ETSI TS 102 550 V1.1.1 (2006-11)

Technical Specification

Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Outer Physical Layer of the Radio Interface

Reference

DTS/SES-00284

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

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2006.
All rights reserved.

DECT™, **PLUGTESTS**™ and **UMTS**™ are Trade Marks of ETSI registered for the benefit of its Members.
TIPHON™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

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 Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

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.

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 transmission, and the inner physical layer with multiple carriers transmission. 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 transmission is specified in TS 102 551-1, and with multiple carriers transmission in TS 102 551-2.

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

Void.

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
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
PF	Physical layer FEC
OPL	Outer Physical Layer
PFIW	Physical Layer FEC Info Word
PL	Payload
QoS	Quality of Service
RFU	Reserved for Future Use
SOF	Start Of Frame
S-TS	Service-TransportStream
TS	ETSI Technical Specification
VBR	Variable Bit Rate
WER	Word Error Rate

3.3 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.
- 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.

3.4 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.

4 Outer physical layer

4.1 Overview

The functionality of the Outer Physical Layer (in the following denoted OPL) 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 1.

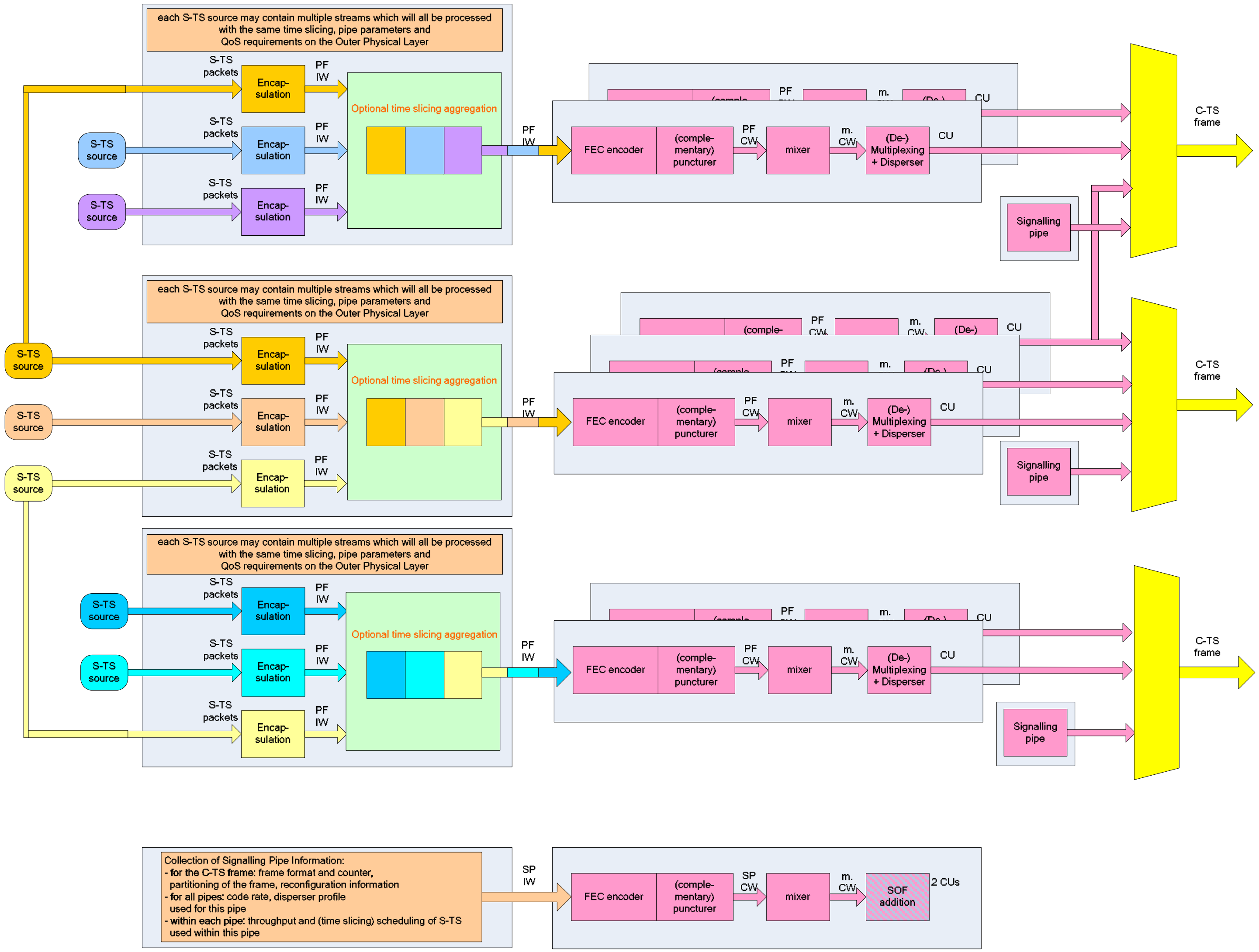

Figure 1: General overview of the OPL functionality

The processing, multiplexing and demultiplexing of the data in the OPL is displayed in Figure 2.

