

Digital Radio Mondiale (DRM); System Specification

European Broadcasting Union



Union Européenne de Radio-Télévision

EBU·UER



Reference

RES/JTC-DRM-07

Keywords

broadcasting, digital, DRM, radio

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, send your comment to:

editor@etsi.org

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 2004.

© European Broadcasting Union 2004.

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	8
Foreword.....	8
Introduction	8
1 Scope	10
2 References	10
3 Definitions, symbols, abbreviations and convention.....	10
3.1 Definitions	10
3.2 Symbols.....	11
3.3 Abbreviations	12
3.4 Convention	13
4 General characteristics	13
4.1 System overview	13
4.2 System architecture	13
4.3 Source coding.....	15
4.4 Transmission modes.....	16
4.4.1 Signal bandwidth related parameters	16
4.4.2 Transmission efficiency related parameters.....	16
4.4.2.1 Coding rates and constellations.....	16
4.4.2.2 OFDM parameter set.....	16
5 Source coding modes.....	18
5.1 Overview	18
5.1.1 AAC audio coding	18
5.1.2 MPEG CELP coding.....	19
5.1.3 MPEG HVXC coding.....	19
5.1.4 SBR coding.....	20
5.1.5 PS coding.....	20
5.2 UEP and audio super framing.....	20
5.3 AAC coding.....	21
5.3.1 AAC.....	21
5.3.1.1 AAC audio super frame	22
5.3.2 AAC + SBR	24
5.4 MPEG CELP coding	25
5.4.1 MPEG CELP	25
5.4.1.1 CELP audio super frame	26
5.4.2 CELP + SBR.....	28
5.5 HVXC coding.....	29
5.5.1 Definitions	29
5.5.1.1 HVXC source coder parameters.....	29
5.5.1.2 CRC bits for fixed bit rate modes.....	29
5.5.2 HVXC decoder	30
5.5.3 HVXC encoder	30
5.5.3.1 LPC analysis and LSP quantization	31
5.5.3.2 Open loop pitch search.....	31
5.5.3.3 Harmonic magnitude and fine pitch estimation.....	31
5.5.3.4 Vector quantization of harmonic magnitudes	32
5.5.3.5 Voiced/Unvoiced decision	32
5.5.3.6 VXC coding of unvoiced signals	32
5.5.4 HVXC channel coding.....	32
5.5.4.1 Protected bit selection	32
5.5.4.2 Syntax of DRM HVXC error robustness (ErHVXCfixframe_CRC)	34
5.5.5 Category interleaving.....	37
5.5.6 HVXC error detection and concealment.....	38
5.5.6.1 Cyclic Redundancy Check (CRC).....	38

5.5.6.2	Error concealment	39
5.5.6.2.1	Parameter replacement	39
5.5.7	HVXC + SBR	41
5.6	SBR coding	41
5.6.1	AAC + SBR	41
5.6.2	CELP/HVXC + SBR	41
5.6.2.1	SBR Payload	41
5.6.2.1.1	Bitstream Syntax	42
5.6.2.1.2	Bitstream element definitions	43
5.6.2.2	SBR decoding process	44
5.6.2.2.1	Notation, Constants and Defaults	44
5.6.2.2.2	Frequency band tables	45
5.6.2.2.3	Time / Frequency grid	46
5.6.2.2.4	Envelope and noise floor decoding and dequantization	46
5.6.2.3	SBR filterbanks and tool overview	48
5.6.2.4	HF Generation and HF adjustment	50
5.7	Parametric Stereo coding	52
5.7.1	Introduction	52
5.7.2	Technical overview	52
5.7.2.1	Stereo parameters	52
5.7.2.2	Signalling	52
5.7.3	Bitstream protocol	53
5.7.3.1	Bitstream integration	53
5.7.3.2	PS syntax	53
5.7.3.3	Bitstream element definitions	55
5.7.4	PS Decoding	56
5.7.4.1	Decoding overview	56
5.7.4.2	Dequantization and mapping	57
5.7.4.3	SA decoding	58
5.7.4.4	Pan decoding	61
6	Multiplex definition	61
6.1	Introduction	61
6.2	Main Service Channel (MSC)	62
6.2.1	Introduction	62
6.2.2	Structure	62
6.2.3	Building the MSC	62
6.2.3.1	Multiplex frames	62
6.2.3.2	Hierarchical frames	63
6.2.4	Reconfiguration	63
6.3	Fast Access Channel (FAC)	63
6.3.1	Introduction	63
6.3.2	Structure	63
6.3.3	Channel parameters	63
6.3.4	Service parameters	65
6.3.5	CRC	67
6.3.6	FAC repetition	67
6.4	Service Description Channel (SDC)	67
6.4.1	Introduction	67
6.4.2	Structure	67
6.4.3	Data entities	68
6.4.3.1	Multiplex description data entity - type 0	69
6.4.3.2	Label data entity - type 1	70
6.4.3.3	Conditional access parameters data entity - type 2	70
6.4.3.4	Alternative frequency signalling: Multiple frequency network information data entity - type 3	71
6.4.3.5	Alternative frequency signalling: Schedule definition data entity - type 4	73
6.4.3.6	Application information data entity - type 5	74
6.4.3.7	Announcement support and switching data entity - type 6	75
6.4.3.8	Alternative frequency signalling: Region definition data entity - type 7	76
6.4.3.9	Time and date information data entity - type 8	77
6.4.3.10	Audio information data entity - type 9	78
6.4.3.11	FAC channel parameters data entity - type 10	80

6.4.3.12	Alternative frequency signalling: Other services data entity - type 11.....	82
6.4.3.13	Language and country data entity - type 12	85
6.4.3.14	Other data entities	85
6.4.4	Summary of data entity characteristics	85
6.4.5	Changing the content of the SDC	86
6.4.6	Signalling of reconfigurations.....	87
6.4.6.1	Service reconfigurations.....	87
6.4.6.2	Channel reconfigurations	87
6.5	Text message application	87
6.5.1	Structure.....	88
6.6	Packet mode	89
6.6.1	Packet structure.....	90
6.6.1.1	Header.....	90
6.6.1.2	Data field.....	90
6.6.2	Asynchronous streams	91
6.6.3	Files	91
6.6.4	Choosing the packet length.....	91
7	Channel coding and modulation.....	91
7.1	Introduction	91
7.2	Transport multiplex adaptation and energy dispersal	92
7.2.1	Transport multiplex adaptation	92
7.2.1.1	MSC	92
7.2.1.2	FAC.....	95
7.2.1.3	SDC.....	95
7.2.2	Energy dispersal.....	95
7.3	Coding	96
7.3.1	Multilevel coding.....	96
7.3.1.1	Partitioning of bitstream in SM.....	99
7.3.1.2	Partitioning of bitstream in HMsym.....	100
7.3.1.3	Partitioning of bitstream in HMmix	100
7.3.2	Component code	101
7.3.3	Bit interleaving	106
7.3.3.1	FAC.....	107
7.3.3.2	SDC.....	107
7.3.3.3	MSC	107
7.4	Signal constellations and mapping	109
7.5	Application of coding to the channels	113
7.5.1	Coding the MSC	113
7.5.1.1	SM.....	113
7.5.1.2	HMsym	114
7.5.1.3	HMmix	114
7.5.2	Coding the SDC.....	115
7.5.3	Coding the FAC.....	115
7.6	MSC cell interleaving.....	115
7.7	Mapping of MSC cells on the transmission super frame structure	117
8	Transmission structure	118
8.1	Transmission frame structure and modes	118
8.2	Propagation-related OFDM parameters.....	120
8.3	Signal bandwidth related parameters.....	120
8.3.1	Parameter definition.....	120
8.3.2	Simulcast transmission	122
8.4	Pilot cells.....	122
8.4.1	Functions and derivation.....	122
8.4.2	Frequency references	122
8.4.2.1	Cell positions.....	122
8.4.2.2	Cell gains and phases	123
8.4.3	Time references	123
8.4.3.1	Cell positions and phases	124
8.4.3.2	Cell gains	125
8.4.4	Gain references	126

8.4.4.1	Cell positions.....	126
8.4.4.2	Cell gains	126
8.4.4.3	Cell phases	126
8.4.4.3.1	Procedure for calculation of cell phases	127
8.4.4.3.2	Robustness Mode A.....	127
8.4.4.3.3	Robustness Mode B.....	128
8.4.4.3.4	Robustness Mode C.....	128
8.4.4.3.5	Robustness Mode D.....	128
8.5	Control cells	129
8.5.1	General.....	129
8.5.2	FAC cells	129
8.5.2.1	Cell positions.....	129
8.5.2.2	Cell gains and phases	131
8.5.3	SDC cells	131
8.5.3.1	Cell positions.....	131
8.5.3.2	Cell gains and phases	131
8.6	Data cells	132
8.6.1	Cell positions	132
8.6.2	Cell gains and phases.....	132
Annex A (informative): Simulated system performance.....		133
Annex B (informative): Definition of channel profiles.....		134
Annex C (informative): Example of mapping of logical frames to multiplex frames.....		136
Annex D (normative): Calculation of the CRC word		137
Annex E (informative): RF protection ratios.....		139
Annex F (informative): Alternative Frequency and announcement signalling		143
F.1	Possibilities of the Alternative Frequency Signalling feature	143
F.2	Possibilities of the announcement feature	144
F.3	SDC data entities overview for Alternative Frequency and announcement signalling	145
F.4	SDC data entities and setup for alternative frequency signalling.....	146
F.5	SDC data entities and setup for announcement.....	146
F.6	Alternative frequency and announcement signalling - coding example.....	147
Annex G (informative): Guidelines for receiver implementation.....		150
G.1	Alternative Frequency checking and Switching (AFS).....	150
G.2	Station buttons for DRM services	151
G.3	Seamless Alternative Frequency checking and Switching (AFS).....	151
G.4	Character sets	154
Annex H (informative): Service capacity and bit rates		155
Annex I (normative): SBR tables.....		156
I.1	Predefined tunings for CELP + SBR.....	156
I.2	Predefined tunings for HVXC + SBR	157
I.3	Parametric stereo Huffman tables	159
I.4	Miscellaneous parametric stereo tables	161
Annex J (informative): Numbers of input bits.....		164

Annex K (informative):	Simulcast transmission, alternate sources, and enhancement signalling	169
Annex L (informative):	Pilot reference illustrations	172
Annex M (informative):	MSC configuration examples.....	176
Annex N (informative):	HVXC parameters	179
Annex O (normative):	Interpretation of schedules for Alternative Frequency Signalling.....	180
Annex P (informative):	Bibliography.....	182
History		183

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 ETSI Standard (ES) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

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

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

With respect to the first edition of ES 201 980 published in April 2003, the present document contains several refinements to the DRM system. These refinements were performed and agreed by the DRM Consortium and include the following areas:

- audio coding clauses are brought into line with the latest MPEG 4 standards;
- signalling of alternate frequency information is revised;
- clarification of text following field use of the DRM system.

Introduction

The frequency bands used for broadcasting below 30 MHz are:

- Low Frequency (LF) band: from 148,5 kHz to 283,5 kHz, in ITU Region 1 [1] only;
- Medium Frequency (MF) band: from 526,5 kHz to 1 606,5 kHz, in ITU Regions 1 [1] and 3 [1] and from 525 kHz to 1 705 kHz in ITU Region 2 [1];
- High Frequency (HF) band: a set of individual broadcasting bands in the frequency range 2,3 MHz to 27 MHz, generally available on a Worldwide basis.

