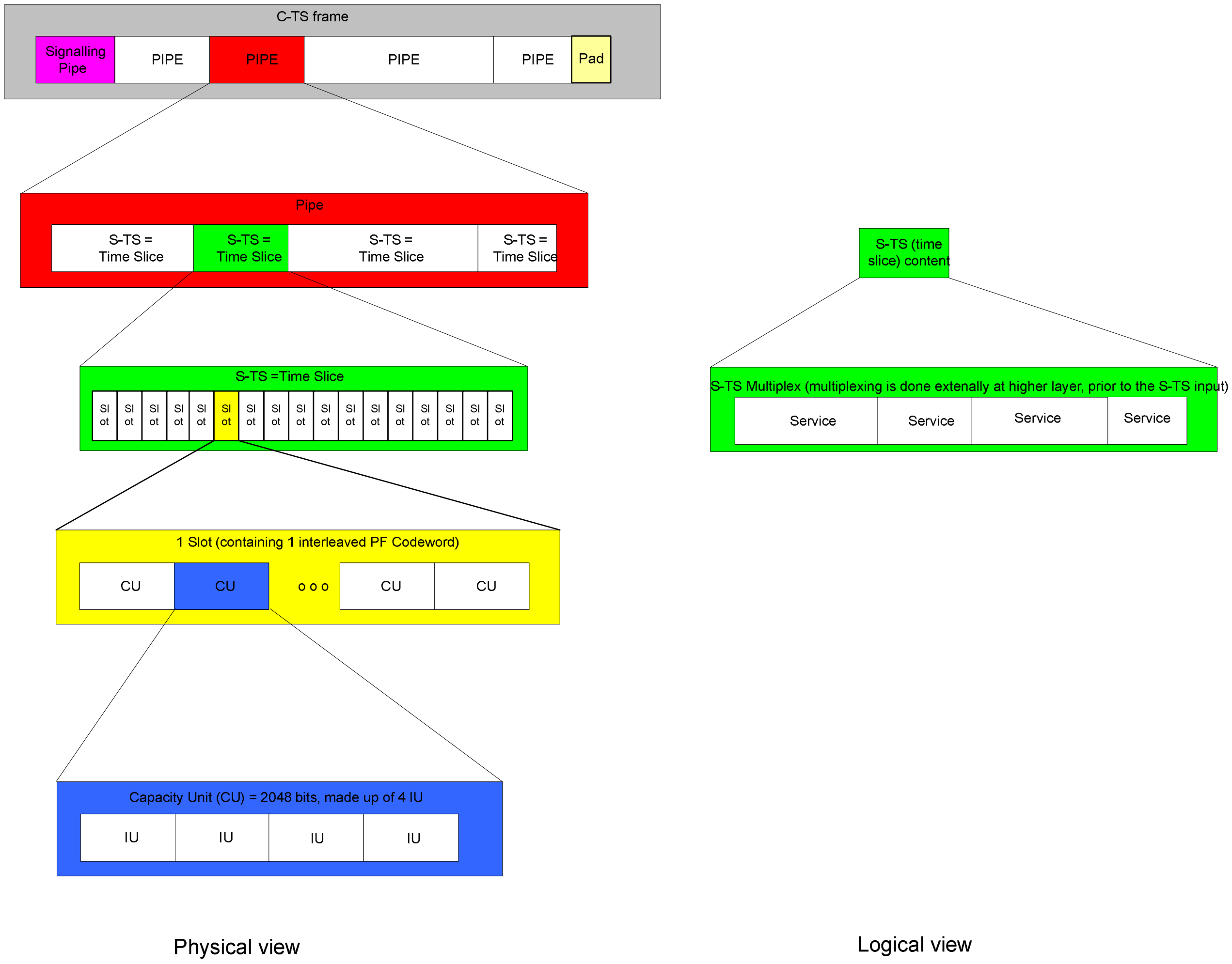


Figure 9: Overview on the C-TS multiplex

4.10 Configuration of the OPL

4.10.1 Signalling pipe

4.10.1.1 Encoding and interleaving of signalling pipe

The infoword format of the signalling pipe is specified in table 14. The infoword length is 762 bits. The signalling pipe uses a coderate of 1/5. The puncturing pattern `Punct_Pat_ID = 0` used in the turbo encoder for the signalling pipe is defined in table 7 and table 8.

The codeword length of the signalling pipe is 3 840 codebits. The signalling pipe codeword is mixed in the mixer with the parameter `CILM_Inc = 61`.

The signalling pipe is not dispersed, i.e. there is only one slot, the disperser profile is with delay 0 in all taps and `Regrouping_Flag = 0`.

A Start-Of-Frame (SOF) preamble of length 256 bits is prepended, such that the concatenation of SOF preamble and the mixed codeword of the signalling pipe occupy exactly 2 CUs.

4.10.1.2 SOF Preamble

The SOF preamble is defined as follows and transmitted row-wise with 0x53 being the first byte to be transmitted:

```
SOF = {0x534f4620, 0x70726561, 0x6d626c65, 0x3a204e65,
       0x7720432d, 0x54532066, 0x72616d65, 0xfffffffc}
```

This represents the ASCII string `"SOF preamble: New C-TS frame"` plus four appended padding bytes, which equalize the number of zeros and ones in the SOF preamble. Otherwise, CUs filled with zeros could cause false detection.

In the sequence of C-TS frames, the SOF preamble is alternately transmitted in the format specified above and in bit-wise inverted form, i.e. every bit value 0 is replaced by value 1 and vice versa. This alternation is a protection against repetitive patterns in the payload matching the SOF preamble by random.

4.10.1.3 Format of the signalling pipe infoword

For detection of transmission errors, the signalling pipe infoword contains 12 parity bits of a Cyclic Redundancy Check (CRC). The CRC polynomial is $x^{12} + x^{11} + x^3 + x^2 + x + 1$ for all S-TS stream types. The calculation of the CRC is described in annex B.

Table 14: Signalling pipe infoword format

Signalling pipe infoword format		
Contents	**No. of bits**	**Comments**
Parameters for the C-TS frame and pipe multiplex	56	See table 15 for further details.
Individual parameters for pipe 1	48	See table 16 for further details.
…		
Individual parameters for pipe N-1	48	See table 16 for further details.
Individual parameters for pipe N (tail pipe)	16	See table 16b for further details.
Individual parameters for S-TS 0	16	See table 18 for further details.
…		
Individual parameters for S-TS M	16	See table 18 for further details.
Individual parameters for disperser section 0 of pipe 1	12	See table 19 for further details.
…		
Individual parameters for disperser section L of pipe 1	12	See table 19 for further details.
Individual parameters for disperser section 0 of pipe 2	12	See table 19 for further details.
…		
Individual parameters for disperser section K of pipe N	12	See table 19 for further details.
Zero-padding of the unused bits	750 minus sum of all bits above	Need to be filled up with zeros to reach 750 bits.
Parity bits of CRC-12 code over preceding bits	12	CRC over the complete 750 parameter bits and zero-padding (indicated with light grey background in the "No. of bits" column of this table).
Total infoword length of signalling pipe	**762**	