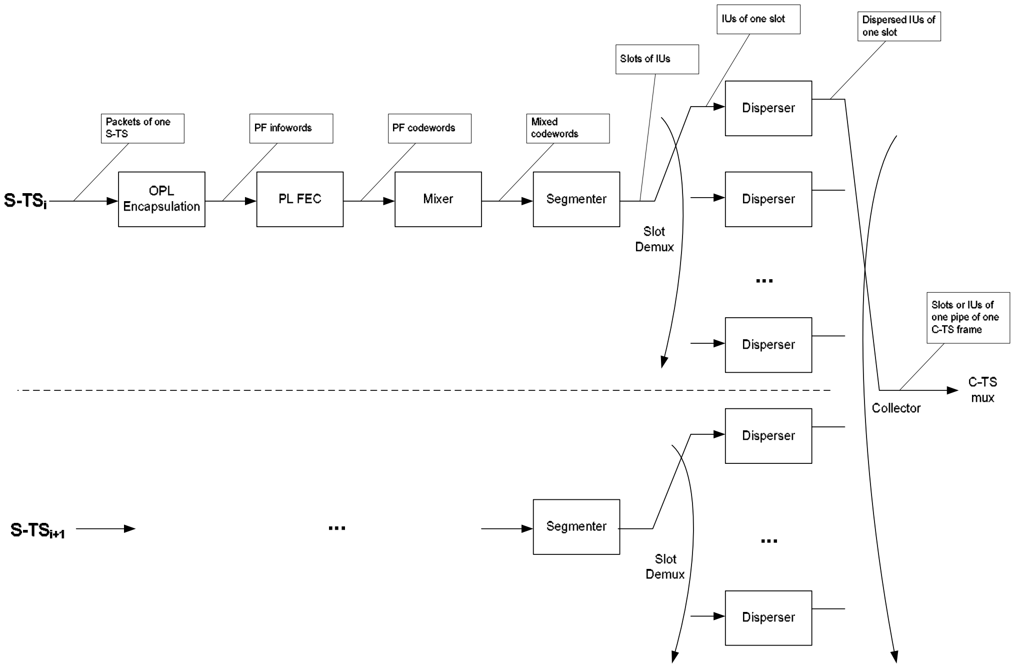

Figure 2: Definition of the different blocks envolved in the OPL processing

4.2 Interfacing to SL (Service Layer)

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 approximatively 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 is identical.
  - An S-TS may not be fed to several pipes inside the same C-TS multiplex.
  - 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.

The detailed format for the different types of S-TS is given in the following clauses.

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 this standard.

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, 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.

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 (1504 bytes)	
12 032	RFU_Error_Correction	Reserved for Error Detection or Outer Error Correction Code	196	4×U49	e.g. 4 times the 49 parity bits for a shortened BCH(3 057,3 008)-code with dmin=10, 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 3 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 (equal to stream type 2).

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.

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	RFU_Error_Correction	Reserved for Error Detection or Outer Error Correction Code	196	4×U49	e.g. 4 times the 49 parity bits for a shortened BCH(3 057,3 008)-code with dmin=10, 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 occured
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		

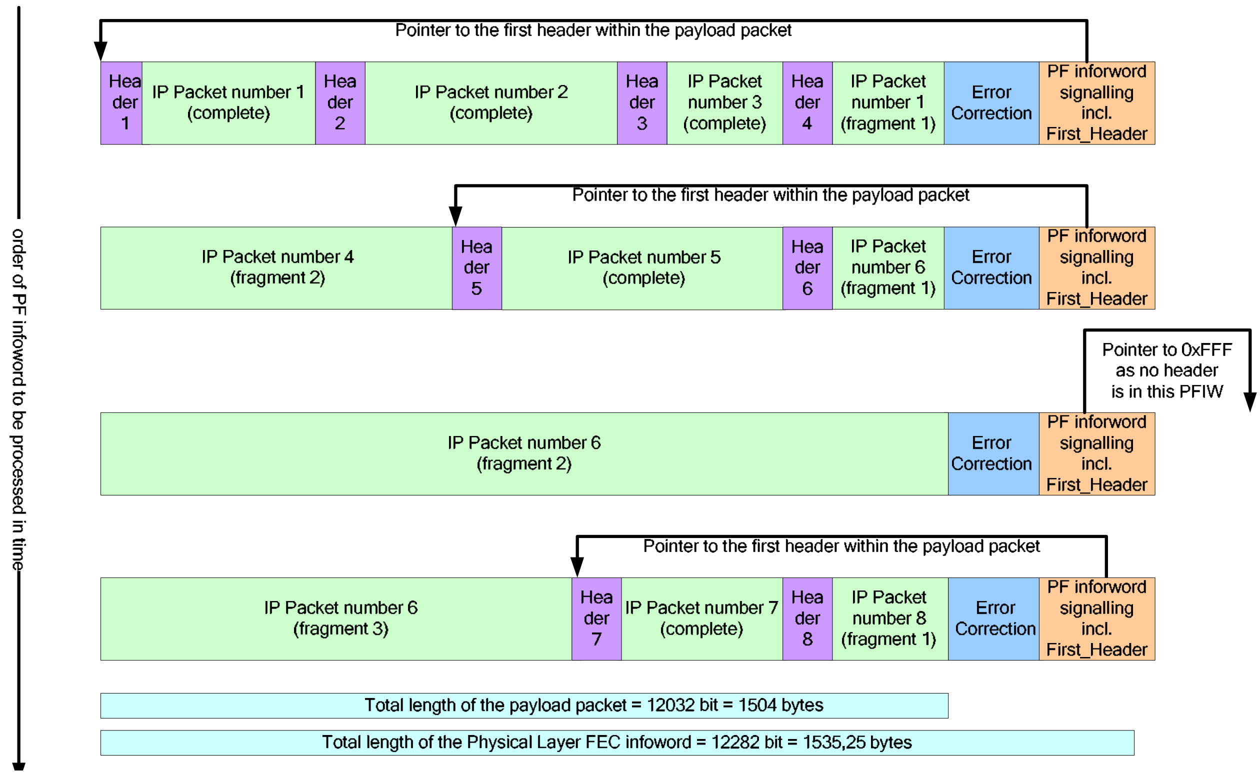

Figure 3: Description of IP packet encapsulation

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 4. 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.