These bands offer unique propagation capabilities that permit the achievement of:

- large coverage areas, whose size and location may be dependent upon the time of day, season of the year or period in the (approximately) 11 year sunspot cycle;
- portable and mobile reception with relatively little impairment caused by the environment surrounding the receiver.

There is thus a desire to continue broadcasting in these bands, perhaps especially in the case of international broadcasting where the HF bands offer the only reception possibilities which do not also involve the use of local repeater stations.

However, broadcasting services in these bands:

- use analogue techniques;
- are subject to limited quality;
- are subject to considerable interference as a result of the long-distance propagation mechanisms which prevail in this part of the frequency spectrum and the large number of users.

As a direct result of the above considerations, there is a desire to effect a transfer to digital transmission and reception techniques in order to provide the increase in quality which is needed to retain listeners who, increasingly, have a wide variety of other programme reception media possibilities, usually already offering higher quality and reliability.

In order to meet the need for a digital transmission system suitable for use in all of the bands below 30 MHz, the Digital Radio Mondiale (DRM) consortium was formed in early 1998. The DRM consortium is a non-profit making body which seeks to develop and promote the use of the DRM system worldwide. Its members include broadcasters, network providers, receiver and transmitter manufacturers and research institutes. More information is available from their website (<http://www.drm.org/>).

1 Scope

The present document gives the specification for the Digital Radio Mondiale (DRM) system for digital transmissions in the broadcasting bands below 30 MHz.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- 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.
- For a non-specific reference, the latest version applies.

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

- [1] ITU Radio Regulations.
- [2] ISO/IEC 14496-3: "Information technology - Coding of audio-visual objects - Part 3: Audio".
- [3] ETSI EN 300 401: "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
- [4] ISO/IEC 10646-1: "Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane".
- [5] ISO 639-2: "Codes for the representation of names of languages - Part 2: Alpha-3 code".
- [6] ISO 3166 (all parts): "Codes for the representation of names of countries and their subdivisions".
- [7] ISO/IEC 8859-1: "Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1".
- [8] ETSI TS 101 968: "Digital Radio Mondiale (DRM); Data applications directory".
- [9] ITU-R Recommendation BS.1615: "'Planning parameters" for digital sound broadcasting at frequencies below 30 MHz".

3 Definitions, symbols, abbreviations and convention

3.1 Definitions

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

cell: sine wave portion of duration T_s , transmitted with a given amplitude and phase and corresponding to a carrier position

NOTE: Each OFDM symbol is the sum of K such sine wave portions equally spaced in frequency.

energy dispersal: operation involving deterministic selective complementing of bits in the logical frame, intended to reduce the possibility that systematic patterns result in unwanted regularity in the transmitted signal

Fast Access Channel (FAC): channel of the multiplex data stream which contains the information that is necessary to find services and begin to decode the multiplex

logical frame: contains data of one stream during 400 ms

Main Service Channel (MSC): channel of the multiplex data stream which occupies the major part of the transmission frame and which carries all the digital audio services, together with possible supporting and additional data services

mod: the modulo operator

NOTE: $(x \bmod y) = z$, where $y > 0$, such that $x = qy + z$, q is an integer, and $0 \leq z < y$.

multiplex frame: logical frames from all streams form a multiplex frame (duration of 400 ms)

NOTE: It is the relevant basis for coding and interleaving.

OFDM symbol: transmitted signal for that portion of time when the modulating amplitude and phase state is held constant on each of the equally-spaced carriers in the signal

reserved for future addition (rfa): bits with this designation shall be set to zero

NOTE: Receivers shall ignore these bits.

reserved for future use (rfu): bits with this designation shall be set to zero

NOTE: Receivers shall check that these bits are zero in order to determine the valid status of the other fields in the same scope.

Service Description Channel (SDC): channel of the multiplex data stream which gives information to decode the services included in the multiplex

NOTE: The SDC also provides additional information to enable a receiver to find alternative sources of the same data.

Single Frequency Network (SFN): network of transmitters sharing the same radio frequency to achieve a large area coverage

transmission frame: number of consecutive OFDM symbols (duration of 400 ms), whereby the first OFDM symbol contains the time reference cells

transmission super frame: three consecutive transmission frames (duration of 1 200 ms), whereby the first OFDM symbols contain the SDC block

3.2 Symbols

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

$E[]$	expectation value of the expression in brackets
f_c	reference frequency of the emitted signal
K	number of active carriers in the OFDM symbol
K_{\max}	carrier index of the upper active carrier in the OFDM signal
K_{\min}	carrier index of the lower active carrier in the OFDM signal
L_{MUX}	number of input bits per multiplex frame for the multilevel encoding
N_{MUX}	number of MSC cells (QAM symbols) per multiplex frame
T	elementary time period, equal to $83^{1/3} \mu\text{s}$ (1/12 kHz)
T_f	duration of the transmission frame, equal to 400 ms
T_g	duration of the guard interval
T_s	duration of an OFDM symbol
T_{sf}	duration of the transmission super-frame built from three transmission frames
T_u	duration of the useful (orthogonal) part of an OFDM symbol, excluding the guard interval
X^*	complex conjugate of value X
$\lceil \]$	round towards plus infinity
$\lfloor \]$	round towards minus infinity

3.3 Abbreviations

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

AAC	Advanced Audio Coding
AF	Audio Frequency
AFS	Alternative Frequency Switching
BER	Bit Error Rate
CELP	Code Excited Linear Prediction
CI	Continuity Index
CRC	Cyclic Redundancy Check
DRM	Digital Radio Mondial
EEP	Equal Error Protection
ESC	Error Sensitivity Categories
FAC	Fast Access Channel
HF	High Frequency
HMmix	mixed Hierarchical Mapping
HMsym	symmetrical Hierarchical Mapping
HVXC	Harmonic Vector eXcitation Coding
IFFT	Inverse Fast Fourier Transform
ISO	International Organization for Standardization
LAV	Largest Absolute Value
LF	Low Frequency
LPC	Linear Predictive Coding
LSb	Least Significant bit
LSP	Line Spectral Pairs
MF	Medium Frequency
MPEG	Moving Picture Experts Group
MSb	Most Significant bit
MSC	Main Service Channel
OFDM	Orthogonal Frequency Division Multiplexing
Pan	Panorama
PNS	Perceptual Noise Substitution
PPI	Padded Packet Indicator
PRBS	Pseudo-Random Binary Sequence
PS	Parametric Stereo
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
rfa	reserved for future addition
rfu	reserved for future use
SA	Stereo Ambience
SBR	Spectral Band Replication
SDC	Service Description Channel
SFN	Single Frequency Network
SI	Side Information
SM	Standard Mapping
SPP	Standard Protected Part
TNS	Temporal Noise Shaping
UEP	Unequal Error Protection
uimsbf	unsigned integer most significant bit first
VSPP	Very Strongly Protected Part

3.4 Convention

Unless otherwise stated, the following convention, regarding the order of bits within each step of processing is used:

- in figures, the bit shown in the left hand position is considered to be first;
- in tables, the bit shown in the left hand position is considered to be first;
- in numerical fields, the Most Significant bit (MSb) is considered to be first and denoted by the higher number. For example, the MSb of a single byte is denoted "b7" and the Least Significant bit (LSb) is denoted "b0";
- in vectors (mathematical expressions), the bit with the lowest index is considered to be first.

4 General characteristics

4.1 System overview

The DRM system is designed to be used at any frequency below 30 MHz, i.e. within the long wave, medium wave and short wave broadcasting bands, with variable channelization constraints and propagation conditions throughout these bands. In order to satisfy these operating constraints, different transmission modes are available. A transmission mode is defined by transmission parameters classified in two types:

- signal bandwidth related parameters;
- transmission efficiency related parameters.

The first type of parameters defines the total amount of frequency bandwidth for one transmission. Efficiency related parameters allow a trade-off between capacity (useful bit rate) and ruggedness to noise, multipath and Doppler.

4.2 System architecture

This clause gives a general presentation of the system architecture, based on the synoptic diagram of figure 1, which gives reference to the clauses defining the individual parts of the system.

Figure 1 describes the general flow of different classes of information (audio, data, etc.) and does not differentiate between different services that may be conveyed within one or more classes of information. A detailed description on the distribution of services onto those classes can be found in clause 6.

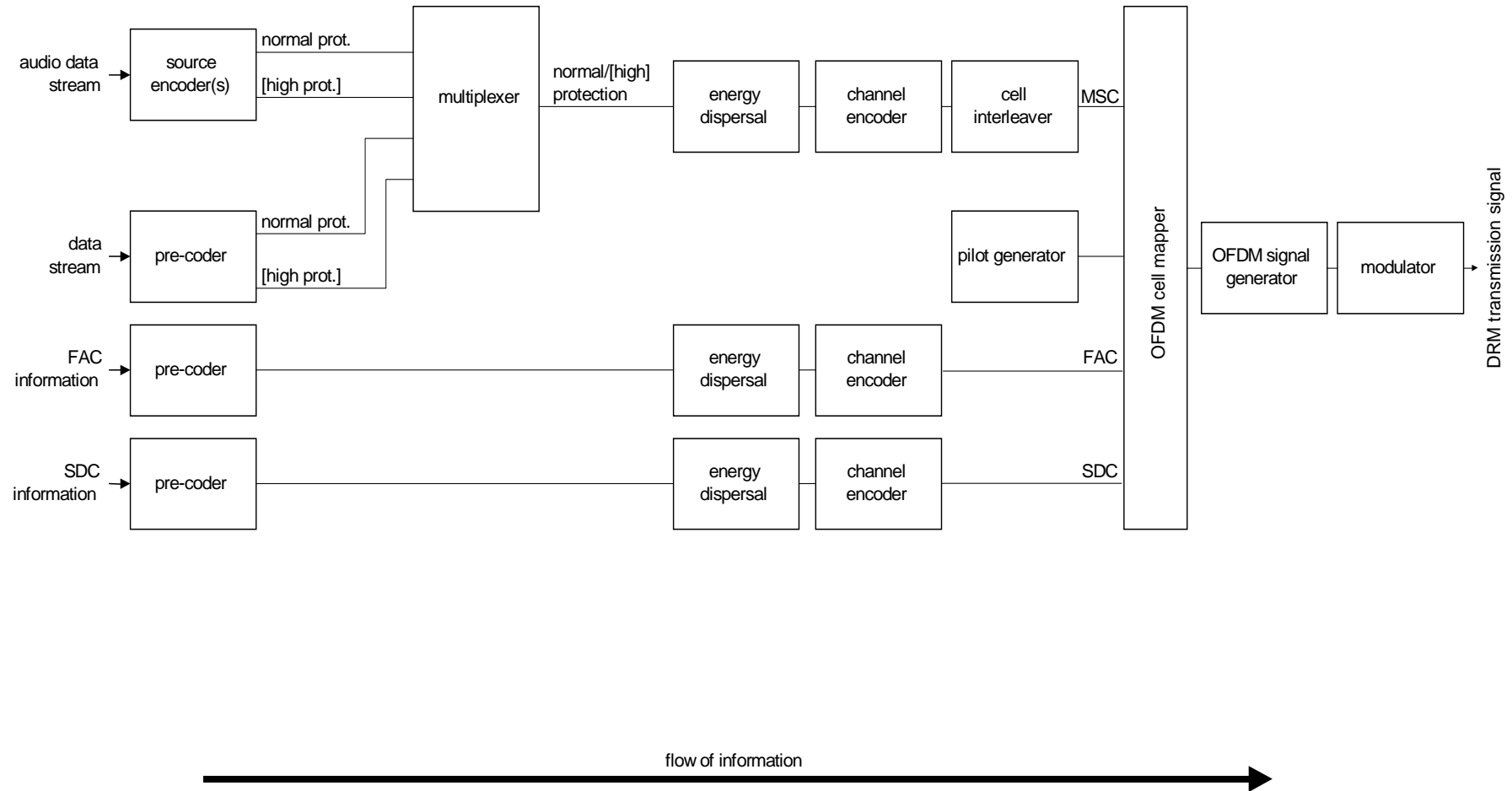


Figure 1: Conceptual DRM transmission block diagram

The source encoder and pre-coders ensure the adaptation of the input streams onto an appropriate digital transmission format. For the case of audio source encoding, this functionality includes audio compression techniques as described in clauses 4.3 and 5. The output of the source encoder(s) and the data stream pre-coder may comprise two parts requiring different levels of protection within the subsequent channel encoder. All services have to use the same two levels of protection.