Table 15: Parameters for the C-TS frame and the pipe multiplex

Parameters for the C-TS frame and the pipe multiplex					
Start bit index	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comments**
0	Sig_Pipe_Ver	Version number of the signalling pipe infoword format	4	U4	Fixed to 0.
4	Reconfig_Flag	Indicator that this signalling pipe infoword contains the *next* pipe configuration	1	U1	Used when a reconfiguration is approaching, which is count down by the Event_Countdown. This reconfiguration announcement is only used for the pipe multiplex. Reconfiguration of the S-TS scheduling inside the pipes are not announced by this indicator, except when new S-TS are added to or removed from the C-TS mux. This flag is not necessarily active for all C-TS frames before a reconfiguration. See comment below.
5	Event_Countdown	Countdown (in C-TS frames) until one of the following events occurs: (a) the next pipe configuration becomes active, (b) the next S-TS schedule becomes active, or (c) a resynchronization needs to be carried out; if 0: no reconfiguration/rescheduling/resynchronization approaching	11	U11	The next pipe configuration can be announced before using Reconfig_Flag and the next S-TS schedule can be announced before using Reschedule_Flag; announcement up to 2 047 frames in advance; i.e. over 14 minutes for 432ms frame duration. About the counting convention: see comment below.
16	Operator_ID	Unique identifier for the network operator	8	U8	
24	Resync_Flag	Indicator that the Physical Layer (IPL + OPL) must get ready for resynchronization for this C-TS multiplex	1	U1	This flag is set, e.g. when a transmitter hand-over is imminent. The re-synchronization shall be carried out when the Event_Countdown reaches 0.
25	Reschedule_Flag	Indicator that this signalling pipe infoword contains the next S-TS schedule	1	U1	Used when a rescheduling is approaching, which is count down by the Event_Countdown. This flag is not necessarily active for all C-TS frames before a rescheduling. See comment below.
26	Frame_Index	Cyclic index of the current C-TS Frame	22	U22	Is reset to 0 always at time 00:00 UTC. See notes in this clause.
48	Max_Inter_Schedule_Period_Length_Frames_Div2_1	Maximum Inter_Mux_Schedule_Period_Length_Frames divided by 2 minus 1	2		The maximum inter-multiplex schedule period length (in frames) over all C-TS multiplexes (of one Operator_ID), of which S-TS can be diversity combined with S-TS of the considered C-TS multiplex, is (Max_Inter_Schedule_Period_Length_Frames_Div2_1 + 1) × 2. See clause 4.11.

Parameters for the C-TS frame and the pipe multiplex					
Start bit index	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comments**
50	Config_Index	Cyclic index of the current pipe configuration on this C-TS multiplex	2	U2	This index is meant as a help for the receiver to detect that its current pipe configuration is outdated and must be updated; whenever the configuration changes, this index is incremented by 1. Hence, the receiver only has to check these bits. This index does not reflect changes in the S-TS scheduling inside the pipes, except when new S-TS are added to or removed from the C-TS mux.
52	Num_Pipes_1	Number of pipes inside this C-TS multiplex minus 1	4	U4	The number of pipes (including the tail pipe, but not including the signalling pipe) inside this C-TS mux is Num_Pipes_1 +1.
	Total length for parameters:		**56**		

Table 16: Individual parameters of each pipe (except the tail pipe)

Individual parameters of each pipe					
Start bit index	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comments**
0	Punct_Pat_ID	ID number of the Turbo code puncturing pattern	6	U6	Supports many additional code rates and additional complementary puncturing patterns.
6	Pipe_Width_CUs	Pipe width in CUs	10	U10	Use 0 if pipe is currently not used. CUs are used as the unit instead of slots in order to allow receivers to know the width of this pipe, even when Punct_Pat_ID is not known (upwards compatibility).
16	Num_STS_1	Number of S-TS inside this pipe minus 1	5	U5	The number of S-TS inside this pipe is Num_STS_1 + 1. Exception: if Pipe_Width_CUs = 0, then the number of S-TS is 0 and there is no element in the S-TS parameter list for this pipe.
21	Num_Disperser_ Sections_1	Number of disperser sections minus 1	3	U3	The number of disperser sections is Num_Disperser_Sections_1 +1. Exception: if Pipe_Width_CUs = 0, no disperser is needed and there is no element in the disperser section parameter list for this pipe.
24	Disperser_Inv_ Flag	Disperser inversion flag	1		1: Disperser profile has inverted order.
25	Regrouping_Flag	Regrouping flag	1		1: Regrouping on.
26	Inter_Schedule_ Period_ Countdown_Slots	The distance in (terms of slots) the first slot of this pipe in the current C-TS frame is away from the first slot of the next inter-multiplex S-TS schedule period	10	U10	In general, Countdown is always > 0; if a new inter-multiplex S-TS schedule period starts in the first slot of this pipe in the current C-TS frame, then not this but the next start is announced by this countdown. Special case: Countdown is 0 if the inter-multiplex S-TS schedule period length is one C-TS frame and a new S-TS inter schedule period starts in the first slot of this pipe in every C-TS frame.
36	Tap_Diff_Mult_Int_1	Integer multiplier (minus 1) for all tap lengths	4	U4	The multiplier value Tap_Diff_Mult_Int is Tap_Diff_Mult_Int_1 + 1.
40	Tap_Diff_Mult_ Fract_1_16	Fractional multiplier (minus 0,0625) for all tap lengths	4	U0.4	The multiplier value Tap_Diff_Mult_Fract is Tap_Diff_Mult_Fract_1_16 + 0,0625 (note that the number format is with 4 fractional bits).
44	Schedule_Index	Cyclic index of the current S-TS schedule configuration in this pipe	4	U4	This index is meant as a help for the receiver to detect that its current S-TS schedule configuration is outdated and must be updated; whenever the S-TS schedule configuration changes, this index is incremented by 1. Hence, the receiver only has to check these bits. This index is particularly useful in on-off channels with quickly changing schedule configurations (e.g. for Variable Bit Rate).
	Total length for parameters:		**48**		

Table 17: Individual parameters of the tail pipe

Individual parameters of *tail* pipe:					
Start bit index offset	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comment**
0	Punct_Pat_ID	ID number of the Turbo code puncturing pattern	6	U6	Supports many additional code rates and additional complementary puncturing patterns.
6	Codeword_Align_Flag	Indicator that the start of this pipe in the current C-TS frame is also the start of a codeword	1	U1	Is 1 whenever the pipe starts with a codeword start.
7	Pipe_Width_CUs	Pipe width in CUs	9	U9	Use 0, if tail pipe is currently not used. CUs are used as the unit instead of slots in order to allow receivers to know the width of this pipe, even when Punct_Pat_ID is not known (upwards compatibility).
	Total length for parameters:		**16**		

Table 18: Individual parameters for each S-TS

Individual parameters for each S-TS					
Start bit index offset	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comment**
0	STS_ID	S-TS identifier	8	U8	This ID is unique in complete network of one operator; only if two C-TS muxes carry the same S-TS (i.e. same content), their STS_ID is identical.
8	STS_Width_Slots_1	Width within one schedule period of the S-TS in slots minus 1	8	U8	The number of slots allocated to this S-TS within one schedule period of the pipe is STS_Width_Slots_1 +1.
	Total length for parameters:		**16**		

Table 19: Individual parameters for each disperser section

Individual parameters for each disperser section					
Start bit index offset	**Parameter**	**Description**	**Wordsize (bits)**	**Format**	**Comment**
0	Num_Taps_Div2_1	Number of taps of this disperser section divided by 2 minus 1	4	U4	The number of taps of this disperser section is (Num_Taps_Div2_1 + 1) × 2.
4	Tap_Diff	Tap length difference	3	U3	This value is multiplied by Tap_Diff_Mult to find the tap length difference in terms of C-TS frames.
7	Gap_Width	Gap between the first tap of this section and the last tap of the previous section	5	U5	This gap is multiplied by Tap_Diff_Mult to find the gap width in terms of C-TS frames; this gap is always zero for the first disperser section.
	Total length for parameters:		**12**		

NOTE 1: Event_Countdown: The accurate time instant, where any of the events reconfiguration/rescheduling/resynchronization takes place, is always situated between two consecutive frames, i.e. immediately after the last symbol (of the final CU) of the preceding frame and immediately before the first symbol of the SOF preamble of the succeeding frame. The Event_Countdown counts the number of C-TS frames until this event including the current frame (where the considered Signalling Pipe is received) and the last frame before the event. Hence, the last frame before the event has Event_Countdown = 1, and the first frame after has Event_Countdown = 0.