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.

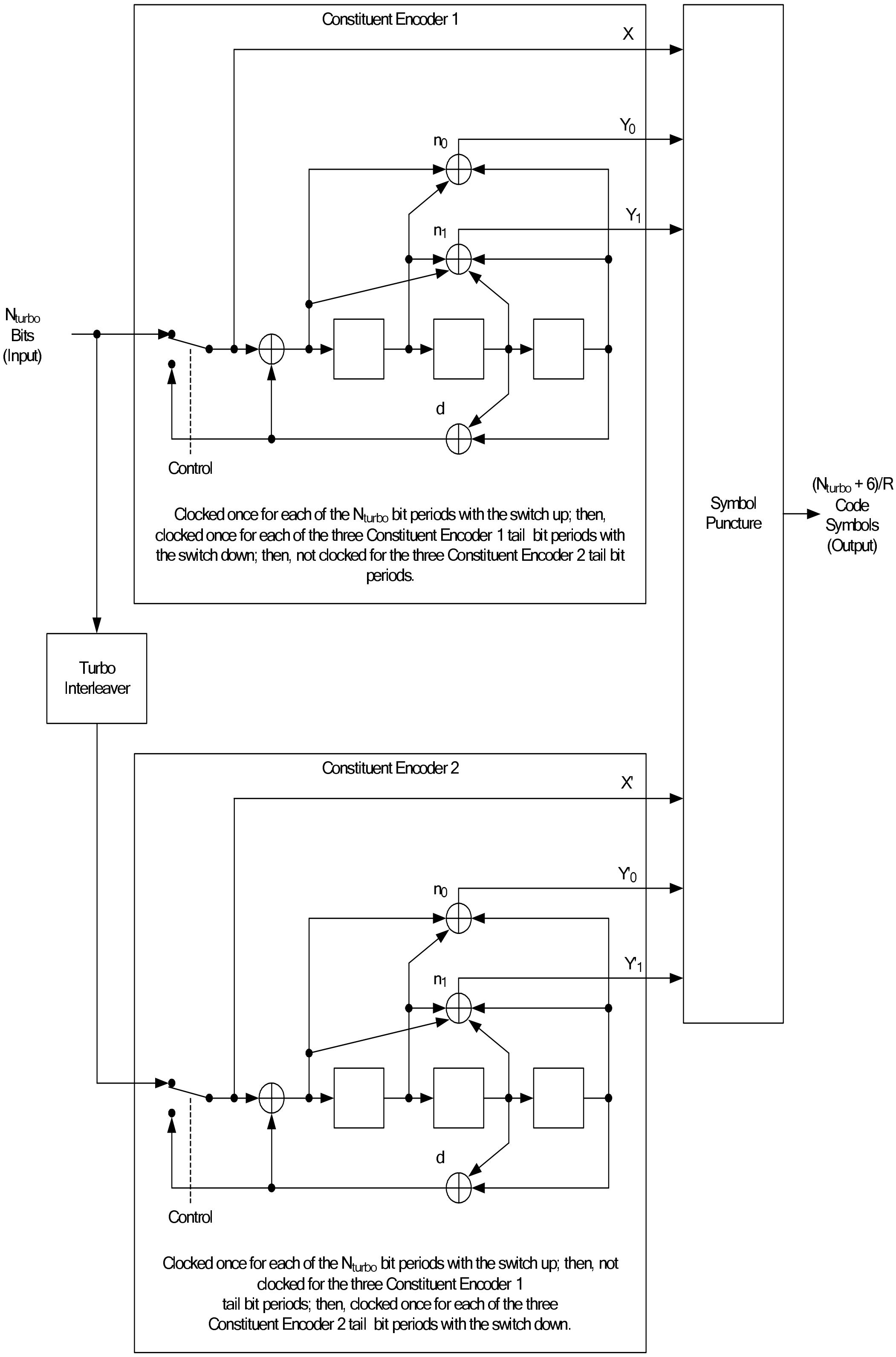

Figure 4: Turbo encoder

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

Punct_Pat_ID										
Code Rate	1/5		1/4		1/3		1/2			
Output										
X	1		11		1		11			
Y_0	1		11		1		10			
Y_1	1		10		0		00			
X'	0		00		0		00			
Y'_0	1		01		1		01			
Y'_1	1		11		0		00			

NOTE 1: For each puncturing pattern, the puncturing table shall be read first from top to bottom and then from left to right.
NOTE 2: Code rates 2/9, 2/7, 3/8, 3/5, 2/3 and 3/4 are under study.

4.4.3 Turbo code termination

The turbo encoder shall generate 6/R 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 4. 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 3/R 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 3/R 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.