The multiplexer combines the protection levels of all data and audio services as described in clause 6.

The energy dispersal provides a deterministic selective complementing of bits in order to reduce the possibility that systematic patterns result in unwanted regularity in the transmitted signal.

The channel encoder adds redundant information as a means for quasi error-free transmission and defines the mapping of the digital encoded information onto QAM cells as described in clause 7.

Cell interleaving spreads consecutive QAM cells onto a sequence of cells quasi-randomly separated in time and frequency, in order to provide robust transmission in time-frequency dispersive channels. The pilot generator provides means to derive channel state information in the receiver, allowing for a coherent demodulation of the signal.

The OFDM cell mapper collects the different classes of cells and places them on the time-frequency grid as specified in clause 7.

The OFDM signal generator transforms each ensemble of cells with same time index to a time domain representation of the signal. Consecutively, the OFDM symbol is obtained from this time domain representation by inserting a guard interval as a cyclic repetition of a portion of the signal, as specified in clause 7.

The modulator converts the digital representation of the OFDM signal into the analogue signal in the air. This operation involves digital-to-analogue conversion and filtering that have to comply with spectrum requirements as described in annex E.

4.3 Source coding

Within the constraints of broadcasting regulations in broadcasting channels below 30 MHz and the parameters of the coding and modulation scheme applied, the bit rate available for source coding is in the range from 8 kbit/s (half channels) to ≈ 20 kbit/s (standard channels) to up to ≈ 72 kbit/s (double channels).

In order to offer optimum quality at a given bit rate, the system offers different source coding schemes:

- a subset of MPEG-4 AAC (Advanced Audio Coding) including error robustness tools for generic mono and stereo audio broadcasting;
- a subset of MPEG-4 CELP speech coder for error robust speech broadcasting in mono, for cases when only a low bit rate is available or especially high error robustness is required;
- a subset of MPEG-4 HVXC speech coding for very low bit rate and error robust speech broadcasting in mono, especially well suited also for speech data base applications;
- Spectral Band Replication (SBR), an audio coding enhancement tool that allows the full audio bandwidth to be achieved at low bit rates. It can be applied to AAC, CELP and HVXC;
- Parametric Stereo (PS), an audio coding enhancement tool relevant to SBR that allows for stereo coding at low bit rates.

The bit-stream transport format of the source coding schemes has been modified to meet the requirements of the DRM system (audio superframing). Unequal Error Protection (UEP) can be applied to improve the system behaviour in error prone channels.

Provision is made for further enhancement of the audio system by linking two DRM signals together.

4.4 Transmission modes

4.4.1 Signal bandwidth related parameters

The current channel widths for radio broadcasting below 30 MHz are 9 kHz and 10 kHz. The DRM system is designed to be used:

- within these nominal bandwidths, in order to satisfy the current planning situation;
- within half these bandwidths (4,5 kHz or 5 kHz) in order to allow for simulcast with analogue AM signals;
- within twice these bandwidths (18 kHz or 20 kHz) to provide for larger transmission capacity where and when the planning constraints allow for such facility.

These signal bandwidth related parameters are specified in clause 8.

4.4.2 Transmission efficiency related parameters

For any value of the signal bandwidth parameter, transmission efficiency related parameters are defined to allow a trade off between capacity (useful bit rate) and ruggedness to noise, multipath and Doppler. These parameters are of two types:

- coding rate and constellation parameters, defining which code rate and constellations are used to convey data;
- OFDM symbol parameters, defining the structure of the OFDM symbols to be used as a function of the propagation conditions.

4.4.2.1 Coding rates and constellations

As a function of the desired protection associated within each service or part of a service, the system provides a range of options to achieve one or two levels of protection at a time. Depending on service requirements, these levels of protection may be determined by either the code rate of the channel encoder (e.g. 0,6, etc.), by the constellation order (e.g. 4-QAM, 16-QAM, 64-QAM), or by hierarchical modulation. Detailed definition of these options is given in clause 7.

4.4.2.2 OFDM parameter set

The OFDM parameter set is presented in this clause. The specification of the signal waveform is given in clause 8. These values are defined for different propagation-related transmission conditions to provide various robustness modes for the signal. In a given bandwidth, the different robustness modes provide different available data rates. Table 1 illustrates typical uses of the robustness modes.

Table 1: Robustness mode uses

Robustness mode	Typical propagation conditions
A	Gaussian channels, with minor fading
B	Time and frequency selective channels, with longer delay spread
C	As robustness mode B, but with higher Doppler spread
D	As robustness mode B, but with severe delay and Doppler spread

The transmitted signal comprises a succession of OFDM symbols, each symbol being made of a guard interval followed by the so-called useful part of the symbol. Each symbol is the sum of K sine wave portions equally spaced in frequency. Each sine wave portion, called a "cell", is transmitted with a given amplitude and phase and corresponds to a carrier position. Each carrier is referenced by the index k , k belonging to the interval $[k_{\min}, k_{\max}]$ ($k = 0$ corresponds to the reference frequency of the transmitted signal).

The time-related OFDM symbol parameters are expressed in multiples of the elementary time period T , which is equal to $83^{1/3}$ μs . These parameters are:

- T_g : duration of the guard interval;
- T_s : duration of an OFDM symbol;
- T_u : duration of the useful (orthogonal) part of an OFDM symbol (i.e. excluding the guard interval).

The OFDM symbols are grouped to form transmission frames of duration T_f .

As specified in clause 8, a certain number of cells in each OFDM symbol are transmitted with a predetermined amplitude and phase, in order to be used as references in the demodulation process. They are called "reference pilots" and represent a certain proportion of the total number of cells.

Table 2: OFDM symbol parameters

Parameters list	Robustness mode			
	A	B	C	D
T (μs)	$83^{1/3}$	$83^{1/3}$	$83^{1/3}$	$83^{1/3}$
T_u (ms)	24 ($288 \times T$)	$21^{1/3}$ ($256 \times T$)	$14^{2/3}$ ($176 \times T$)	$9^{1/3}$ ($112 \times T$)
T_g (ms)	$2^{2/3}$ ($32 \times T$)	$5^{1/3}$ ($64 \times T$)	$5^{1/3}$ ($64 \times T$)	$7^{1/3}$ ($88 \times T$)
T_g/T_u	1/9	1/4	4/11	11/14
$T_s = T_u + T_g$ (ms)	$26^{2/3}$	$26^{2/3}$	20	$16^{2/3}$
T_f (ms)	400	400	400	400

5 Source coding modes

5.1 Overview

The source coding options in the DRM system are shown in figure 2.

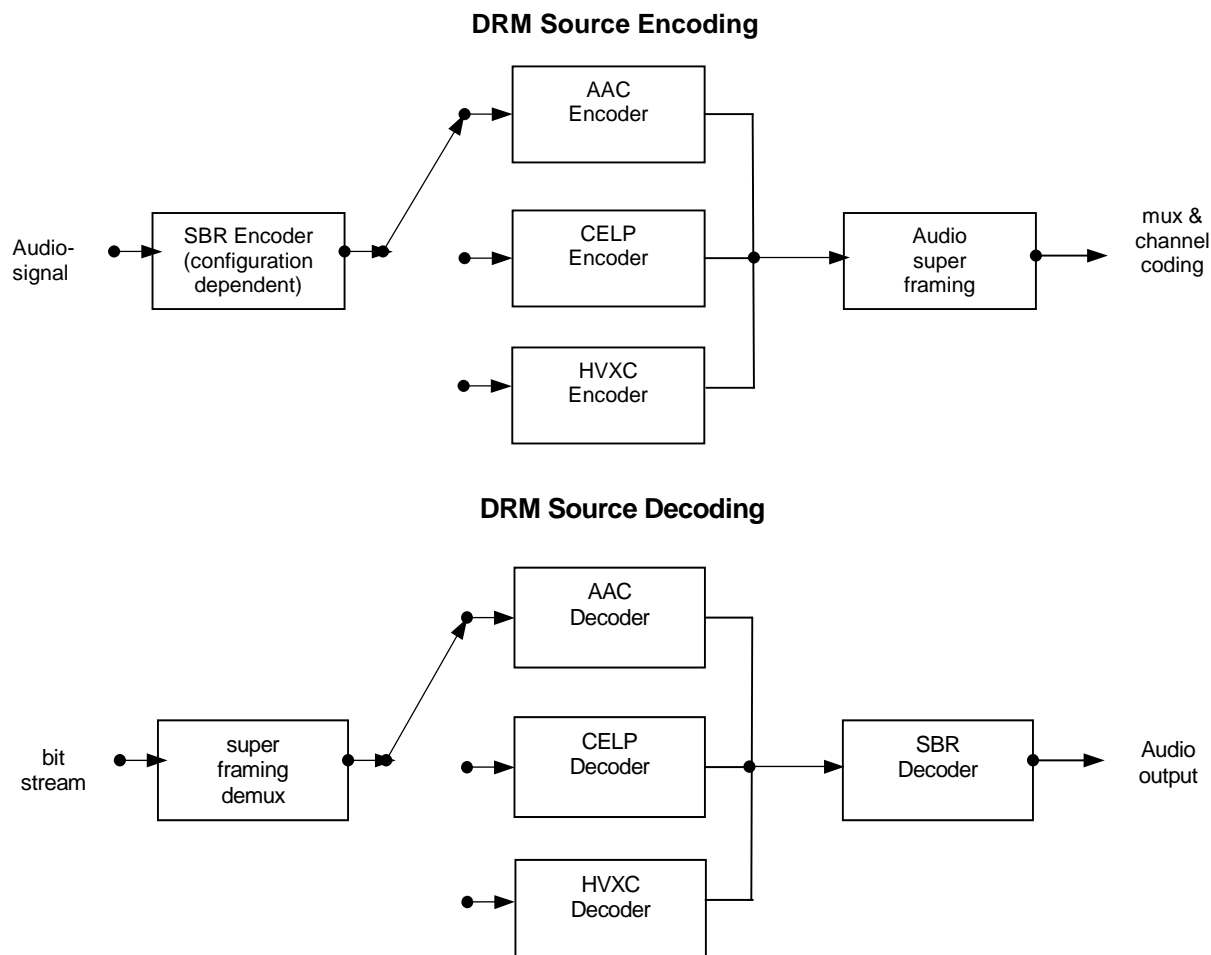


Figure 2: Source coding overview

As described in clause 4.3, the DRM system offers audio coding (AAC) and speech coding (CELP and HVXC). In addition, a high frequency reconstruction method (SBR) can be used to enhance the perceptual audio quality of the three different source coding schemes. Special care is taken so that the encoded audio can be composed into audio super frames of constant length. Multiplexing and UEP of audio/speech services is done by means of the multiplex and channel coding units. Audio specific configuration information is transmitted in the SDC (see clause 6.4.3.10).