NOTE 2: Reconfig_Flag: When Reconfig_Flag == 1, *all* elements of the configuration belong to the *next* configuration, that is announced in this signalling pipe infoword, except Operator_ID and Frame_Index (which are the same for the current and the next configuration) and the S-TS parameter list. Config_Index also represents the value for the next configuration, whereas Schedule_Index depends only on the S-TS schedule (see below), but it is independent from the configuration. Observe that also Schedule_Period_Countdown_Slots is the value for the *next* configuration, and it represents the value, that should be used in the frame immediately after the switch-over from the current to the next configuration, i.e. when the new configuration is transmitted as the *current* one in the signalling pipe infoword for the first time.

NOTE 3: Reschedule _Flag: When Reschedule _Flag == 1, the S-TS parameter list inside this signalling pipe infoword belongs to the *next* S-TS schedule of the pipes. Moreover, the Schedule_Index element inside each pipe parameter set represents the index value for the S-TS schedule of that pipe *after* the rescheduling event. Observe that a rescheduling event caused by a change in the S-TS parameter list means only, that there is a new S-TS schedule in at least one of the pipes, but not necessarily in all. Hence, for some pipes, the Schedule_Index will be incremented after a rescheduling, whereas it may remain constant for others.
When new S-TS are added to or removed from the C-TS mux, this is announced as both a reconfiguration (Reconfig_Flag == 1) as well as a rescheduling (Reschedule_Flag == 1).

NOTE 4: Frame_Index: The Frame_Index is always reset to 0 for the first C-TS frame, whose first symbol of the first CU can be received anywhere in the coverage area after and including time 00:00:00.0 UTC. Note that the maximum value of the Frame_Index before the reset event is not always the same: on some days, leap seconds are introduced in UTC such that one such day lasts 24 hours plus 1 second. On these days, the Frame_Index counts higher than on regular days.

NOTE 5: Note that the time offset between the start of the first C-TS frame of a UTC day and the start of that day cannot always be kept constant, if the C-TS frame duration is not an integer divisor of one second. Therefore, this time offset necessarily changes on those days, when leap seconds are introduced. The above rules for resetting the Frame_Index ensure only that each C-TS frame with Frame_Index 0 has a time offset between 0 and the C-TS frame duration with respect to 00:00:00.0 UTC.

NOTE 6: Schedule_Period_Countdown_Slots: This countdown is used to signal in one C-TS multiplex whether the inter-multiplex schedule period is exactly one frame or not, and to signal the scheduling period starts of other C-TS multiplexes, if diversity combining is possible between S-TS of both C-TS multiplexes. It allows to predict, when the next inter-multiplex schedule alignment point will be, and thus, when any S-TS will be received latest in other C-TS multiplexes.

NOTE 7: Note again that any S-TS with its associated STS_ID are unique within a C-TS multiplex, i.e. any STS_ID may appear only once in the S-TS parameter list of a signalling pipe.

4.10.2 Partitioning of the C-TS multiplex

Every C-TS frame is partitioned with a granularity of 1 capacity unit (`CU`), i.e. 2 048 `C_Chips`. For every configurable code rate, any payload pipe is always a multiple integer number of `CUs` wide. This is ensured by the processing of each pipe presented above. The concatenation of `SOF` preamble with the signalling pipe is always 2 `CUs` wide.

The number of pipes inside the C-TS multiplex is variable, and it is configured by the value `Num_Pipes`, which can be calculated from the parameter `Num_Pipes_1` inside the signalling pipe by:

```
Num_Pipes = Num_Pipes_1 + 1
```

Inside the signalling pipe, there is a list of Num_Pipes elements directly after the parameters for the C-TS frame and pipe multiplex. Inside this list, referred to as the pipe parameter list, element i with i from 1 to Num_Pipes states the parameters for payload pipe i. Observe that the parameter set for the pipe with index Num_Pipes, the so-called tail pipe, differs from the parameter set used for the other payload pipes.

The width of pipe i in terms of CUs is transmitted in the parameter Pipe_Width_CUs inside list element i. From this parameter, the width Pipe_Width_Slots of pipe i in terms of slots depends on the pipe's coderate and can be calculated as follows:

```
   Pipe_Width_Slots = Pipe_Width_CUs / CU_Per_CW
```

The total number of CUs inside one C-TS frame depends on the IPL used.

4.10.3 S-TS schedule and slot allocation

For defining the scheduling of S-TS inside a pipe, an ordered list of S-TS is read from the signalling pipe and evaluated as follows.

For pipe i with i from 1 to Num_Pipes - 1, the number Num_STS[i] of S-TS transported inside this pipe is calculated from the parameter Num_STS_1 inside the i-th element of the pipe parameter list by:

```
   Num_STS[i] = Num_STS_1 + 1
```

For the tail pipe, we have always

```
   Num_STS[Num_Pipes] = 1.
```

However, if the parameter Pipe_Width_CUs of the considered pipe is 0, then Num_STS[i] is set to 0 irrespective of the value Num_STS_1 transmitted in the signalling pipe. This is also true for the tail pipe.

Inside the signalling pipe, there is a list of Total_Num_STS elements, with:

```
   Total_Num_STS = Num_STS[1] + Num_STS[2] + … + Num_STS[Num_Pipes]
```

This list is referred to as the S-TS parameter list, and it is stored inside the signalling pipe directly after the pipe parameter list. Inside the S-TS parameter list, element j with j from 0 to Total_Num_STS - 1 states the parameters for S-TS j.

The width STS_Width_Slots[j] of S-TS j in terms of slots is calculated from the parameter STS_Width_Slots_1 inside the j-th element of the S-TS parameter list by:

```
   STS_Width_Slots [j] = STS_Width_Slots _1 + 1
```

The mapping of S-TS to pipes, which they are transported over, are done in the following way: The first Num_STS[1] S-TS are transported over pipe 1, the next Num_STS[2] S-TS are transported over pipe 2 etc.

There are Num_STS[i] S-TS's inside pipe i. Pipe i transports the S-TS M, M+1, …, M + Num_STS[i] - 1 with M representing Num_STS[1] + … Num_STS[i-1].

Without regrouping, the time scheduling of these S-TS is to transport first STS_Width_Slots[M] slots of S-TS M, then STS_Width_Slots[M+1] slots of S-TS M+1 until finally STS_Width_Slots[M + Num_STS[i] - 1] slots of S-TS M + Num_STS[i] - 1. This sequence of S-TS - or their constituent slots, respectively, is called one intra(-multiplex) schedule period.

The prefix intra distinguishes this schedule period from an inter(-multiplex) schedule period as presented in clause 4.11. In the sequel, the prefix intra(-multiplex) is left out for the sake of readability, and the term schedule period is equivalent to intra schedule period, unless the term inter schedule period is used explicitly.

The length of the schedule period of pipe i in terms of slots inside pipe i is:

```
Schedule_Period_Length_Slots[i] = STS_Width_Slots[M] + STS_Width_Slots[M+1] + …

      + STS_Width_Slots[M + Num_STS[i] - 1]
```

The schedule period is partitioned into C-TS frames by the width `Pipe_Width_Slots` of the pipe in terms of slots. The pipe's schedule period is periodically repeated over time.

For a specific C-TS frame, the above rule therefore defines the allocation of the available `Pipe_Width_Slots` slots of the considered pipe to the S-TS of that pipe. The slot allocation may vary from C-TS frame to C-TS frame, but it is periodic with the schedule period (in terms of C-TS frames):

```
Schedule_Period_Length_Frames[i] = Schedule_Period_Length_Slots[i] /
     Pipe_Width_Slots[i]
```

Synchronization to the current schedule period is possible by the value `Inter_Schedule_Period_Countdown_Slots` transmitted inside the signalling pipe. This value announces the distance (in terms of slots of the considered pipe) of the first slot of the considered pipe in the current C-TS frame to the first slot of the next inter schedule period. If both are identical, i.e. when the next inter schedule period starts in the first slot of the current C-TS frame, we have either `Inter_Schedule_Period_Countdown_Slots = 0`, if the inter schedule period length is 1 frame, otherwise `Inter_Schedule_Period_Countdown_Slots` takes the length in slots of the inter schedule period . From this value together with the known S-TS schedule and the corresponding Schedule_Period_Length_Slots value, the position inside the current schedule period can be derived.