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	1/5		1/4		1/3		1/2			
Output										
X	333000		222000		222000		111000			
Y_0	111000		111000		111000		111000			
Y_1	111000		111000		000000		000 000			
X'	000333		000222		000222		000111			
Y'_0	000111		000111		000111		000111			
Y'_1	000111		000111		000000		000000			

NOTE 1: For each puncturing pattern, the puncturing table shall be read first from top to bottom and then from left to right.
NOTE 2: Code rates 2/9, 2/7, 3/8, 3/5, 2/3 and 3/4 are under study.

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 5 and described below:

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

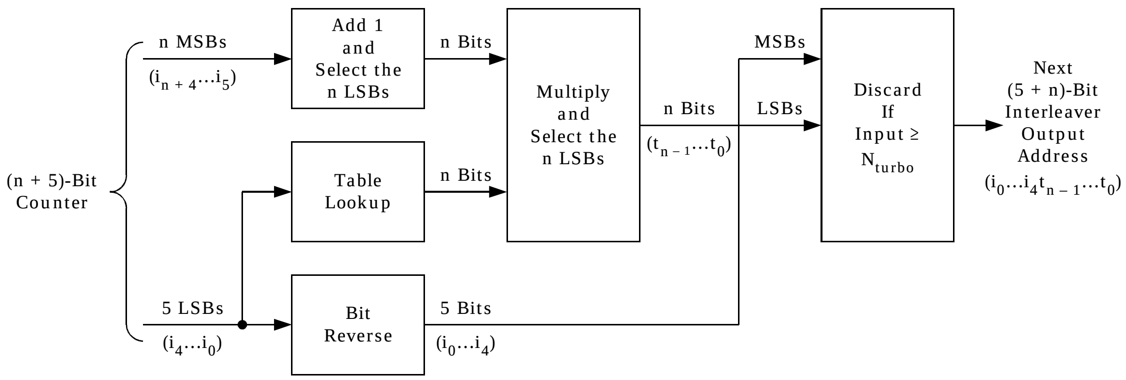

Figure 5: Turbo interleaver output address calculation procedure

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 refered 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 implicitely also defines the code rate) is transmitted in the signalling pipe. Table 11 applies.

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

Punct_Pat_ID	Turbocode coderate	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
NOTE: Some of these puncturing schemes are not needed with the present standard. They are provisions for future extensions that are under study.		

4.4.7 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 of the code rate only, it is not signalled additionally but has to be derived from the parameter Punct_Pat_ID.

As this mixer is used for the payload, only the values necessary for the payload code word sizes are given here.

Table 12: Mixer address increment definition

Code Rate	1/5		1/4		1/3		1/2			
Codeword_Len	61 440		49 152		36 864		24 576			
CILM_Inc	251		217		199		167			
NOTE: Code rates 2/9, 2/7, 3/8, 3/5, 2/3 and 3/4 are under study.										

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 codeword (IU_Per_CW and CU_Per_CW).

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

Code Rate	Number of CU per Codeword (CU_Per_CW)	Number of IU per Codeword (IU_Per_CW)
1/5	30	120
2/9	27	108
1/4	24	96
2/7	21	84
1/3	18	72
3/8	16	64
1/2	12	48
3/5	10	40
2/3	9	36
3/4	8	32

After the chopping, a demultiplexer distributes these IUs to slots. A slot is a sub-stream of IUs that is processed by its individual disperser. The number of slots allocated to an S-TS is denoted by Num_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 Num_Slots codewords have been demultiplexed to Num_Slots slots, the demultiplexer starts with the first slot again and feeds it with the Num_Slots+1 codeword. This behaviour is repeated periodically.

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 `Num_Slots` parallel dispersers for one S-TS. 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 6 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`.