5.1.1 AAC audio coding

For generic audio coding, a subset of the MPEG-4 Advanced Audio Coding (AAC) toolbox chosen to best suit the DRM system environment is used. For example a standard configuration for use in one short wave channel could be 20 kbit/s mono AAC.

Specific features of the AAC stream within the DRM system are:

- Bit rate: AAC can be used at any bit rate. Byte-alignment of the 400 ms audio super frame leads to a granularity of 20 bit/s for the AAC bit rate.
- Sampling rates: permitted sampling rates are 12 kHz and 24 kHz.

- Transform length: the transform length is 960 to ensure that one audio frame corresponds to 80 ms or 40 ms in time. This is required to harmonize CELP and AAC frame lengths and thus to allow the combination of an integer number of audio frames to build an audio super frame of 400 ms duration.
- Error robustness: a subset of MPEG-4 tools is used to improve the AAC bit stream error robustness in error prone channels.
- Audio super framing: 5 (12 kHz) or 10 (24 kHz) audio frames are composed into one audio super frame, which always corresponds to 400 ms in time. The audio frames in the audio super frames are encoded together such that each audio super frame is of constant length, i.e. that bit exchange between audio frames is only possible within an audio super frame. One audio super frame is always placed in one logical frame (see clause 6). In this way no additional synchronization is needed for the audio coding. Retrieval of frame boundaries and provisions for UEP are also taken care of within the audio super frame.
- UEP: better graceful degradation and better operation at higher BERs is achieved by applying UEP to the AAC bit stream. Unequal error protection is realized via the multiplex/coding units.

5.1.2 MPEG CELP coding

MPEG CELP speech coding is offered in the DRM system to allow for reasonable speech quality at bit rates significantly below the standard rate (for example "half rate" operation at 8 kbit/s). Possible scenarios for the use of the speech coder are:

- Dual/triple speech applications: instead of one audio program at 20 kbit/s to 24 kbit/s, the channel contains two or three speech signals of 8 kbit/s to 10 kbit/s each, allowing simultaneous speech transmissions.
- Speech services in addition to an audio service.
- Simulcast transmissions: in case of analogue/digital simulcast only bit rates as low as 8 kbit/s may be available.
- Very robust speech applications: due to its nature a speech coder can be expected to offer higher robustness against channel errors. Therefore 8 kbit/s speech coding can be used to do ultra robust speech coding in one channel.

Basic features of MPEG CELP coding are:

- 8 kHz or 16 kHz sampling rate;
- Bit rates between 4 kbit/s and 20 kbit/s;
- error robustness;
- composition of an integer number of CELP frames to build one audio super frame.

5.1.3 MPEG HVXC coding

MPEG-4 HVXC (Harmonic Vector eXcitation Coding) speech coding is offered in the DRM system to allow for reasonable speech quality at very low bit rates such as 2,0 kbit/s. The operating bit rates of HVXC open up new applications for DRM such as:

- Speech services in addition to an audio service.
- Multi-language application.
- Solid-state storage of multiple programs such as news, data base in a card radio (e.g. total of about 4,5 hours of radio programs can be stored in 4 MByte Flash memory).
- Time scale modification for fast playback/browsing of stored program.
- Highly error robust transmission with or without hierarchical modulation scheme.

Basic features of HVXC coding are:

- 8 kHz sampling rate;
- Bit rates of 2,0 kbit/s and 4,0 kbit/s for fixed rate coding;
- Time scale and pitch modification of arbitrary amounts;
- error robust syntax is supported, and a CRC tool can be used to improve the error resilience of the HVXC bitstream in error prone channels;
- composition of a constant integer number of HVXC frames (20) to build one audio super frame.

5.1.4 SBR coding

To maintain a reasonable perceived audio quality at low bit rates, classical audio or speech source coding algorithms need to limit the audio bandwidth and to operate at low sampling rates. It is desirable to be able to offer high audio bandwidth also in very low bit rate environments. This can be realized by the use of Spectral Band Replication (SBR).

The purpose of SBR is to recreate the missing high frequency band of the audio signal that could not be coded by the encoder. In order to do this in the best possible way, some side information needs to be transmitted in the audio bitstream, removing a small percentage of the available data rate from the audio coder. This side information is computed on the full bandwidth signal, prior to encoding and aids the reconstruction of the high frequencies after audio/speech decoding.

SBR exists in two versions. The version difference is only reflected in the decoder design. High Quality SBR uses a complex filterbank whereas Low Power SBR uses a real-valued filterbank plus anti-aliasing modules. The Low Power version of SBR offers a significant reduction in complexity as compared to the High Quality version without compromising too much on audio quality. AAC + SBR is defined in MPEG-4 Audio (High Efficiency AAC profile). SBR is also used in the configurations HVXC + SBR and CELP + SBR.

5.1.5 PS coding

For improved performance at low bitrate stereo coding, a Parametric Stereo (PS) coder partly based upon the SBR framework is available. The PS tool can be used when running the configuration AAC + SBR (MPEG High Efficiency AAC profile). In the source coding signal chain depicted in figure 2, the encoding and decoding of PS are performed within the SBR encoding and decoding blocks.

The general idea with PS coding is to send stereo image describing data as side information along with a downmixed mono signal. This stereo side information is very concise and only requires a small fraction of the total bitrate allowing the mono signal to have maximum quality for the total bitrate given.

The stereo synthesis at the decoder reconstructs spatial properties but does not affect the total spectral energy. Hence, there is no colorization of the frequency spectrum compared to the mono compatible core signal. The target bitrates for applying parametric stereo coding on AAC + SBR are preferably any bitrate range where traditional stereo cannot be afforded.

5.2 UEP and audio super framing

Today's coding schemes are highly optimized in terms of coding efficiency and according to information theory this should lead to the fact, that the entropy of the bits is nearly equal. If this assumption is true, then the channel coding must be optimized, such that the total amount of residual errors usually referred to as Bit Error Rate (BER) is minimized. This criterion can be fulfilled by a channel coding method called Equal Error Protection (EEP), were all information bits are protected with the same amount of redundancy.

However, the audible effects of errors are not independent of the part of the bitstream that was hit by the error. This behaviour of unequal error sensitivity is well known for source coding schemes that are used in broadcast and communication systems, like DAB (Eureka 147) or GSM. The optimized solution to cope with this unequal error sensitivity is called Unequal Error Protection (UEP). In such a system, higher protection is assigned to the more sensitive information, whereas lower protection is assigned to the less sensitive part of the bitstream.

To accommodate for UEP channel coding, it is necessary to have frames with a constant length and a UEP profile that is constant as well for a given bit rate. Since AAC is a coding scheme with a variable length, several coded frames are grouped together to build one audio super frame. The bit rate of the audio super frame is constant. Since the channel coding is based on audio super frames, the audio super frames themselves consist of two parts: a higher protected part and a lower protected part. Therefore, the coded audio frames itself have to be split into these two part. Further details on the audio super frame structure of AAC, CELP and HVXC are provided in the subsequent clauses. Note that HVXC is intended for use with the EEP scheme only.

Table 3: Syntax of audio_super_frame()

Syntax	No. of bits	Note
<pre>audio_super_frame(audio_info) //audio info from the SDC { switch (audio_info.audio_coding) { case AAC: aac_super_frame(audio_info) break; case CELP: celp_super_frame(audio_info) break; case HVXC: hvxc_super_frame(audio_info) break; } }</pre>		
<p>NOTE: The SDC describes the audio coder used, and the parameters associated with that coder. It also provides information about the sampling rate and bit rate used (see clause 6).</p>		

5.3 AAC coding

The following two clauses explain how the AAC and AAC + SBR frames fit into the audio super frame.

5.3.1 AAC

ISO/IEC 14496-3 [2] defines the MPEG-4 Audio standard. The audio coding standard MPEG-4 AAC is part of the MPEG-4 Audio standard. From the possible audio object types, only the Error Robust (ER) AAC Scalableobject type (Object Type ID = 20), which is part of the High Quality Audio Profile, is used in the DRM system.

DRM specific usage of MPEG-4 AAC: Three error robustness tools may be used within an MPEG-4 ER AAC bitstream: HCR (Huffman Codeword Reordering), VCB11 (Virtual Codebooks for Codebook 11) and RVLC (Reversible Variable Length Coding). In the DRM system, all AAC bitstreams shall use the HCR tool, since this tool reduces the error sensitivity of the bitstream significantly with a minimum of overhead. The VCB11 tool shall be used, since for low bit rates, the VCB11 overhead is less than 1 %. The RVLC tool is not used, since it introduces a significant bit rate overhead that is a major drawback for the low bit rates used by DRM.

The MPEG-4 AAC tool PNS (Perceptual Noise Substitution) is not used in DRM since SBR provides this functionality more appropriately.

For DRM the 960 transform shall be used.

When 12 kHz sampling is used, 5 AAC frames shall be combined into one audio super frame.

When 24 kHz sampling is used, 10 AAC frames shall be combined into one audio super frame.

The AAC sampling rate shall be 24 kHz when the stereo mode is used.

No standard extension_payload() shall be used and the only allowed extension is SBR (signalled via SDC).

The left and the right channel in one stereo audio frame are transmitted in an interleaved way to achieve a decreasing error sensitivity within the stereo frame.

Any DRM AAC bitstream can easily be translated into an MPEG-4 ER compliant bitstream by applying the above rules.

When the transmission is a base layer (the Base/Enhancement flag in the FAC is 0, see clause 6.3.3), the AAC frame corresponds to `aac_scalable_main_element()` as defined in the MPEG-4 standard [2].

The MPEG-4 standard defines how the bits for one raw error robust AAC audio frame are stored. Each element of the error robust AAC bitstream is assigned an error sensitivity category. In the DRM system there are two possible error robust AAC audio frames:

mono audio frame

One mono audio frame consists of three consecutive parts, hereinafter called `mono1`, `mono2` and `mono3`. `mono1` contains the Side Information (SI) bits, `mono2` contains the Temporal Noise Shaping (TNS) bits and `mono3` contains the spectral data bits. The error sensitivity decreases from `mono1` to `mono3`.

stereo audio frame

One stereo audio frame consists of seven consecutive parts, hereinafter called `stereo1` (common side info), `stereo2` (side info left channel), `stereo3` (side info right channel), `stereo4` (TNS left channel), `stereo5` (TNS right channel), `stereo6` (spectral data left channel), `stereo7` (spectral data right channel). With this interleaving of left and right channel, the error sensitivity is decreasing from `stereo1` to `stereo7`.