With regrouping, the slot allocation of the S-TS is the same as described above, but after the collector all slots belonging to the same pipe and C-TS frame are re-arranged as described in clause 4.8.

4.10.4 S-TS re-scheduling and slot re-allocation

When the schedule of existing S-TS inside one pipe is changed, i.e. when only the widths of the S-TS in the pipe are changed and no S-TS are added to or removed from the pipe and no other parameters like pipe width, code rate and disperser profile are changed, then this re-scheduling is not considered as a reconfiguration event, and it does not have any impact on other pipes. The re-scheduling event shall become effective at the start of a schedule period.

The S-TS schedule transmitted in the signalling pipe is always valid for the current schedule period as it is seen at the output of the un-disperser in the receiver. That is the signalling pipe always reflects the S-TS schedule for those codewords, whose IUs have been delayed correctly in the receiver and are output within the current schedule period. If the previous schedule period ends within a C-TS frame, and the next schedule period starts in the same frame, the S-TS schedule announced in the signalling pipe of that frame is valid for the next schedule period. At the first C-TS frame of the schedule period, where the new S-TS schedule is used in the considered pipe, the parameter `Schedule_Index` is incremented for this pipe.

4.10.5 Birth/death of S-TS

When a new S-TS is added to or an existing S-TS is removed from a pipe, this event is signalled in the signalling pipe just like a pipe reconfiguration, that is, a pipe reconfiguration is announced using the parameters `Reconfig_Flag` and `Event_Countdown`, although the announcement can be on much shorter notice than for a pipe reconfiguration. Moreover, the parameters `Schedule_Index` of the affected pipes and `Config_Index` are incremented.

Hence, the configuration of every pipe, including the parameter `Num_STS_1` that links the S-TS in the S-TS parameter list to their associated pipes, has to be reread by the terminal, even if it is receiving and decoding a different pipe that is not affected by the addition or removal of an S-TS. If the pipe width, coderate, and disperser profile of the pipes do not change, the addition or removal of an S-TS is treated like an S-TS rescheduling, i.e. the pipe's slots are re-allocated for the C-TS frames inside a scheduling period. This re-scheduling event shall become effective at the start of a schedule period.

4.10.6 S-TS ID

The S-TS identifier (ID) `STS_ID` of S-TS `j` is stored inside the `j`-th element of the S-TS parameter list.

Over the total ensemble of C-TS multiplexes from one network operator (i.e. one unique `Operator_ID`, that is transmitted inside the signalling pipe) all S-TS that carry the same S-TS ID also carry the same content. This is for example useful, if the same S-TS is transmitted over satellite and terrestrial networks. The network operator shall synchronize these S-TS in such a way that they can be diversity combined in the receivers, see clause 4.11.

Only the S-TS with STS_ID = 0 plays a different role: this S-TS is C-TS multiplex specific and may not be diversity combined with other S-TS having STS_ID = 0 (on other C-TS multiplexes), since it carries the configuration of the Service Layer for this C-TS multiplex.

4.10.7 Calculation of the disperser profile

The disperser profile of each pipe is calculated based on the parameters transmitted inside the signalling pipe:

From the parameter Num_Disperser_Sections_1, the number of disperser sections Num_Disperser_Sections is calculated by:

```
Num_Disperser_Sections = Num_Disperser_Sections_1 + 1.
```

However, if the parameter Pipe_Width_CUs of the considered pipe is 0, then Num_Disperser_Sections is set to 0 irrespective of the value Num_Disperser_Sections_1 transmitted in the signalling pipe.

The integer tap length difference multiplier Tap_Diff_Mult_Int is calculated by:

```
   Tap_Diff_Mult_Int = Tap_Diff_Mult_Int_1 + 1,
```

where the parameter Tap_Diff_Mult_Int_1 is transmitted in the signalling pipe.

The fractional tap length difference multiplier Tap_Diff_Mult_Fract is calculated by:

```
   Tap_Diff_Mult_Fract = Tap_Diff_Mult_Fract_1_16 + 0,0625
```

Observe that the parameter Tap_Diff_Mult_Fract_1_16 is transmitted inside in the signalling pipe in format U0.4 (i.e. with 4 fractional bits).

Inside the signalling pipe, there is a list of Total_Num_Disperser_Sections elements, with:

```
Total_Num_Disperser_Sections = Num_ Disperser_Sections[1] +
    Num_ Disperser_Sections[2] + … + Num_ Disperser_Sections[Num_Pipes]
```

This list is referred to as the disperser section parameter list, and it is stored inside the signalling pipe directly after the S-TS parameter list.

The disperser profile of pipe k comprises the disperser sections associated with elements M, M+1, …, M + Num_Disperser_Sections [k] - 1 of the disperser section parameter list with M representing Num_Disperser_Sections [1] + … Num_Disperser_Sections [k-1].

Note that the parameters inside the signalling pipe characterize the disperser profile to be used inside the *receiver*. It can be calculated from these parameters in the following steps:

For each disperser section i from 0 to Num_Disperser_Sections - 1 of the considered pipe (do not confuse i, which is the number of the disperser section of the considered pipe, with the index of the corresponding element inside the disperser section parameter list), the following values are calculated from the respective parameter inside the signalling pipe:

The number of taps Num_Taps[i] inside this disperser section i are calculated by:

```
   Num_Taps[i] = (Num_Taps_Div2_1 + 1) × 2
```

For each tap j from 0 to Num_Taps[i] - 1 of disperser section i of the considered pipe, an intermediate length is calculated by:

```
Intermed_Len[i][j] = Intermed_Len[i-1][Num_Taps[i-1]-1] + ...
Gap_Width[i] + j × Tap_Diff[i]
```

For the first disperser section i = 0, the first two terms are dropped, i.e.

```
   Intermed_Len[0][j] = j × Tap_Diff[0]
```

The total number of taps over all disperser sections is always 32. The tap lengths Tap_Len[i][j] in terms of C-TS frames are calculated from the intermediate lengths by:

```
   Tap_Length[i][j] = Tap_Diff_Mult_Int × floor(Intermed_Len[i][j] ×
...Tap_Diff_Mult_Fract),
```

where floor(x) represents rounding to the largest integer <= x. The maximum delay of all disperser taps is:

```
   Max_Delay = Tap_Length[Num_Disperser_Sections-1][Num_Taps[Num_Disperser_Sections-1]-1].
```

When the tap lengths Tap_Length[i][j] have been calculated for all disperser sections, they are handled as a single vector of 32 elements in the order taps 0 to Num_Taps[0] -1 of disperser section 0, then taps 0 to Num_Taps[1] - 1 of disperser section 1, etc.

These 32 elements are now re-ordered as follows: They are sequentially row-wise written into a 4 row by 8 column-matrix and column-wise read out.

For the case of an inverted disperser profile, i.e. if Disperser_Inv_Flag == 1, the re-ordered 32 element-vector is flipped, i.e. the element of index 0 is swapped with that of index 31, the element of index 1 is swapped with that of index 30, etc.

The vector of disperser tap delays Tap_Delay_Rec[k] with k from 0 to IU_Per_CW - 1, which is actually used in the disperser of the *receiver*, is gained from this re-ordered (and possibly flipped) 32 element vector by periodically repeating it, i.e. Tap_Delay_Rec[k] = Tap_Delay_Rec[k+32] for k from 0 to IU_Per_CW-33. Observe that IU_Per_CW need not be an integer multiple of 32, such that the last period of this repetition pattern remains possibly incomplete. If IU_Per_CW ≤ 32, there is no periodicity at all.