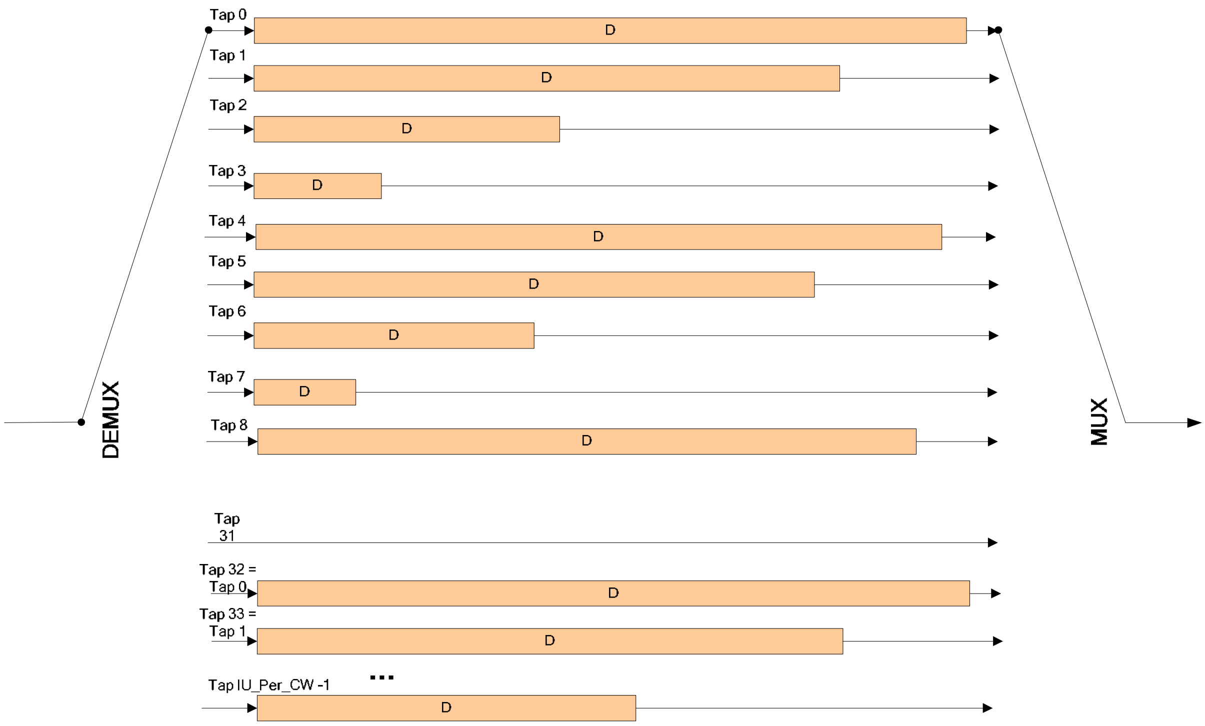

Figure 6: Example of a disperser profile

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 dispersers of those S-TS, which are transmitted in the considered pipe inside the current C-TS frame, and multiplexing these `IUs` together.

Let `Num_Slots[i]` represent the number of slots for the i-th S-TS of the considered pipe inside the current C-TS frame with `i` from `0` to `N-1` (`N` being the total number of S-TS inside the considered pipe in the current C-TS frame). In this case, the collector reads from a total of
`M = Num_Slots[0] + Num_Slots[1] + … + Num_Slots[N-1]`
parallel dispersers for the `N` S-TS, each disperser processing its individual slot. Each slot's length is `IU_Per_CW IUs`. Hence, the considered pipe is exactly `M` slots (or `M × IU_Per_CW IUs`) wide.

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 modi 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 7. The collector multiplexes the disperser outputs with a granularity of one slot, i.e. it first reads all IU_Per_CW IUs from the first slot of S-TS 0 (i.e. from the disperser associated with this slot), then it reads IU_Per_CW IUs from the second slot of S-TS 0 etc. until slot number Num_Slot[0]-1 of S-TS 0. Then it continues with all IU_Per_CW IUs from the first slot of S-TS 1 etc. When slot number Num_Slot[N-1] of S-TS N-1 is read, i.e. all M slots have been read, 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 7. The collector multiplexes the disperser outputs with a granularity of one IU, i.e. it first reads the first IU from the first slot of S-TS 0 (i.e. from the disperser associated with this slot), then it reads the first IU from the second slot of S-TS 0 etc. until slot number Num_Slot[0]-1 of S-TS 0. Then it continues with the first IU from the first slot of S-TS 1 etc. When the first IU from slot number Num_Slot[N-1] of S-TS N-1 is read, i.e. the first IU of all M slots has been read, the collector reads the second IU of all M slots in the same way etc. When the last IU (number IU_Per_CW-1) has been read from slot number Num_Slot[N-1] of S-TS N-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).

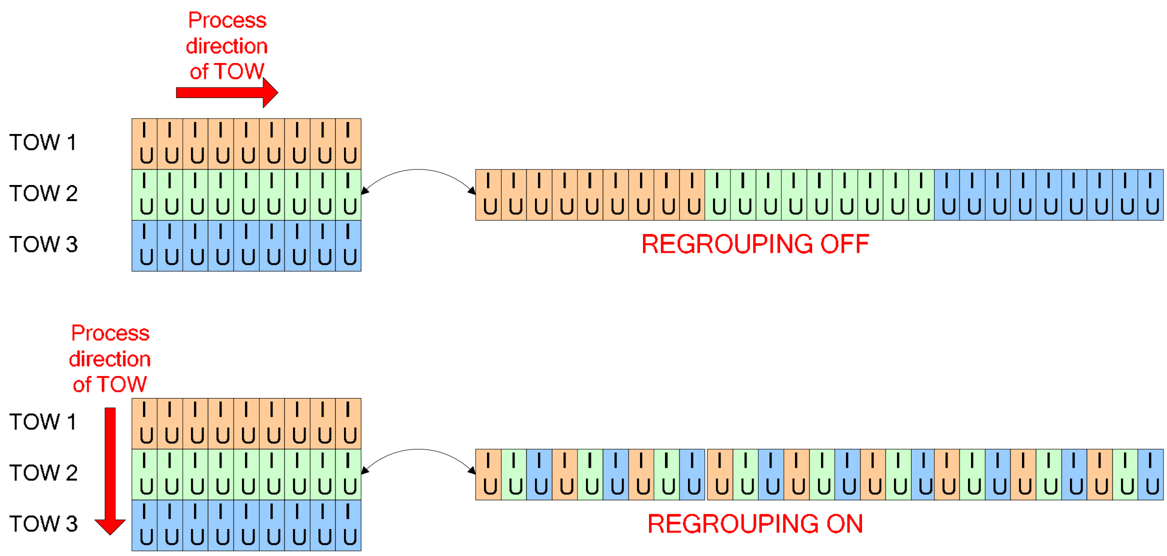

Figure 7: Demonstration of regrouping

4.9 C-TS multiplexer

The C-TS multiplex consists of Capacity Units (CU), whereas 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 8 shows the sub-partitioning of a C-TS frame into its smaller units and into logical content.