5.3.1.1 AAC audio super frame

Table 4: Syntax of `aac_super_frame()`

Syntax	No. of bits	Note
<pre> aac_super_frame(audio_info) //audio info from the SDC { switch (audio_info.audio_sampling_rate) { //only 12 000 and 24 000 is allowed case 12 000: num_frames = 5; break; case 24 000: num_frames = 10; break; } aac_super_frame_header(num_frames - 1) for (f = 0; f < num_frames; f++) { // higher_protected_block for (b = 0; b < num_higher_protected_bytes; b++) audio_frame[f][b] aac_crc_bits[f] } //lower_protected_part for (f = 0; f < num_frames; f++) { num_lower_protected_bytes = frame_length[f] - num_higher_protected_bytes; for (b = 0; b < num_lower_protected_bytes; b++) audio_frame[f][num_higher_protected_bytes + b] } } </pre>	<p>8</p> <p>8</p> <p>8</p>	<p>see annex D</p>
NOTE 1: <code>num_higher_protected_bytes</code> is derived from the UEP profile used (see clause 6).		
NOTE 2: <code>audio_frame</code> is either an AAC or an AAC + SBR frame.		

Table 5: Syntax of aac_super_frame_header()

Syntax	No. of bits	Note
<pre> aac_super_frame_header(num_borders) { previous_border = 0; for (n = 0; n < num_borders; n++) { frame_length[n] = frame_border - previous_border; // frame border in bytes previous_border = frame_border; } frame_length[num_borders] = audio_payload_length - previous_border; if (num_borders == 9) reserved // byte-alignment } </pre>	<p>12</p> <p>4</p>	
<p>NOTE: The audio_payload_length is derived from the length of the audio super frame (data_length_of_part_A + data_length_of_part_B) subtracting the audio super frame overhead (bytes used for the audio super frame header() and for the aac_crc_bits).</p>		

higher protected part

The higher protected part contains one header followed by num_frames higher protected blocks. num_frames is the number of audio frames in the audio super frame.

header

The header contains information to recover the frame lengths of the num_frames AAC frames stored in the audio super frame.

All the frame lengths are derived from the absolute positions of the frame borders. These frame borders are stored consecutively in the header. Each frame border occupies 12 bits (unsigned integer, most significant bit first). The frame border is measured in bytes from the start of the AAC bitstream sequence. 4 padding bits are added in case num_frames==10. num_frames-1 frame borders are stored in the header.

higher protected block

One higher protected block contains a certain amount of bytes from the start of each AAC frame, dependent upon the UEP profile. One 8-bit CRC check derived from the CRC-bits of the corresponding AAC frame follows (see annex D for CRC calculation). For a mono signal, the CRC-bits cover (mono1, mono2). For a stereo signal, the CRC-bits cover (stereo1, stereo2, stereo3, stereo4, stereo5).

lower protected part

The lower protected bytes (the remaining bytes not stored in the higher protected part) of the AAC frames are stored consecutively in the lower protected part.

Figure 3 illustrates an example audio super frame for a 24 kHz sampled signal.

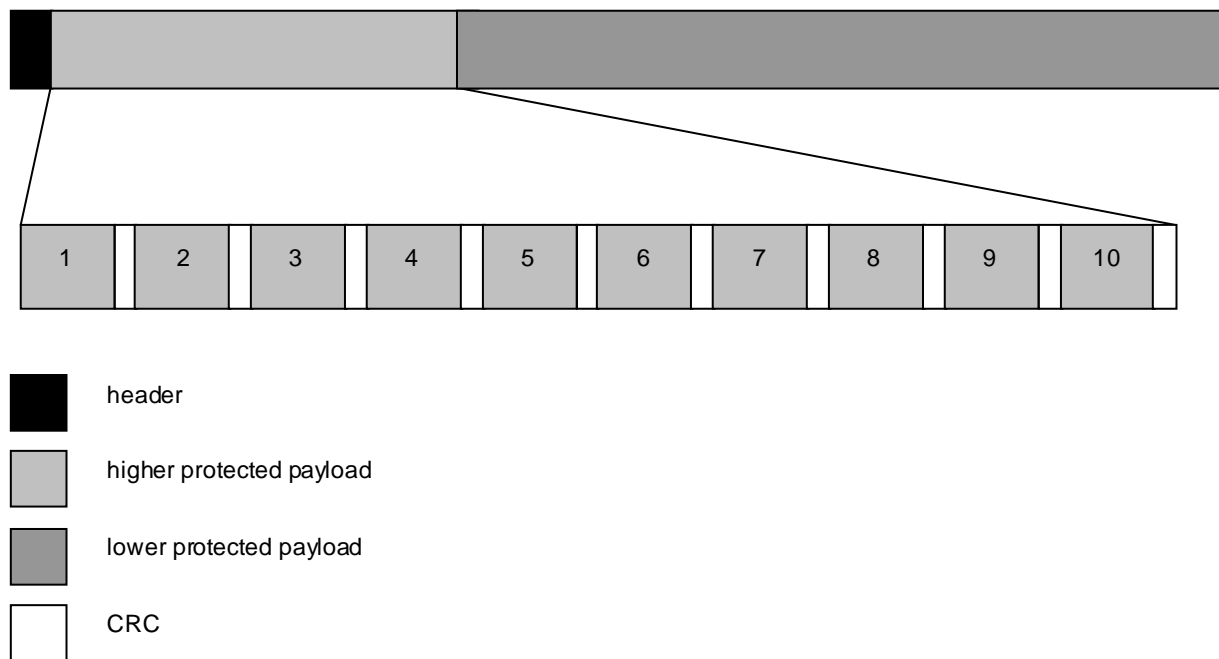


Figure 3: Example AAC audio super frame (24 kHz)

5.3.2 AAC + SBR

The SBR sampling rate is twice the AAC sampling rate. One raw AAC + SBR frame contains an AAC part and a SBR part. The SBR part of the data is located at the end of the frame. The first bit in the SBR-bitstream is the last bit in the frame, and the SBR bits are thus written/read in reversed order. In this way, the starting points of respective part of the frame data are always easily found.

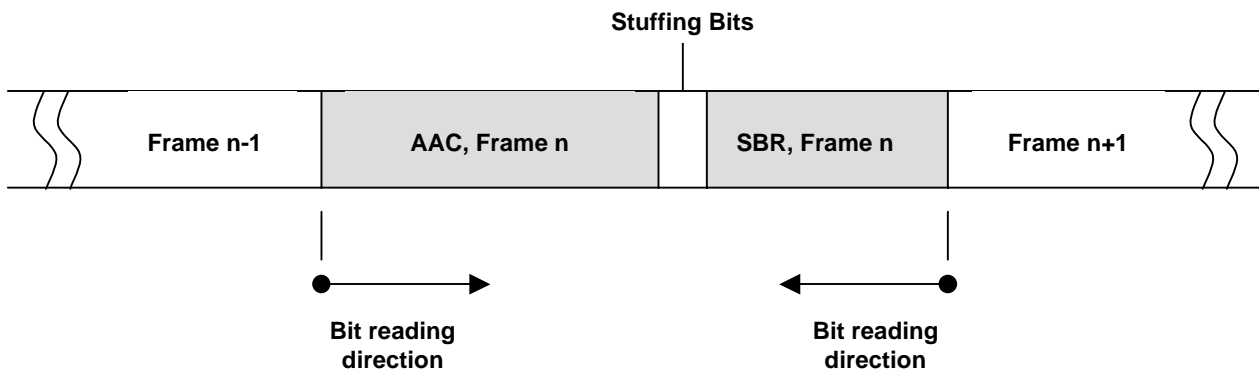


Figure 4: AAC + SBR frame

Both AAC and SBR data-sizes vary from frame to frame. The total size of the individual frames, now including the SBR data, can be derived from the `aac_super_frame_header()` as described in clause 5.3.1. Thus no extra signalling due to the varying SBR bit rate is needed.

The AAC + SBR frames are inserted into the audio super frame in the same manner as when SBR is not used.

For source coding bit rates at 20 kbit/s or greater, SBR shall be used. For bit rates below 20 kbit/s, SBR may be used.

The details of the SBR-bitstream are described in clause 5.6.1.

5.4 MPEG CELP coding

5.4.1 MPEG CELP

ISO/IEC 14496-3 [2] defines the MPEG-4 Audio standard. The speech coding standard MPEG-4 CELP (Code Excited Linear Prediction) is part of the MPEG-4 Audio standard. Two versions are defined, but only version 2 is intended for the use in error prone channels. The CELP bitstreams in the DRM system are therefore MPEG-4 version 2 bitstreams. From the possible audio object types, only the Error Robust (ER) CELP object type (Object Type ID = 24), which is part of the High Quality Audio Profile, is used in the DRM system.

The MPEG-4 CELP covers the compression and decoding of natural speech sound at bit rates ranging between 4 kbit/s and 24 kbit/s. MPEG-4 CELP is a well-known coding algorithm with new functionality. Conventional CELP coders offer compression at a single bit rate and are optimized for specific applications. Compression is one of the functionalities provided by MPEG-4 CELP, but MPEG-4 also enables the use of one basic coder in multiple applications. It provides scalability in bit rate and bandwidth, as well as the ability to generate bitstreams at arbitrary bit rates. The MPEG-4 CELP coder supports two sampling rates, namely, 8 kHz and 16 kHz. The associated bandwidths are 100 Hz to 3 800 Hz for 8 kHz sampling and 50 Hz to 7 000 Hz for 16 kHz sampling.

A basic block diagram of the CELP decoder is given in figure 5.

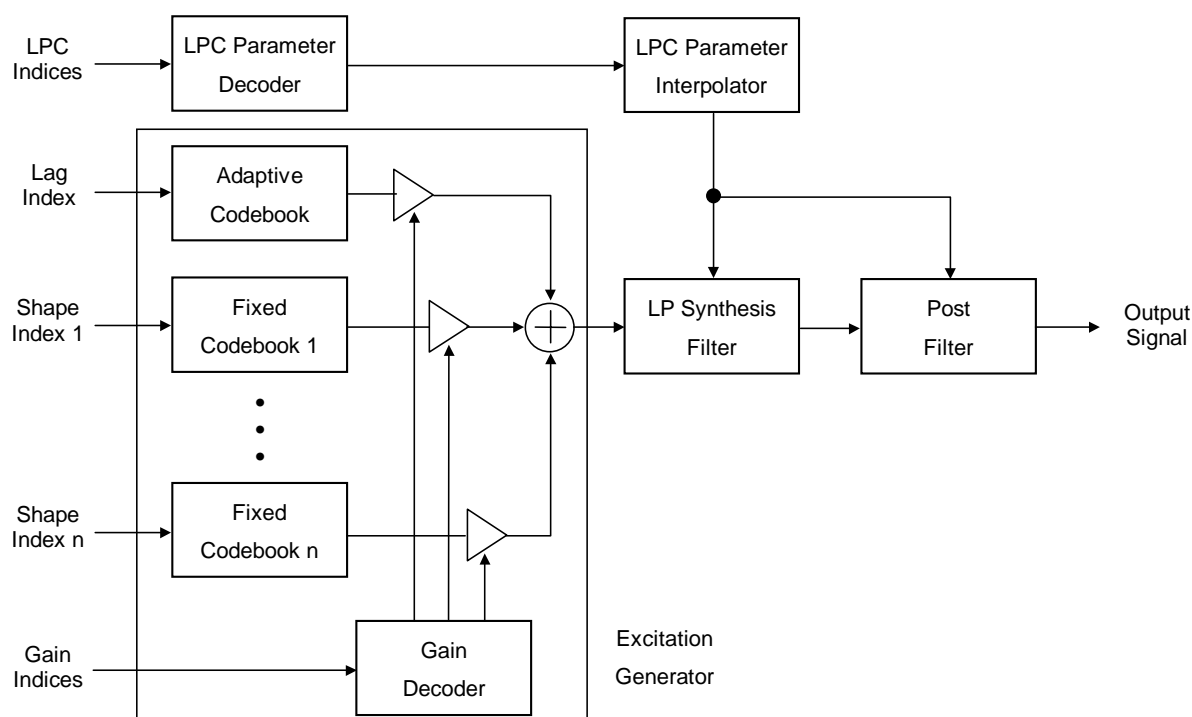


Figure 5: Block diagram of a CELP decoder

The CELP decoder primarily consists of an excitation generator and a synthesis filter. Additionally, CELP decoders often include a post-filter. The excitation generator has an adaptive codebook to model periodic components, fixed codebooks to model random components and a gain decoder to represent a speech signal level. Indices for the codebooks and gains are provided by the encoder. The codebook indices (pitch-lag index for the adaptive codebook and shape index for the fixed codebook) and gain indices (adaptive and fixed codebook gains) are used to generate the excitation signal. The excitation signal is then filtered by the linear predictive synthesis filter (LP synthesis filter). Filter coefficients are reconstructed using the LPC indices, then are interpolated with the filter coefficients of successive analysis frames. Finally, a post-filter can optionally be applied in order to enhance the speech quality.