The vector of disperser tap delays Tap_Delay[k] with k from 0 to IU_Per_CW - 1, which is used in the disperser of the *transmitter*, is calculated by

```
   Tap_Delay[k] = Max_Delay - Tap_Delay_Rec[k].
```

Observe that the minimum of all Tap_Delays[k] may be larger than zero for IU_Per_CW < 32 because of the above equation.

4.10.8 Configuration of the tail pipe

The last pipe (tail pipe) is configured by a special parameter set inside the signalling pipe. It carries always only one S-TS (or none if the tail pipe is not used, i.e. if Pipe_Width_CUs is set to 0), it has no dispersing, cannot be time-sliced and has not the restriction that the pipe width is an integer multiple of the slot/codeword width. This may lead to rotating codeword starts inside the pipe, which in turn needs a flag Codeword_Align_Flag to synchronize to this rotation among the parameters of this pipe. The parameters of the single S-TS inside the tail pipe are taken from the S-TS parameter list as specified in clause 4.10.3.

If the tail pipe is not used, Pipe_Width_CUs shall be set to 0 for this pipe.

4.10.9 Unused pipes

If a previously used pipe (i.e. Pipe_Width_CUs > 0) is reconfigured in order not to be used any more, its Pipe_Width_CUs shall be set to 0. All S-TS from this pipe's S-TS schedule inside the S-TS parameter list of the signalling pipe shall be deleted from the list.

4.10.10 Announcing reconfigurations and reschedulings

Announcing a reconfiguration (with Reconfig_Flag = 1) and/or a rescheduling (with Reschedule_Flag == 1) is not mandatory, when switching over from an old to a new configuration and/or S-TS schedule. These announcements are provided rather to make the transmission more robust to signal dropouts and to be able to prepare the receiver for the approaching changes in the received signal.

NOTE: A switch-over without such an announcement is valid, yet not recommended. Without announcement, the receiver could exhibit an unpredictable behaviour, when the old configuration and/or S-TS schedule is erroneously applied instead of the correct new configuration/schedule, which can result in a loss of the received services even after the reconfiguration/rescheduling event has been detected.

The duration of the announcement and the rhythm used for alternating between the current and the next configuration/S-TS parameter list is up to the network operator and depends strongly on the parameters used for the transmitted signal and on the addressed receiver terminals.

4.10.11 Pipe reconfiguration

A pipe reconfiguration is a more complex process than an S-TS re-scheduling event or the birth/death of S-TS. A pipe reconfiguration is characterized by the change of any one of the following parameters in any pipe:

- Pipe width (i.e. parameter `Pipe_Width_CUs`).
- Code rate (i.e. parameter `Punct_Pat_ID`).
- Disperser profile (any of the parameters associated with the disperser, see clause 4.7).
- Inter-multiplex schedule period length changes from 1 C-TS frame to a different value or vice versa (can be detected from parameter Inter_Schedule_Period_Countdown_Slots).
- For the ensemble of pipes: Change of Max. schedule period length (i.e. parameter Max_Inter_Schedule_Period_Length_Frames_Div2_1).

A pipe reconfiguration is signalled in advance using the parameters `Reconfig_Flag` and `Event_Countdown` inside the signalling pipe.

`Event_Countdown` counts down the number of C-TS frames until the reconfiguration takes place. The C-TS frame, at which this counter reaches the value `zero`, is the first one using the new configuration. If no reconfiguration is approaching, this counter remains `zero`.

`Reconfig_Flag = 1` signals that the parameters transmitted inside the signalling pipe of the current C-TS frame are those for the next configuration and not those for the current one. By this mechanism, the receiver is able to know the next configuration in advance and can prepare for the reconfiguration event.

Whenever the pipe configuration changes, the cyclic counter `Config_Index` (parameter inside the signalling pipe) is incremented at the first C-TS frame using the new configuration.

Table 20 gives an overview over the three possible reconfiguration events.

Table 20: Possible reconfiguration events

	S-TS re-scheduling	Birth/death of an S-TS	Pipe reconfiguration
Characterization	The slot allocation of an S-TS within a scheduling period changes.	At least one S-TS is added to or removed from at least one pipe.	The QoS (throughput, error-protection, time-interleaving/end-to-end-delay) of a pipe changes.
Affects	S-TS scheduling of a pipe.	S-TS scheduling of a pipe.	Configuration of a pipe.
Changes inside signalling pipe	Only values of parameter `STS_Width_Slots_1` change; structure of signalling pipe infoword is unchanged.	Parameter value `Num_STS_1` changes and possibly length of S-TS parameter list changes (in this case the structure of the signalling pipe infoword changes).	Parameters `Pipe_Width_CU`, `Punct_Pat_ID` and/or the disperser-related or the inter-multiplex schedule period-related parameters change; possibly the length of the disperser section parameter list changes (in this case the structure of the signalling pipe infoword changes).

	S-TS re-scheduling	Birth/death of an S-TS	Pipe reconfiguration
Possible Signalization	Advance signalization by the parameters `Reschedule_Flag` and `Event_Countdown`; event increments the parameter `Schedule_Index` in the concerned pipe.	Advance Signalization as for pipe reconfiguration; event increments the parameters `Schedule_Index` in the affected pipes (where something changes) and `Config_Index.`	Advance Signalization by the parameters `Reconfig_Flag` and `Event_Countdown`; event increments the parameter `Config_Index.`

Disperser profile of pipe changes: The disperser profile is switched abruptly at the reconfiguration instant. The transmitter must in all cases keep the position of the latest IU, which remains non-delayed in the receiver, in the same grid after the reconfiguration event as it was before. The positions of all other IUs are changed abruptly at the reconfiguration event according to the new disperser profile. Following these rules, some transmitted codewords may have a lower number of transmitted IUs than IU_Per_CW (since some IUs are jumped over in positive time by the reconfiguration event and are not transmitted), whereas others may have a higher number (they are jumped over in negative time and are transmitted twice). This behaviour is illustrated in three examples in figures 10, 11 and 12. The first two figures display codewords with 2 IUs, the third one a codeword with 3 IUs. IUs belonging to the old disperser profile are numbered e.g. 1a, whereas those of the new interleaver profile are numbered e.g. 1b. Hatched IUs are not transmitted.

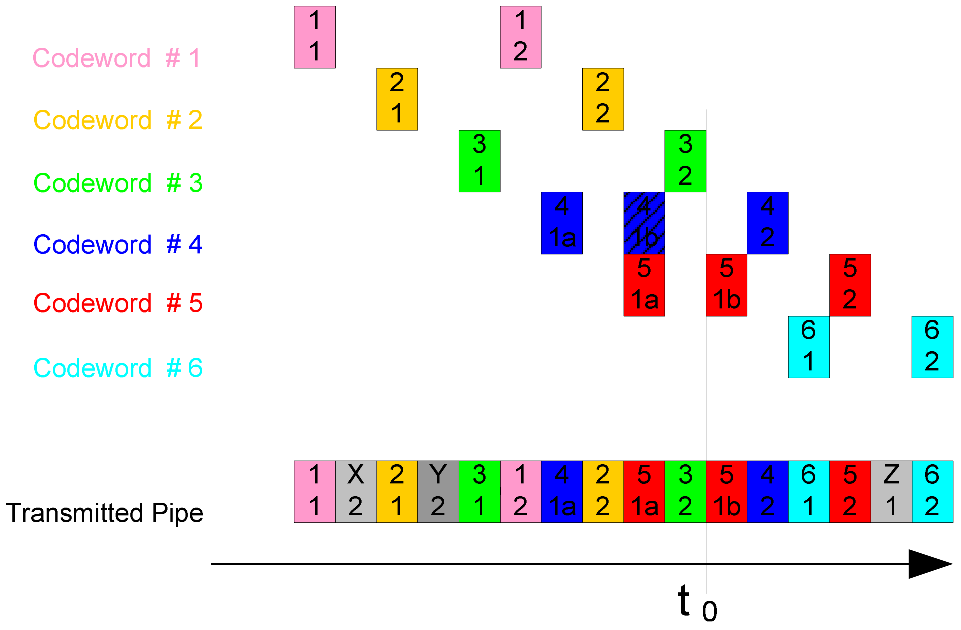

Codeword # 1
Codeword # 2
Codeword # 3
Codeword # 4
Codeword # 5
Codeword # 6
Transmitted Pipe
1 1 | X 2 | 2 1 | Y 2 | 3 1 | 1 2 | 4 1a | 2 2 | 5 1a | 3 2 | 5 1b | 4 2 | 6 1 | 5 2 | Z 1 | 6 2
t 0