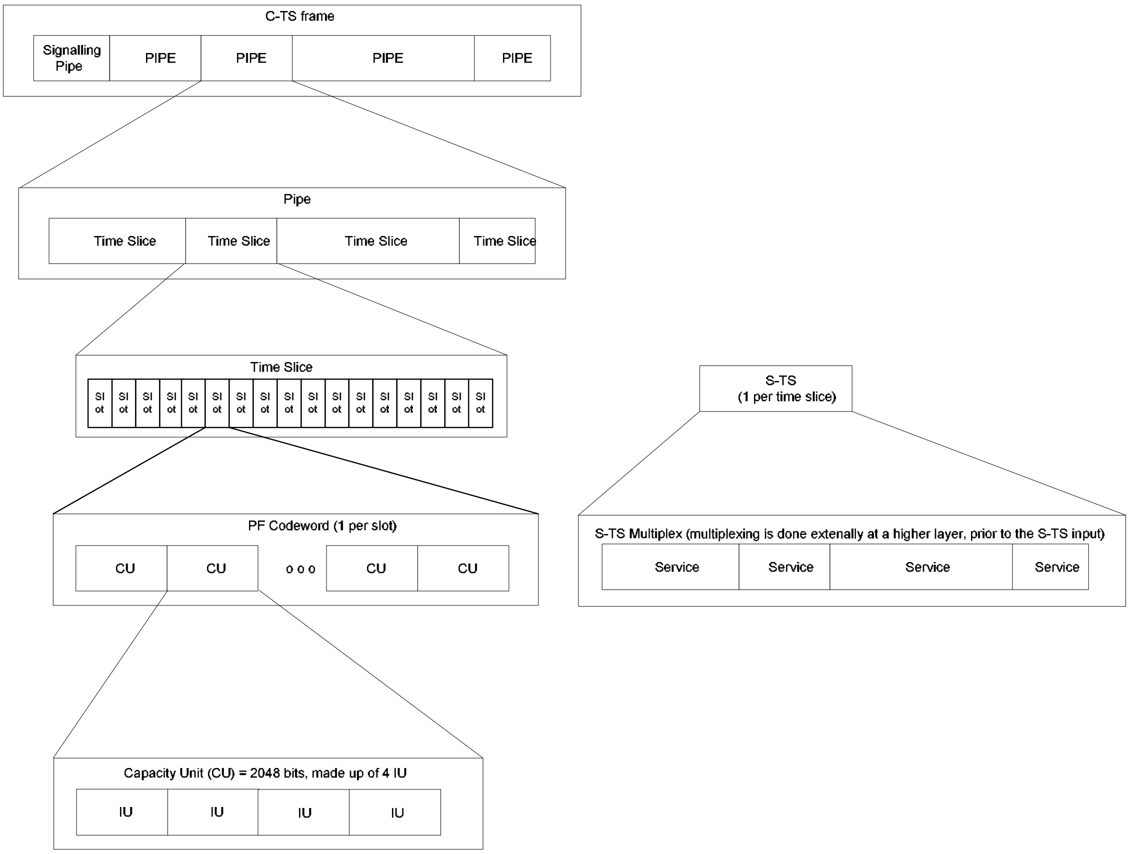

Figure 8: Overview on the C-TS multiplex

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`.

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 occupies exactly 2 CUs.

4.10.1.2 SOF Preamble

The SOF preamble is defined as follows and transmitted row-wise with 0x80 being the first 2 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 alternatingly transmitted in the format specified above and in inverted form. 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

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	40	See Table 16 for further details
…		
Individual parameters for pipe N-1	40	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 (marked light grey)
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 (when a reconfiguration is approaching)	1	U1	This reconfiguration announcement is only used for the pipe multiplex. Reconfiguration of the S-TS scheduling inside the pipes are not announced, except when new S-TS are added to or removed from the C-TS mux.
5	Reconfig_Counter	Countdown (in C-TS frames) until the next pipe configuration becomes active; if 0: no reconfiguration scheduled	11	U11	announcement up to 2 047 frames in advance; i.e. over 14 minutes.
16	Operator_ID	Unique identifier for the network operator	8	U8	
24	RFU		2		
26	Frame_Index	Cyclic index of the current C-TS Frame	22	U22	Is reset to 0 always at time 0:00 UTC.
48	RFU		2		
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 inside this C-TS mux is Num_Pipes_1 +1.
	Total length for parameters:		**56**		

Table 16: Individual parameters of each 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	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 pips is Num_STS_1 + 1.
21	Num_Disperser_Sections_1	Number of disperser sections minus 1	3	U3	The number of disperser sections is Num_Disperser_Sections_1 +1.
24	Disperser_Inv_Flag	Disperser inversion flag	1		Disperser profile has inverted order.
25	Regrouping_Flag	Regrouping flag	1		1: Regrouping on.
26	Tap_Diff_Mult_1_8	Multiplier (minus 0.125) for all tap lengths	6	U3.3	The multiplier value Tap_Diff_Mult is Tap_Diff_Mult_1_8 + 0,125 (note that the number format is with 3 fractional bits).
32	Schedule_Period_Counter	C-TS frame counter inside the schedule period	4		Counter starts with 0 for the C-TS frame where a schedule period starts and is incremented by 1 for each subsequent C-TS frame within the same schedule period.
36	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 configurationss (e.g. for Variable Bit Rate)..
	Total length for parameters:		40		

Table 17: Individual parameters of each 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.
6	Pipe_Width_CUs	Pipe width in CUs	9	U9	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.
16	RFU		0		
	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	U2	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	U4	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.
12	RFU		0		
	Total length for parameters:		**12**		

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`.