The MPEG-4 CELP coder offers the following functionalities: Multiple bit rates, Bit rate Scalability, Bandwidth Scalability, and Fine Rate Control. DRM only uses the multiple bit rates functionality.

Multiple bit rates: the available bit rates depend on the sampling rate. The following fixed bit rates can be used.

Table 6: Fixed bit rates for the CELP coder

Bit rates for the 8 kHz sampling rate (bit/s)	Bit rates for the 16 kHz sampling rate (bit/s)
3 850, 4 250, 4 650, 5 700, 6 000, 6 300, 6 600, 6 900, 7 100, 7 300, 7 700, 8 300, 8 700, 9 100, 9 500, 9 900, 10 300, 10 500, 10 700, 11 000, 11 400, 11 800, 12 000, 12 200	10 900, 11 500, 12 100, 12 700, 13 300, 13 900, 14 300, 14 700, 15 900, 17 100, 17 900, 18 700, 19 500, 20 300, 21 100, 13 600, 14 200, 14 800, 15 400, 16 000, 16 600, 17 000, 17 400, 18 600, 19 800, 20 600, 21 400, 22 200, 23 000, 23 800

The algorithmic delay of the CELP coder comes from the frame length and an additional look ahead length. The frame length depends on the coding mode and the bit rate. The look ahead length, which is an informative parameter, also depends on the coding mode. The delays presented below are applicable to the modes used in DRM.

Table 7: Delay and frame length for the CELP coder at 8 kHz sampling rate

Bit rate (bit/s)	Delay (ms)	Frame length (ms)
3 850, 4 250, 4 650	45	40
5 700, 6 000, 6 300, 6 600, 6 900, 7 100, 7 300, 7 700, 8 300, 8 700, 9 100, 9 500, 9 900, 10 300, 10 500, 10 700	25	20
11 000, 11 400, 11 800, 12 000, 12 200	15	10

Table 8: Delay and frame length for the CELP coder at 16 kHz sampling rate

Bit rate (bit/s)	Delay (ms)	Frame length (ms)
10 900, 11 500, 12 100, 12 700, 13 300, 13 900, 14 300, 14 700, 15 900, 17 100, 17 900, 18 700, 19 500, 20 300, 21 100	25	20
13 600, 14 200, 14 800, 15 400, 16 000, 16 600, 17 000, 17 400, 18 600, 19 800, 20 600, 21 400, 22 200, 23 000, 23 800	15	10

5.4.1.1 CELP audio super frame

CELP frames have a fixed frame length. The CELP audio frames are grouped together to form audio super frames of 400 ms duration. UEP is applicable. The start of each audio frame is mapped into the higher protected part, the remaining bits are allocated to the lower protected part. The partitioning of the CELP frames is given in tables 10 and 11. The CELP bit rate index is signalled in the SDC.

Table 9: Syntax of celp_super_frame()

Syntax	No. of bits	Note
<pre> celp_super_frame(celp_table_ind) // CELP table index from the SDC { switch (audio_info.audio_sampling_rate) { //only 8 000 and 16 000 is allowed case 8 000: (num_frames, num_higher_protected_bits, num_lower_protected_bits) = read_table_10 (CELP_index) break; case 16 000: (num_frames, num_higher_protected_bits, num_lower_protected_bits) = read_table_11 (CELP_index) break; } for (f = 0; f < num_frames; f++) { // higher_protected_block for (b = 0; b < num_higher_protected_bits; b++) celp_frame[f][b] if (audio_info.CELP_CRC == 1) celp_crc_bits[f] } // lower_protected_part for (f = 0; f < num_frames; f++) { for (b = 0; b < num_lower_protected_bits; b++) celp_frame[f][num_higher_protected_bits + b] } } </pre>	<p>1</p> <p>8</p> <p>1</p>	see annex D

Table 10: UEP parameters for 8 kHz sampling CELP

CELP bit rate index	Bit rate (bits/s)	Audio frame length (ms)	Higher protected part (bits/audio frame)	Lower protected part (bits/audio frame)	Higher protected part (bytes/audio super frame)	Lower protected part (bytes/audio super frame)	Audio super frame length (bytes)
0	3 850	40	36	118	45	148	193 (see note)
1	4 250	40	36	134	45	168	213 (see note)
2	4 650	40	36	150	45	188	233 (see note)
6	5 700	20	24	90	60	225	285
7	6 000	20	24	96	60	240	300
8	6 300	20	24	102	60	255	315
9	6 600	20	24	108	60	270	330
10	6 900	20	24	114	60	285	345
11	7 100	20	24	118	60	295	355
12	7 300	20	24	122	60	305	365
13	7 700	20	36	118	90	295	385
14	8 300	20	36	130	90	325	415
15	8 700	20	36	138	90	345	435
16	9 100	20	36	146	90	365	455
17	9 500	20	36	154	90	385	475
18	9 900	20	36	162	90	405	495
19	10 300	20	36	170	90	425	515
20	10 500	20	36	174	90	435	525
21	10 700	20	36	178	90	445	535
22	11 000	10	24	86	120	430	550
23	11 400	10	24	90	120	450	570
24	11 800	10	24	94	120	470	590
25	12 000	10	24	96	120	480	600
26	12 200	10	24	98	120	490	610

NOTE: For these bit rates, the last four bits of the audio super frame are padded with 0 s.

Table 11: UEP parameters for 16 kHz sampling CELP

CELP bit rate index	Bit rate (bits/s)	Audio frame length (ms)	Higher protected part (bits/audio frame)	Lower protected part (bits/audio frame)	Higher protected part (bytes/audio super frame)	Lower protected part (bytes/audio super frame)	Audio super frame length (bytes)
0	10 900	20	64	154	160	385	545
1	11 500	20	64	166	160	415	575
2	12 100	20	64	178	160	445	605
3	12 700	20	64	190	160	475	635
4	13 300	20	64	202	160	505	665
5	13 900	20	64	214	160	535	695
6	14 300	20	64	222	160	555	715
8	14 700	20	92	202	230	505	735
9	15 900	20	92	226	230	565	795
10	17 100	20	92	250	230	625	855
11	17 900	20	92	266	230	665	895
12	18 700	20	92	282	230	705	935
13	19 500	20	92	298	230	745	975
14	20 300	20	92	314	230	785	1 015
15	21 100	20	92	330	230	825	1 055
16	13 600	10	50	86	250	430	680
17	14 200	10	50	92	250	460	710
18	14 800	10	50	98	250	490	740
19	15 400	10	50	104	250	520	770
20	16 000	10	50	110	250	550	800
21	16 600	10	50	116	250	580	830
22	17 000	10	50	120	250	600	850
24	17 400	10	64	110	320	550	870
25	18 600	10	64	122	320	610	930
26	19 800	10	64	134	320	670	990
27	20 600	10	64	142	320	710	1 030
28	21 400	10	64	150	320	750	1 070
29	22 200	10	64	158	320	790	1 110
30	23 000	10	64	166	320	830	1 150
31	23 800	10	64	174	320	870	1 190

5.4.2 CELP + SBR

Figure 6 outlines the audio super frame composition for CELP + SBR. The CELP core part is identical to the CELP audio super frame defined in clause 5.4.1.1 and includes the possible 4 padding bits. Regardless of the frame lengths of the CELP core (10 ms, 20 ms or 40 ms), SBR employs 40 ms frames.

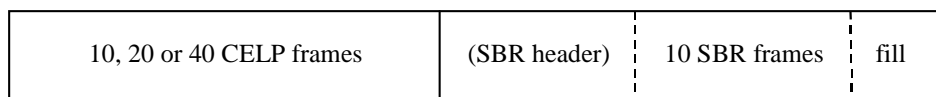


Figure 6: CELP + SBR audio super frame overview

The number of bits per SBR frame is constant and no signalling of frame locations within the superframe is used. After the last SBR frame, additional bits for byte alignment follow. If the *sbr_header_flag* in the SDC is set (see clause 6.4.3.10) the SBR frames are preceded by a single SBR header. In this mode, a wide range of SBR tunings, and correspondingly different overheads, are possible. If the flag is not set, no header is sent, and predefined combinations of tuning parameters are used as defined in clause I.1.

5.5 HVXC coding

The MPEG-4 HVXC (Harmonic Vector eXcitation Coding) speech coding toolset as defined in ISO/IEC 14496-3 [2] covers the compression and decoding of natural speech sound at bit rates of 2,0 kbit/s and 4,0 kbit/s. HVXC employs harmonic coding of LPC residual signals for voiced segments and Vector eXcitation Coding (VXC) for unvoiced segments. HVXC provides communications-quality to near-toll-quality speech in the 100 Hz to 3 800 Hz band at 8 kHz sampling rate. In addition, the functionality of pitch and speed change during decoding is supported. This functionality is useful for fast speech database search or browsing. HVXC has a syntax providing error sensitivity categories that can be used with an error robustness tool. Additionally the error concealment functionality is supported for the use in error-prone channels.

DRM uses a subset of the HVXC description in ISO/IEC 14496-3 [2], which limits the syntax to the error robust syntax and the data rates to the two options of 2,0 kbit/s and 4,0 kbit/s. Further, HVXC is used with the non-scalable syntax only. For robust decoding in error-prone channels a low-complexity error concealment tool (CRC and intra-frame interleaving) is defined specifically for DRM.

The syntax of the HVXC audio super frame is identical for all possible HVXC modes, since HVXC does not support UEP functionality and the length of a HVXC audio frame is always 20 ms.

Table 12: Syntax of hvxc_super_frame()

	Syntax	No. of bits	Note
hvxc_super_frame(audio_info)	// audio info from the SDC		
{			
num_frames = 20;			
}			

The number of bits contained in one audio frame is given by the audio information from the SDC (HVXC_rate, HVXC_CRC).

In case the 4,0 kbit/s fixed rate HVXC coder is used with the CRC tool, the last 4 bits of each audio super frame are padded with zeros and the receiver shall ignore these bits. The resulting bit rate therefore is 4,66 kbit/s.

Only the fixed rate modes of HVXC (2,0 kbit/s or 4,0 kbit/s) are used in audio super frames. Variable rate modes may be applicable for use with packet mode applications in the future.