Figure 10: Shorter disperser delay after reconfiguration

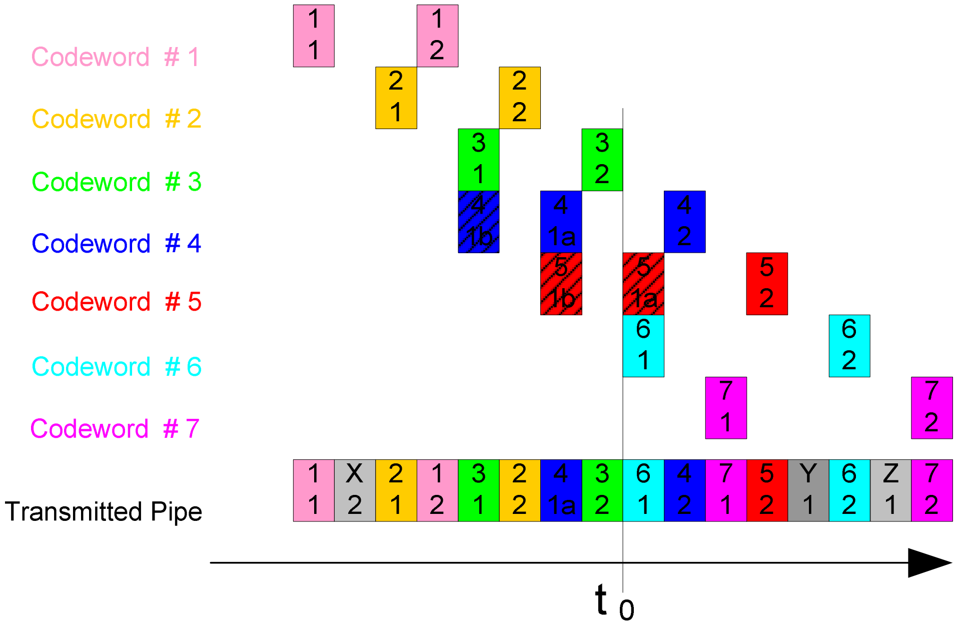


Figure 11: Longer disperser delay after reconfiguration

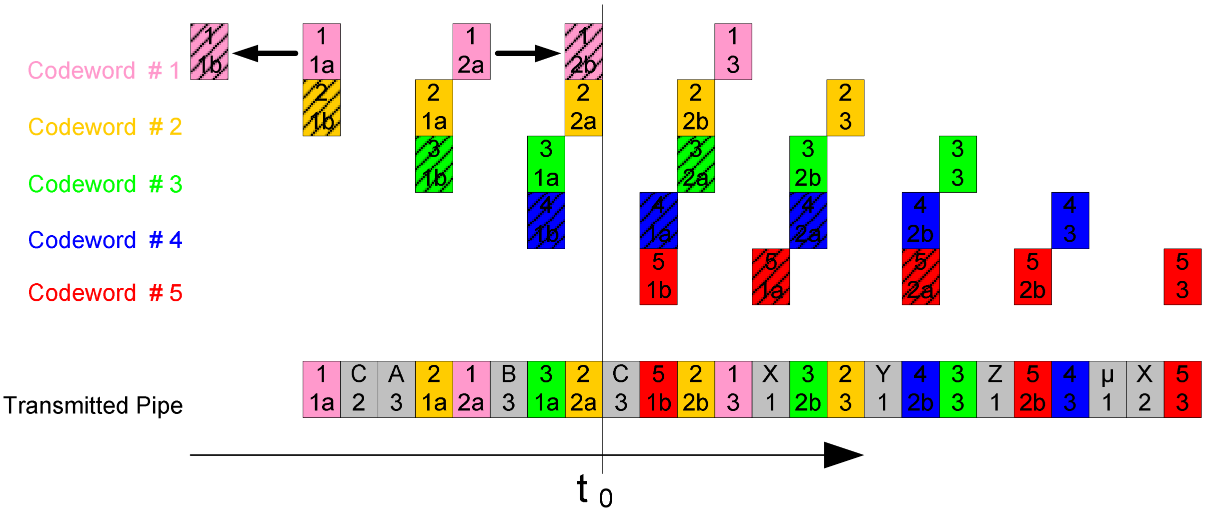
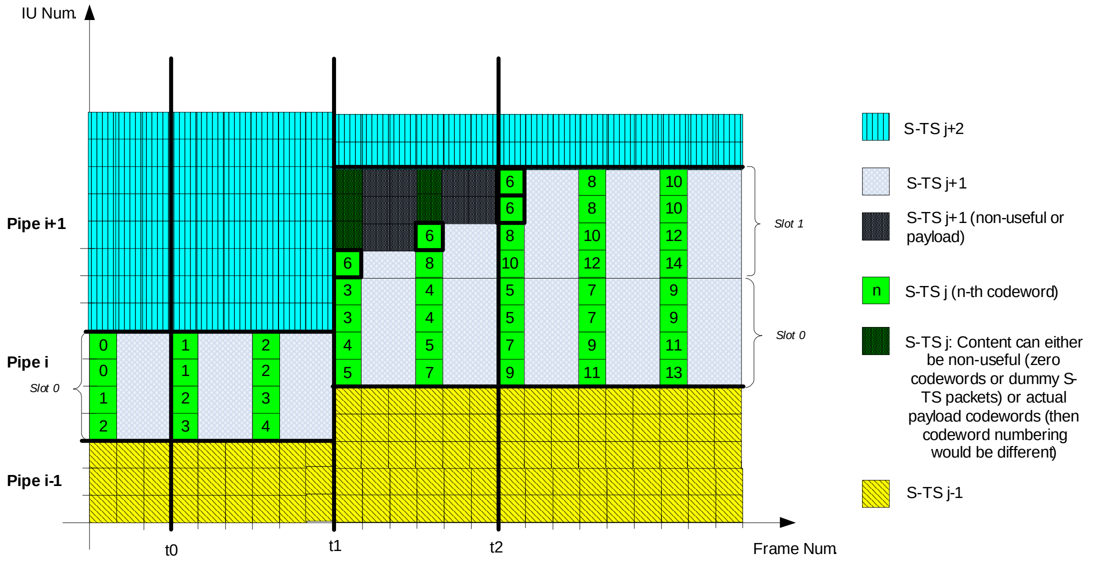

Codeword # 1
Codeword # 2
Codeword # 3
Codeword # 4
Codeword # 5
Transmitted Pipe
1 1a, C 2, A 3, 2 1a, 1 2a, B 3, 3 1a, 2 2a, C 3, 5 1b, 2 2b, 1 3, X 1, 3 2b, 2 3, Y 1, 4 2b, 3 3, Z 1, 5 2b, 4 3, μ 1, X 2, 5 3
t 0

Figure 12: Other example of disperser reconfiguration

- Code rate of pipe changes:
  - Lower code rate after reconfiguration: `IU_Per_CW` increases, i.e. dispersers are extended, new taps are added after the existing taps; new taps are initialized with content zero; new coderate and disperser width become effective at reconfiguration instant (i.e. receiver can use lower coderate only after the end-to-end delays of the interleaver).
  - Higher code rate after reconfiguration: last taps are killed, new coderate and disperser width become effective before reconfiguration instant (that instant minus the interleaver's end-to-end delay); flushing of the dying taps is continued until the reconfiguration instant by feeding them with zero input, this is done in order to transmit the last valid `IUs` still contained in these taps; after the reconfiguration instant, the superfluous taps are dead and discarded.