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`, 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
```

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 schedule period. 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. One schedule period shall always start in the first slot of the considered pipe in a C-TS frame, and it shall always occupy the considered pipe for an integer multiple of C-TS frames, i.e. `Schedule_Period_Length_Slots[i]` must be an integer multiple of `Pipe_Width_Slots[i]`. 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 all 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]
```

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. No advance signalling is envisaged here. 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 `Reconfig_Counter`, although the announcement can be on much shorter notice than for a pipe reconfiguration. Moreover, the parameters `Schedule_Index` 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. The network operator shall synchronize these S-TS in such a way that they can be diversity combined in the receivers.

Only the S-TS with `STS_ID = 0` plays a different role: this S-TS is C-TS 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 paramater `Num_Disperser_Sections_1`, the number of disperser sections `Num_Disperser_Sections` is calculated by:

`Num_Disperser_Sections = Num_Disperser_Sections_1 + 1`.

The tap length difference multiplier `Tap_Diff_Mult` is calculated by:

```
Tap_Diff_Mult = Tap_Diff_Mult_1_8 + 0.125
```

Observe that the parameter `Tap_Diff_Mult_1_8` is transmitted inside in the signalling pipe in format U3.3 (i.e. with 3 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*. The disperser profile of the transmitter 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] = floor(Intermed_Len[i][j] * Tap_Diff_Mult)`,

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 as the other pipes but has no dispersing and cannot be time-sliced and has not the restriction that the length is an integer multiple of codewords. This may lead to rotating codeword starts inside the pipe, which in turn needs a flag Codeword_Align_Flag for this inside parameters of this pipe.

4.10.9 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).

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

`Reconfig_Counter` 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 parameters change; possibly the length of the disperser section parameter list changes (in this case the structure of the signalling pipe infoword changes).
Signalisation	No advance signalling; event increments the parameter `Schedule_Index` in the concerned pipe.	Advance signalisation as for pipe reconfiguration; event increments the parameters `Schedule_Index` in all pipes and `Config_Index.`	Advance signalisation by the parameters `Reconfig_Flag` and `Reconfig_Counter`; 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 9, 10 and 11. 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.