5.5.1 Definitions

5.5.1.1 HVXC source coder parameters

The definition of the basic data entities of a MPEG-4 compliant HVXC speech coding system is given in ISO/IEC 14496-3 [2] and is reproduced as table N.1.

5.5.1.2 CRC bits for fixed bit rate modes

Table 13 describes the various CRC bits added for error protection in the DRM system.

Table 13: CRC bits for fixed bit rate modes

Parameter	Description	length (bits)
CRC0_2k	CRC bits for ESC0 at 2 kbit/s	6
CRC1_2k	CRC bits for ESC1 at 2 kbit/s	1
CRC2_2k	CRC bits for ESC2 at 2 kbit/s	1
CRC0_4k	CRC bits for ESC0 at 4 kbit/s	6
CRC1_4k	CRC bits for ESC1 at 4 kbit/s	5
CRC2_4k	CRC bits for ESC2 at 4 kbit/s	1
CRC3_4k	CRC bits for ESC3 at 4 kbit/s	1

5.5.2 HVXC decoder

Figure 7 shows the overall structure of the HVXC decoder. The basic decoding process is composed of four steps; de-quantization of parameters, generation of excitation signals for voiced frames by sinusoidal synthesis (harmonic synthesis) and noise component addition, generation of excitation signals for unvoiced frames by codebook look-up, and LPC synthesis. To enhance the synthesized speech quality spectral post-filtering is used.

For voiced frames, a fixed dimension harmonic spectral vector, obtained by de-quantization of the spectral magnitude, is first converted to the one having the original dimension which varies frame by frame in accordance with the pitch value. This is done by the dimension converter in which a band-limited interpolator generates a set of spectral magnitude values at harmonic frequencies without changing the shape of the spectral envelope. Using the spectral magnitude values, a time domain excitation signal is then generated by the fast harmonic synthesis algorithm using an IFFT. In order to make the synthesized speech sound natural, a noise component is additionally used. A Gaussian noise spectral component, covering 2 kHz to 3,8 kHz, is coloured in accordance with the harmonic spectral magnitudes in the frequency domain, and its IDFT is added to voiced excitation signals in the time domain. The amount and bandwidth of this additive noise is controlled by the transmitted two-bit V/UV value, which is encoded based on the normalized maximum autocorrelation of the LPC residual signal. Noise added harmonic excitation signals for voiced segments are then fed into the LPC synthesis filter followed by the postfilter.

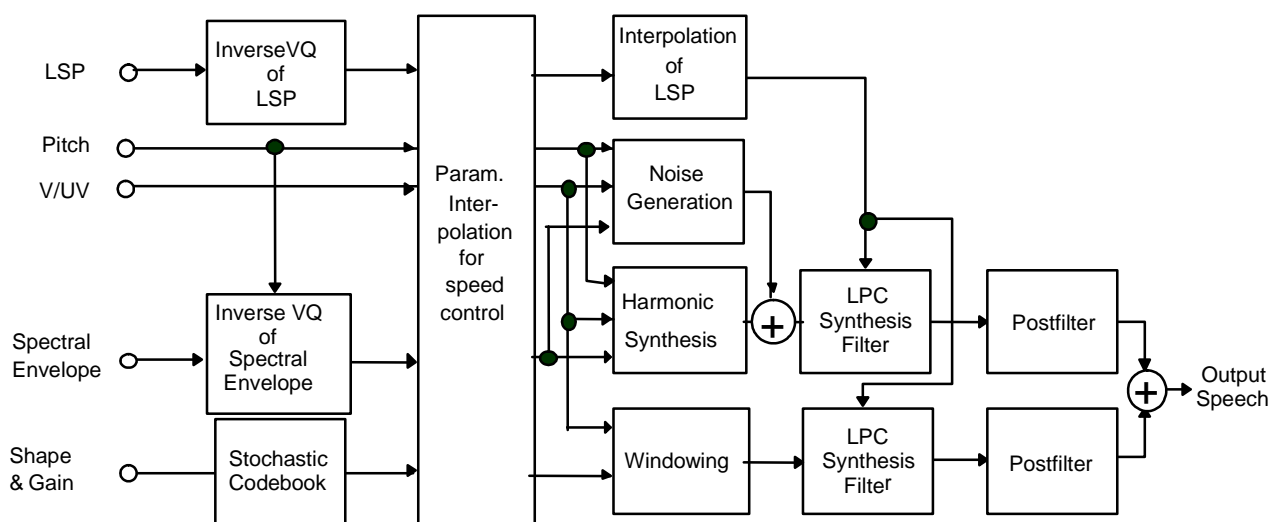


Figure 7: Block diagram of the HVXC decoder

For unvoiced segments, the usual VXC decoding algorithm is used where an excitation signal is generated by multiplying the gain value with the stochastic code vector. The result is then fed into the LPC synthesis filter followed by the postfilter. Finally, the synthesized speech components for voiced and unvoiced segments are added to form the output signal. The description of the time-scale modification standard can be found in the MPEG-4 [2].

5.5.3 HVXC encoder

Figure 8 shows the overall structure of the encoder of the MPEG-4 HVXC. Table N.2 shows the bit allocations for the 2,0 kbit/s and 4,0 kbit/s coders using fixed rate coding. The parameters followed by (enh) are used only for the 4,0 kbit/s mode. Operation of each part is described in figure 8.

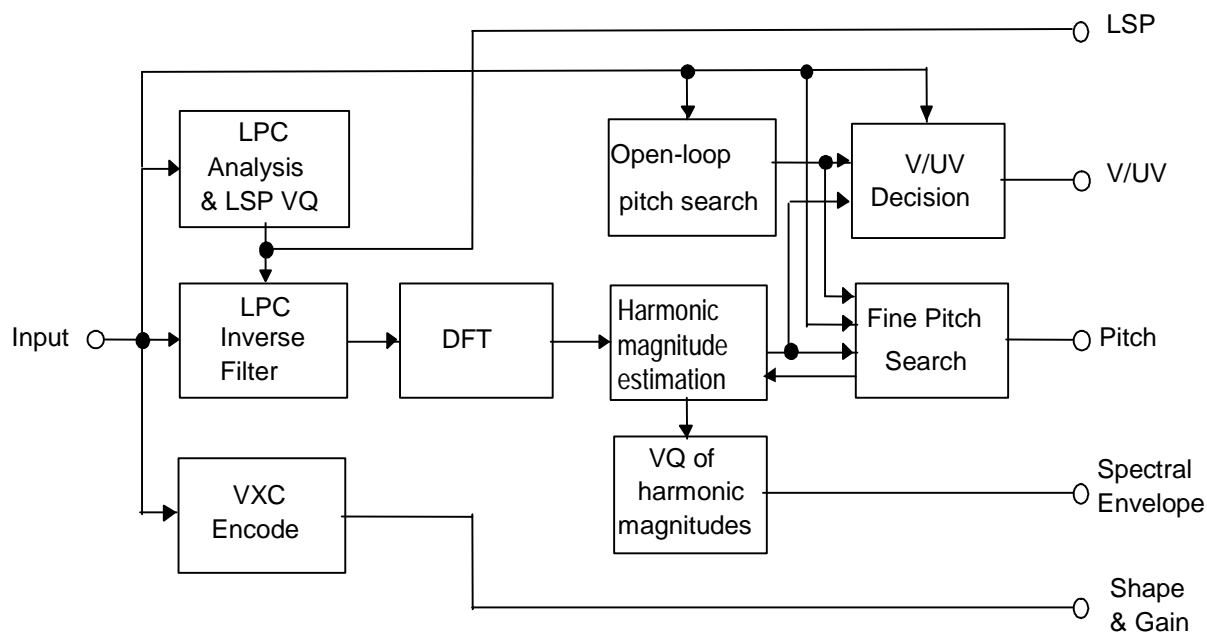


Figure 8: Block diagram of the HVXC encoder

5.5.3.1 LPC analysis and LSP quantization

The speech input at a sampling rate of 8 kHz is formed into frames with a length and interval of 256 samples and 160 samples, respectively. Tenth order LPC analysis is carried out using windowed input data over one frame. LPC parameters are converted to LSP parameters and vector quantized with a partial prediction and multi-stage vector quantization scheme. LPC residual signals are computed by inverse filtering the input data using quantized and interpolated LSP parameters.

5.5.3.2 Open loop pitch search

The open loop pitch value is estimated based on the peak values of the autocorrelation of the LPC residual signals. Using estimated past and current pitch values, pitch tracking is conducted to have a continuous pitch contour and to make the reliability of the pitch estimation higher. The voiced/unvoiced decision of the previous frame is also used to ensure the pitch tracking operation.

5.5.3.3 Harmonic magnitude and fine pitch estimation

The power spectrum of the LPC residual signal is then fed into the fine pitch and harmonic magnitude estimation block, where the harmonic spectral envelope of the LPC residual signal is estimated as follows: A basis spectrum representing one harmonic spectrum is gain scaled and arranged with the spacing of the fundamental frequency obtained by the open loop pitch search. The gain scaling for each harmonic and fundamental frequency is adjusted simultaneously so that the difference between the synthesized power spectrum and actual LPC residual spectrum is minimized. The harmonic spectral envelope for a voiced segment is then vector quantized.

5.5.3.4 Vector quantization of harmonic magnitudes

In order to vector quantize a spectral envelope composed of a variable number of harmonics, the harmonic spectral vector is first converted to a fixed-dimension vector. Band-limited interpolation by a polyphase filter bank is used for the dimensional conversion. A fixed-dimension spectral vector \mathbf{x} is then quantized with weighted distortion measure D :

$$D = \|\mathbf{WH}(\mathbf{x} - g(s_0 + s_1))\|^2$$

where \mathbf{s}_0 is the output of the first shape codebook, \mathbf{s}_1 is the output of the second shape codebook, and g is the output of the gain codebook. The diagonal components of the matrices \mathbf{H} and \mathbf{W} are the magnitudes of the frequency response of the LPC synthesis filter and the perceptual weighting filter, respectively. In order to reduce the memory requirements and search complexity while maintaining a high performance, two-stage vector quantization scheme is employed for the spectral shape together with a scalar quantizer for the gain under 2,0 kbit/s coding. For the 4,0 kbit/s mode, the quantized harmonic magnitude vector with fixed dimension is first converted to the dimension of original harmonics by the same band-limited interpolation mentioned above. The difference between the original harmonics and de-quantized and dimension converted harmonics are then quantized with additional vector quantizers. This multi-stage structure allows generation of scalable bit-streams.

5.5.3.5 Voiced/Unvoiced decision

The Voiced/Unvoiced decision is made based on the maximum autocorrelation of the LPC residual signals, the number of zero crossing and the harmonic structure of the power spectrum of the LPC residual signals.

5.5.3.6 VXC coding of unvoiced signals

For unvoiced segments, regular VXC coding is carried out, where only stochastic codebooks are used. A 6 bits shape codebook of dimension 80 and 4 bits gain codebook are used for the 2,0 kbit/s mode. For the 4,0 kbit/s mode, the quantization error of the 2,0 kbit/s mode is quantized using a 5 bits shape codebook of dimension 40 and a 3 bits gain codebook at the additional stage.