- Total throughput of a pipe changes (variation of Pipe_Width_Slots, e.g. changing pipe width - parameter Pipe_Width_CUs - in case of constant code rate); this is a simple re-allocation/re-scheduling of the slots to all S-TS of the pipe:
  - Pipe width extension - total throughput of the pipe becomes higher (`Pipe_Width_Slots` increases): new slots are added after the existing slots of the pipe and new dispersers are initialized with content zero. The new S-TS schedule becomes effective at the reconfiguration instant (t1 in figure 13 in the receiver, i.e. the new slot allocation in the transmitter can become effective at an instant t0, which is `Max_Delay` frames (the disperser's end-to-end delay) before the reconfiguration instant t1, when the new slots become active. Thus, the receiver outputs higher throughput immediately after the reconfiguration (instant t1). However, the received codewords used for decoding are not complete, since in the frame after the reconfiguration instant t1, only the latest IUs of the codewords in the new slots have been received, in the next frame only the IUs with delay 0 or 1 frames etc. In order to improve the robustness of the transported payload for such a pipe width extension, an alternative for the transmitter side is to input only non-useful data into the new slots between instants t0 and t1. This data could be either all-zero codewords or dummy S-TS packets (see clause 4.3.1), both of which will be discarded in the receiver. For this solution, the receiver outputs higher throughput only after the end-to-end delay of the disperser (instant t2).


IU Num.
Pipe i+1
Pipe i
Slot 0
Pipe i-1
Slot 1
Slot 0
t0
t1
t2
Frame Num.
S-TS j+2
S-TS j+1
S-TS j+1 (non-useful or payload)
n S-TS j (n-th codeword)
S-TS j: Content can either be non-useful (zero codewords or dummy S-TS packets) or actual payload codewords (then codeword numbering would be different)
S-TS j-1

t0: transmitter starts to input codewords (either non-useful - i.e. zero codewords or dummy S-TS packets – or payload) to the disperser of new Slot 1

t1: pipe width changes and S-TS schedule changes (two slots for S-TS j) and both slots are filled with payload; partially complete codewords can be received in Slot 1

t2: after the disperser's end-to-end delay after t1, complete codewords can be received in Slot 1

Figure 13: Pipe width extension

  - Pipe width reduction - total throughput of pipe becomes lower (`Pipe_Width_Slots` decreases): last slots are killed, new slot allocation in the transmitter becomes effective at an instant t0 (see figure 14), which is the disperser's end-to-end delay Max_Delay frames before reconfiguration instant t1. The S-TS schedule for the receiver changes at the same time as the pipe width, i.e. at the reconfiguration instant t1; flushing of the dying slots is continued in the transmitter until the reconfiguration instant by feeding them with zero input, this is done in order to transmit the last valid IUs still contained in these slots; after the reconfiguration instant, the superfluous slots are dead and discarded.

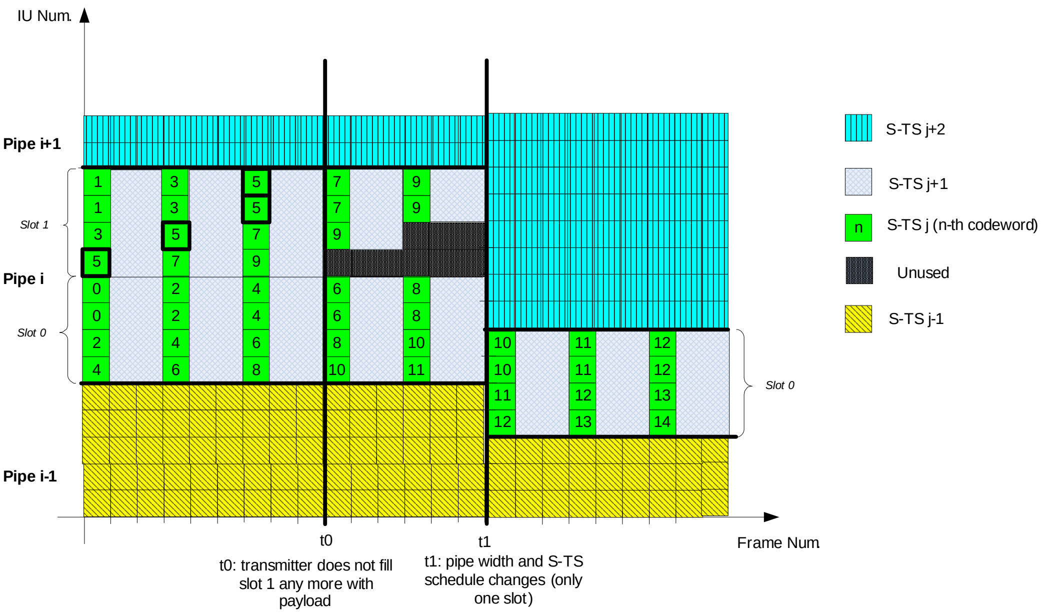


Figure 14: Pipe width reduction

4.11 Network aspects

The operator of a transmitter network (which may comprise multiple terrestrial and/or satellite C-TS multiplexes) must ensure that the C-TS frames with the same Frame_Index from all transmitters supplying a considered coverage region and for all C-TS multiplexes arrive at every point inside the coverage region with a delay not exceeding plus minus 10 ms for the case that the frame duration is 432 ms.

Any S-TS with a given STS_ID must carry exactly the same payload content (i.e. PF infowords) in every C-TS multiplex that contains it. This requirement allows the combining of an S-TS received over different C-TS multiplexes in order to maximize the reception diversity (diversity combining, see clause 4.4.7). Pipes carrying S-TS, that are intended for diversity combining, must obey the following restrictions with regard to their S-TS scheduling.

Consider an S-TS X, that is transmitted on pipe i on one C-TS multiplex 1 and on pipe j on another C-TS multiplex 2. Besides S-TS X, pipe i of C-TS multiplex 1 can carry further S-TS, i.e. S-TS X is part of an intra-multiplex schedule period. After one or several frames, a new intra-multiplex schedule period starts on pipe i of C-TS multiplex 1. The first slot of such an intra-multiplex schedule period can, but need not, be the first slot of pipe i in a frame - in any case, every one or several intra-multiplex schedule periods, this slot can be identified using the field Inter_Schedule_Period_Countdown_Slots inside the signalling pipe part of pipe i. The same is true for pipe j on C-TS multiplex 2, even though completely different S-TS may accompany S-TS X in the intra-multiplex schedule period, that are not intended for diversity combining with C-TS multiplex 1.

Whenever pipe i on C-TS multiplex 1 carries the same S-TS X as pipe j on C-TS multiplex 2, the diversity combining mechanism (for S-TS X) requires that the first slots of the intra-multiplex schedule periods of the two considered pipes/C-TS multiplexes must be in the same frame (same Frame_Index) every couple of frames, where the maximum is Max_Inter_Schedule_Period_Length_Frames frames (calculated from the parameter Max_Inter_Schedule_Period_Length_Frames_Div2_1 transmitted in the signalling pipe of each of the considered C-TS multiplexes). Whenever this is the case, the starts of the intra-multiplex schedule periods on the two considered pipes are said to be aligned, and the starts of the first slots in the two intra-multiplex schedule periods are referred to as the alignment points of the considered pipes. It is these alignment points, that are announced in the field Inter_Schedule_Period_Countdown_Slots inside the signalling pipe part of pipe i . The sequence of slots (and of their associated S-TS) in a pipe between two consecutive alignment points is referred to as an inter-multiplex schedule period. One inter(-multiplex) schedule period corresponds to N intra(-multiplex) schedule periods of pipe i in C-TS multiplex 1 and M intra(-multiplex) schedule periods of pipe j in C-TS multiplex 2, where N and M are positive integers figure 15 illustrates these relationships.