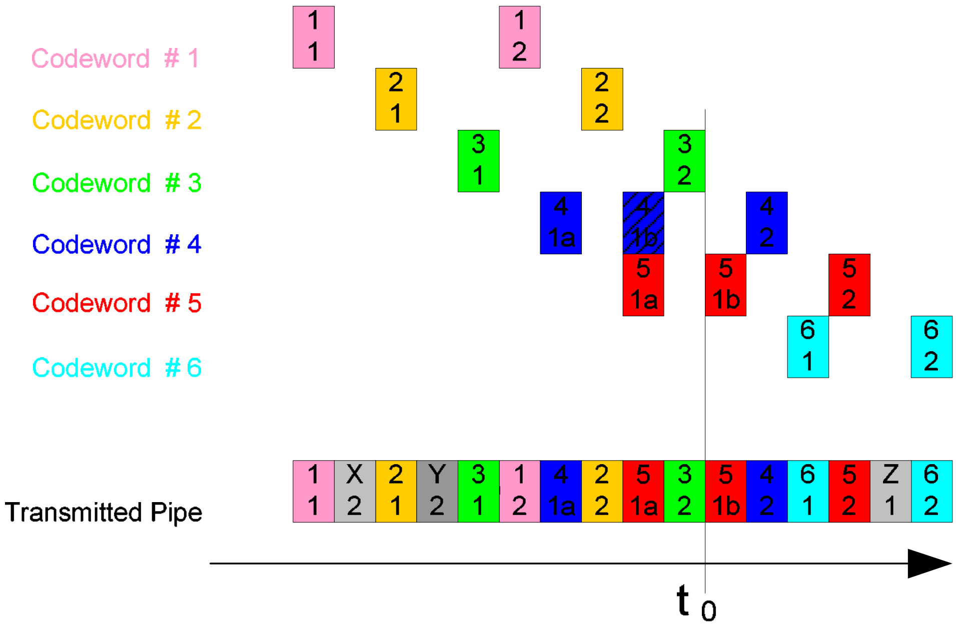

Figure 9: Shorter disperser delay after reconfiguration

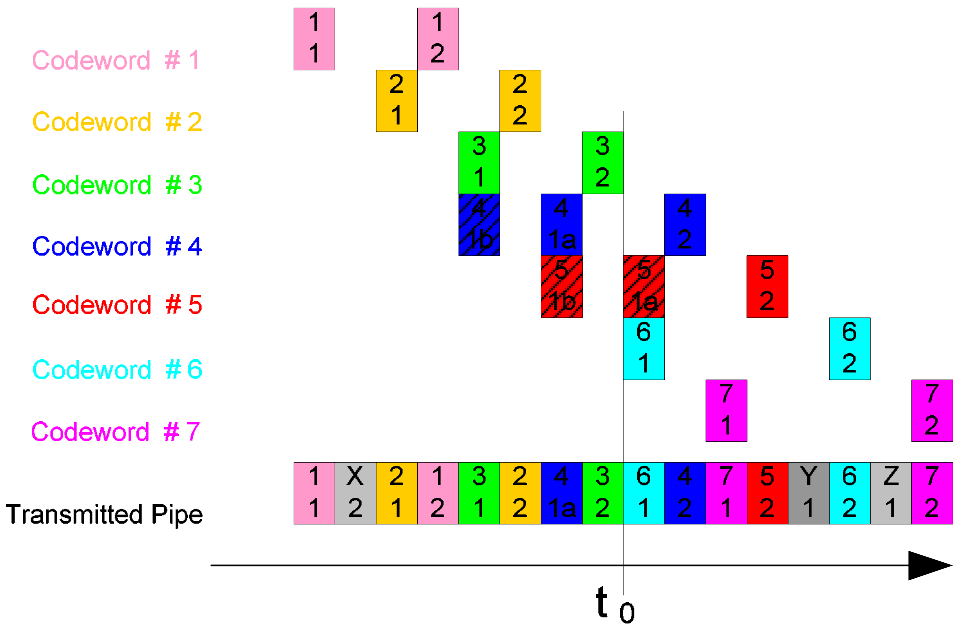

Figure 10: Longer disperser delay after reconfiguration

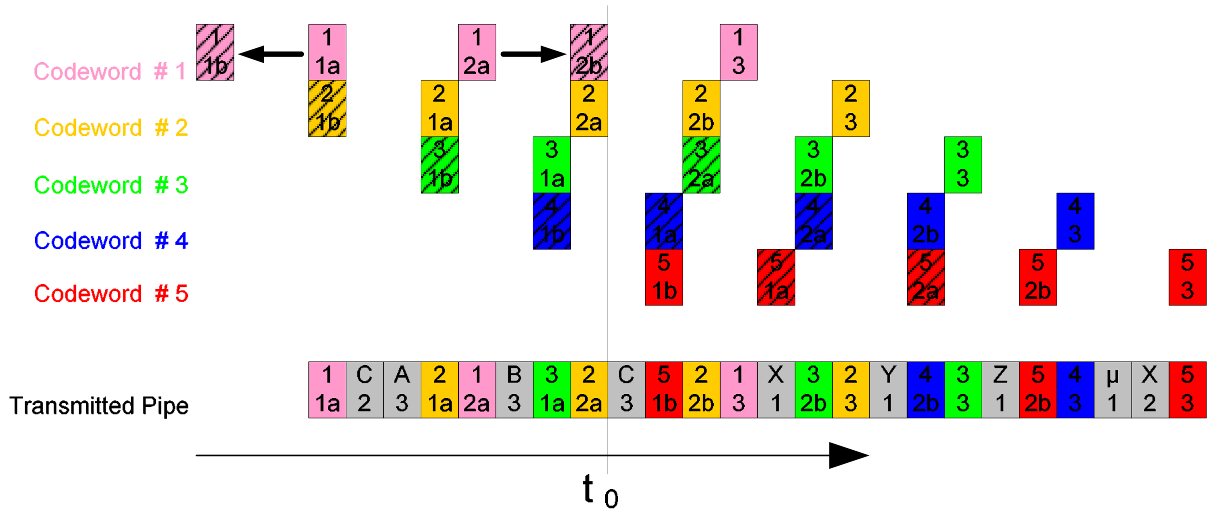

Figure 11: Other example of disperser reconfiguration

- 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.
- Throughput of S-TS changes (i.e. changing pipe width in case of constant code rate); this is a simple re-allocation/re-scheduling of the slots to all S-TS of the pipe:
  - If total throughput of pipe remains constant: no changes.
  - If total throughput of the pipe becomes higher: `sum` of `Num_Slots` increases; new slots are added after the existing slots of pipe and new dispersers are initialized with content zero, new slot allocation becomes effective at reconfiguration instant (i.e. receiver outputs higher throughput only after the end-to-end delays of the interleaver).
  - If total throughput of pipe becomes lower: `sum` of `Num_Slots` decreases; last slots are killed, new slot allocation becomes effective before reconfiguration instant (that instant minus the interleaver's end-to-end delay); flushing of the dying slots 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 slots; after the reconfiguration instant, the superfluous slots are dead and discarded.

Annex A (informative): Time slicing can be achieved on two layers

- Physical Layer: since each time slice is a one-to-one mapping of an S-TS, the PL can simply extract the desired S-TS ID from the C-TS multiplex and hand it over to the SL. Switching on and off of the analog and digital components inside the PL is handled by the PL itself.

- Link Layer: this mode is not supported by this standard.

Annex B (informative): Bibliography

ETSI TR 102 525: " Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) service; Functionalities, architecture and technologies, Technologies for the radio interface."

ETSI TS 102 551-1: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Inner Physical Layer of the Radio Interface Part 1 - Single Carrier Transmission".

ETSI TS 102 551-2: "Satellite Earth Stations and Systems (SES); Satellite Digital Radio (SDR) Systems; Inner Physical Layer of the Radio Interface Part 2; Multiple Carrier Transmission".

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

ISO/IEC 13818-1: "Information Technology - Generic Coding of moving pictures and associated audio - Part 1: Systems".

History

Document history		
V1.1.1	November 2006	Publication