The inter schedule period length is the least common multiple of the intra schedule period lengths of the corresponding pipes in all concerned C-TS multiplexes. The maximum inter schedule period length (in frames) is Max_Inter_Schedule_Period_Length_Frames (for the number of slots: see clause 4.10.3).

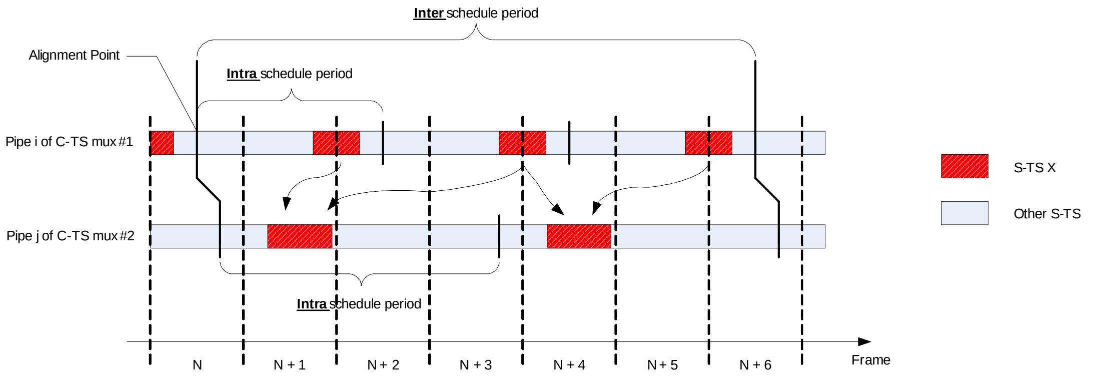


Figure 15: Intra- and Inter-multiplex schedule periods and their alignment points

While the alignment points of all pipes carrying S-TS X in different C-TS multiplexes must be in the same frame, there are no restrictions about the position of the alignment points inside these frames.

The sequence of slots inside an intra- or inter-multiplex schedule period of a pipe corresponds to the associated sequence of codewords in the following way: the p-th codeword is the one, whose last IU is transmitted in the p-th slot.

The position of a codeword inside an S-TS with respect to the most recent alignment point identifies it for diversity combining this S-TS from all C-TS multiplexes carrying this S-TS: all codewords having position p (inside the considered S-TS) in any of the C-TS multiplexes carries the same PF infoword and thus may be combined.

In order to allow this alignment of codewords representing the same PF infoword in different C-TS multiplexes, the following has to be done, if the parameter Max_Delay is not the same in these C-TS multiplexes (see clause 4.10.7): it is necessary to introduce additional disperser delay at the transmitter side in one or more C-TS multiplexes for the pipes carrying the considered S-TS, such that the end-to-end delay becomes the same in all pipes and C-TS multiplexes carrying the considered S-TS.

The intra (S-TS) schedule of a pipe may change only at an alignment point.

Annex A (normative): Number format definitions

A.1 Number format and transmission order

Unless otherwise stated, all bit/symbol streams and values are transmitted with the following convention:

- In a stream, bits/symbols with a lower index are transmitted temporally earlier than those with a higher index.
- A prefix of a block of bits/symbols is transmitted temporally first, whereas a suffix is transmitted temporally last.
- Signed integer and signed fixed-point values are stored in two's complement format.
- If a value is represented by N bits, the Most Significant Bit (MSB), i.e. bit N-1, is transmitted temporally first followed by bits N-2 down to bit 0, the Least Significant Bit (LSB). This order is referred to as Big Endian.
- For Bytes, the MSB, bit 7, is transmitted temporally first and the LSB, bit 0, last.
- Symbols of a BCH, Reed-Solomon or CRC-code are transmitted temporally in the following order: the symbol with highest degree in polynomial representation comes first and the symbol with degree 0 comes last.
- The format of integer and fix-point values are specified in the following way: the first letter is U for unsigned and S for signed values, the following value following that letter states the number of integer bits. In the case of fixed-point values, this value is followed by a dot"." and another value, which specifies the number of fractional bits. Examples: U8, S3.2.

A.2 SI-Prefix Notation

The present document uses the prefix notation as defined by the "Système International d'Unités", i.e. M (mega) represents 1 000 000 units, k (kilo) represents 1 000 units and m (milli) represents 0,001 units.

A.3 Default Settings

If not stated otherwise, the following default settings are used:

- RFU bits have value 0.

Annex B (normative): Calculation of the CRC word

The implementation of Cyclic Redundancy Check codes (CRC-codes) allows the detection of transmission errors at the receiver side. For this purpose CRC words shall be included in the transmitted data. These CRC words shall be defined by the result of the procedure described in this annex.

A CRC code is defined by a polynomial of degree *n*:

$$G_n(x) = x^n + g_{n-1}x^{n-1} + \ldots + g_2x^2 + g_1x + 1$$

with $n \geq 1$:

and: $$g_i \in \{0,1\}, \quad i = 1.....n-1$$

The CRC calculation may be performed by means of a shift register containing *n* register stages, equivalent to the degree of the polynomial (see figure B.1). The stages are denoted by b_0 ... b_{n-1}, where b_0 corresponds to 1, b_1 to *x*, b_2 to x^2, ..., b_{n-1} to x^{n-1}. The shift register is tapped by inserting XORs at the input of those stages, where the corresponding coefficients g_i of the polynomial are "1".

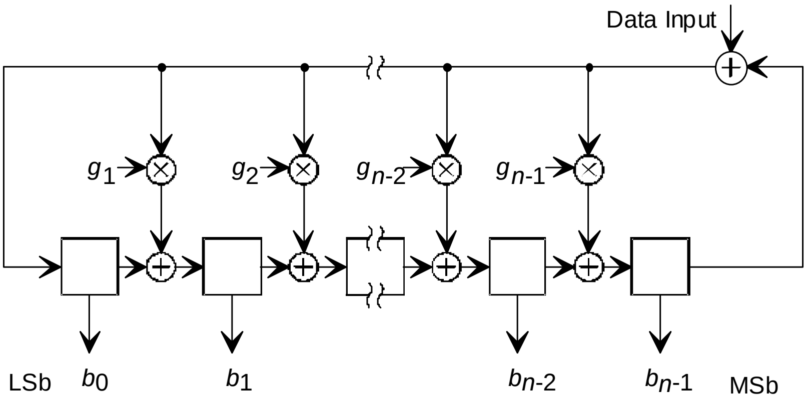


Figure B.1: General CRC block diagram

At the beginning of the CRC calculation, all register stage contents are initialized to all ones. After applying the first bit of the data block (MSB first) to the input, the shift clock causes the register to shift its content by one stage towards the MSB stage (b_{n-1}), while loading the tapped stages with the result of the appropriate XOR operations. The procedure is then repeated for each data bit. Following the shift after applying the last bit (LSB) of the data block to the input, the shift register contains the CRC word which is then read out. Data and CRC word are transmitted with MSB temporally first. The CRC shall be inverted (1's complement) prior to transmission.

Annex C (informative): Bibliography

3GPP2 C.S0002-D: "Physical Layer Standard for cdma2000 Spread Spectrum Systems, Release D".
http://www.3gpp2.org/Public_html/specs/C.S0002-D_v2.0_051006.pdf

History

Document history		
V1.1.1	November 2006	Publication as TS 102 550
V1.2.1	January 2007	Publication as TS 102 550
V1.3.1	January 2008	Publication as TS 102 550
V1.1.0	July 2009	Public Enquiry PE 20091122: 2009-07-25 to 2009-11-23