5.5.4 HVXC channel coding

5.5.4.1 Protected bit selection

According to the sensitivity of bits, encoded bits are classified to several Error Sensitivity Categories (ESC). The number of bits for each ESC is shown in table 14 (2,0 kbit/s, voiced sound), table 15 (2,0 kbit/s, unvoiced sound), table 16 (4,0 kbit/s, voiced sound) and table 17 (4,0 kbit/s, unvoiced sound). ESC0 is the group of the most error sensitive bits and ESC4 is the group of the least sensitive bits. Bit rate setting of total of 2,4 kbit/s using 2,0 kbit/s source coder rate, and total of 4,66 kbit/s using 4,0 kbit/s source coder rate are shown.

NOTE: The overall bit rate due to the usage of a CRC is 4,65 kbit/s, but additionally 4 padding bits have to be inserted for each audio super frame, resulting in an overall bit rate of 4,66 kbit/s, see clause 5.5.

Table 14: Number of ESC bits at 2,0 kbit/s fixed rate mode (voiced sound)

Parameters	Voiced frame				
	ESC0 (bits)	ESC1 (bits)	ESC2 (bits)	ESC3 (bits)	total (bits)
LSP1	5	-	-	-	5
LSP2	2	-	-	5	7
LSP3	1	-	-	4	5
LSP4	1	-	-	-	1
VUV	2	-	-	-	2
Pitch	6	-	-	1	7
SE_gain	5	-	-	-	5
SE_shape1	-	4	-	-	4
SE_shape2	-	-	4	-	4
total	22	4	4	10	40
CRC	6	1	1	-	8
total + CRC	28	5	5	10	48

Table 15: Number of ESC bits at 2,0 kbit/s fixed rate mode (unvoiced sound)

Parameters	Unvoiced frame				
	ESC0 (bits)	ESC1 (bits)	ESC2 (bits)	ESC3 (bits)	total (bits)
LSP1	5	-	-	-	5
LSP2	4	3	-	-	7
LSP3	2	1	2	-	5
LSP4	1	-	-	-	1
VUV	2	-	-	-	2
VX_gain1 [0]	4	-	-	-	4
VX_gain1 [1]	4	-	-	-	4
VX_shape1 [0]	-	-	2	4	6
VX_shape1 [1]	-	-	-	6	6
total	22	4	4	10	40
CRC	6	1	1	-	8
total + CRC	28	5	5	10	48

Table 16: Number of ESC bits at 4,0 kbit/s fixed rate mode (voiced sound)

Parameters	Voiced frame					
	ESC0 (bits)	ESC1 (bits)	ESC2 (bits)	ESC3 (bits)	ESC4 (bits)	total (bits)
LSP1	5	-	-	-	-	5
LSP2	4	-	-	-	3	7
LSP3	1	-	-	-	4	5
LSP4	1	-	-	-	-	1
LSP5	1	-	-	-	7	8
VUV	2	-	-	-	-	2
Pitch	6	-	-	-	1	7
SE_gain	5	-	-	-	-	5
SE_shape1	-	-	4	-	-	4
SE_shape2	-	-	-	4	-	4
SE_shape3	5	-	-	-	2	7
SE_shape4	1	9	-	-	-	10
SE_shape5	1	8	-	-	-	9
SE_shape6	1	5	-	-	-	6
Total	33	22	4	4	17	80
CRC	6	5	1	1	-	13
Total + CRC	39	27	5	5	17	93

Table 17: Number of ESC bits at 4,0 kbit/s fixed rate mode (unvoiced sound)

Parameters	Unvoiced frame					
	ESC0 (bits)	ESC1 (bits)	ESC2 (bits)	ESC3 (bits)	ESC4 (bits)	total (bits)
LSP1	5	-	-	-	-	5
LSP2	4	3	-	-	-	7
LSP3	1	4	-	-	-	5
LSP4	1	-	-	-	-	1
LSP5	1	7	-	-	-	8
VUV	2	-	-	-	-	2
VX_gain1 [0]	4	-	-	-	-	4
VX_gain1 [1]	4	-	-	-	-	4
VX_shape1 [0]	-	6	-	-	-	6
VX_shape1 [1]	-	1	4	1	-	6
VX_gain2 [0]	3	-	-	-	-	3
VX_gain2 [1]	3	-	-	-	-	3
VX_gain2 [2]	3	-	-	-	-	3
VX_gain2 [3]	2	1	-	-	-	3
VX_shape2 [0]	-	-	-	3	2	5
VX_shape2 [1]	-	-	-	-	5	5
VX_shape2 [2]	-	-	-	-	5	5
VX_shape2 [3]	-	-	-	-	5	5
total	33	22	4	4	17	80
CRC	6	5	1	1	-	13
total + CRC	39	27	5	5	17	93

5.5.4.2 Syntax of DRM HVXC error robustness (ErHVXCfixframe_CRC)

The bitstream syntax consists of several Error Sensitivity Categories (ESC). Some ESCs include source bits and CRC bits, where CRC bits are computed from source bits within the same ESC.

The HVXC_CRC field in the SDC is used to indicate whether the stream includes CRC parity bits or not (see clause 6.4.3.10). The bitstream syntax of the input of the CRC checker is given in figure 9.

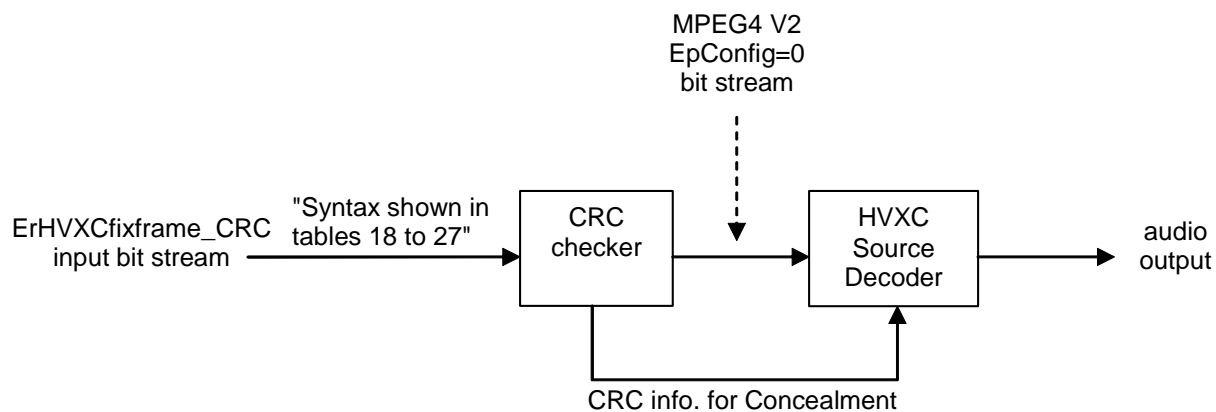


Figure 9: Block diagram of HVXC syntax

Table 18: Syntax of ErHVXCfixframe_CRC()

Syntax	No. of bits	Mnemonic
<pre>ErHVXCfixframe_CRC(rate) { if (rate == 2000) { 2k_ESC0_CRC0(); 2k_ESC1_CRC1(); 2k_ESC2_CRC2(); 2k_ESC3_NoCRC(); } else { 4k_ESC0_CRC0(); 4k_ESC1_CRC1(); 4k_ESC2_CRC2(); 4k_ESC3_CRC3(); 4k_ESC4_NoCRC(); } }</pre>		

Table 19: Syntax of 2k_ESC0_CRC0()

Syntax	No. of bits	Mnemonic
<pre>2k_ESC0_CRC0() { 2k_ESC0(); if (HVXC_CRC==1) { CRC0_2k, 5-0; } }</pre>	6	uimbsf
NOTE: 2k_ESC0() is defined in the MPEG-4 Audio standard [2].		

Table 20: Syntax of 2k_ESC1_CRC1()

Syntax	No. of bits	Mnemonic
<pre>2k_ESC1_CRC1() { 2k_ESC1(); if (HVXC_CRC==1) { CRC1_2k, 0; } }</pre>	1	uimbsf
NOTE: 2k_ESC1() is defined in the MPEG-4 Audio standard [2].		

Table 21: Syntax of 2k_ESC2_CRC2()

Syntax	No. of bits	Mnemonic
<pre>2k_ESC2_CRC2() { 2k_ESC2(); if (HVXC_CRC==1) { CRC2_2k, 0; } }</pre>	1	uimbsf
NOTE: 2k_ESC2() is defined in the MPEG-4 Audio standard [2].		

Table 22: Syntax of 2k_ESC3_NoCRC()

Syntax	No. of bits	Mnemonic
<pre>2k_ESC3_NoCRC() { 2k_ESC3(); }</pre>		
NOTE: 2k_ESC3() is defined in the MPEG-4 Audio standard [2].		

Table 23: Syntax of 4k_ESC0_CRC0()

Syntax	No. of bits	Mnemonic
<pre>4k_ESC0_CRC0() { 4k_ESC0(4000); if (HVXC_CRC==1) { CRC0_4k, 5-0; } }</pre>	6	uimsbf
NOTE: 4k_ESC0() is defined in the MPEG-4 Audio standard [2].		

Table 24: Syntax of 4k_ESC1_CRC1()

Syntax	No. of bits	Mnemonic
<pre>4k_ESC1_CRC1() { 4k_ESC1(4000); if (HVXC_CRC==1) { CRC1_4k, 4-0; } }</pre>	5	uimsbf
NOTE: 4k_ESC1() is defined in the MPEG-4 Audio standard [2].		

Table 25: Syntax of 4k_ESC2_CRC2()

Syntax	No. of bits	Mnemonic
<pre>4k_ESC2_CRC2() { 4k_ESC2(); if (HVXC_CRC==1) { CRC2_4k, 0; } }</pre>	1	uimsbf
NOTE: 4k_ESC2() is defined in the MPEG-4 Audio standard [2].		

Table 26: Syntax of 4k_ESC3_CRC3()

Syntax	No. of bits	Mnemonic
<pre>4k_ESC3_CRC3() { 4k_ESC3(); if (HVXC_CRC==1) { CRC3_4k, 0; } }</pre>	1	uimsbf
NOTE: 4k_ESC3() is defined in the MPEG-4 Audio standard [2].		

Table 27: Syntax of 4k_ESC4_NoCRC()

Syntax	No. of bits	Mnemonic
<pre>4k_ESC4_NoCRC() { 4k_ESC4(4000); }</pre>		
NOTE: 4k_ESC4() is defined in the MPEG-4 Audio standard [2].		

5.5.5 Category interleaving

In order to improve the robustness of the bitstream to channel errors, bit interleaving is carried out where data bits are reordered inside one frame. Bit interleaving is carried out after the bitstream is ordered and the CRC bits are added as shown in tables 18 to 27. The interleaving is done in two steps. First, the HVXC bitstream is divided into two sequences as:

- $x[0..N_1 - 1]$: sequence consists of bits of ESC0 where N_1 is the number of bits included in ESC0.
- $y[0..N_2 - 1]$: sequence consists of bits other than ESC0.

The sequence $y[]$ is composed of ESC1, ESC2 ..., in this order. N_1 is the number of bits of ESC0, and N_2 is the number of bits other than ESC0. The total number of bits is $N = N_1 + N_2$.

Next, the sequences $x[]$ and $y[]$ are interleaved into one sequence $z[0..N - 1]$. At first, the variables i_0 and i_1 are set to 0. A single bit from $x[]$ is inserted when i_1 is larger than or equal to i_0 . In this case, N is added to i_0 after bit insertion. Otherwise a single bit from $y[]$ is inserted. After a single bit insertion, N_1 is added to i_1 . The flowchart is shown in figure 10.

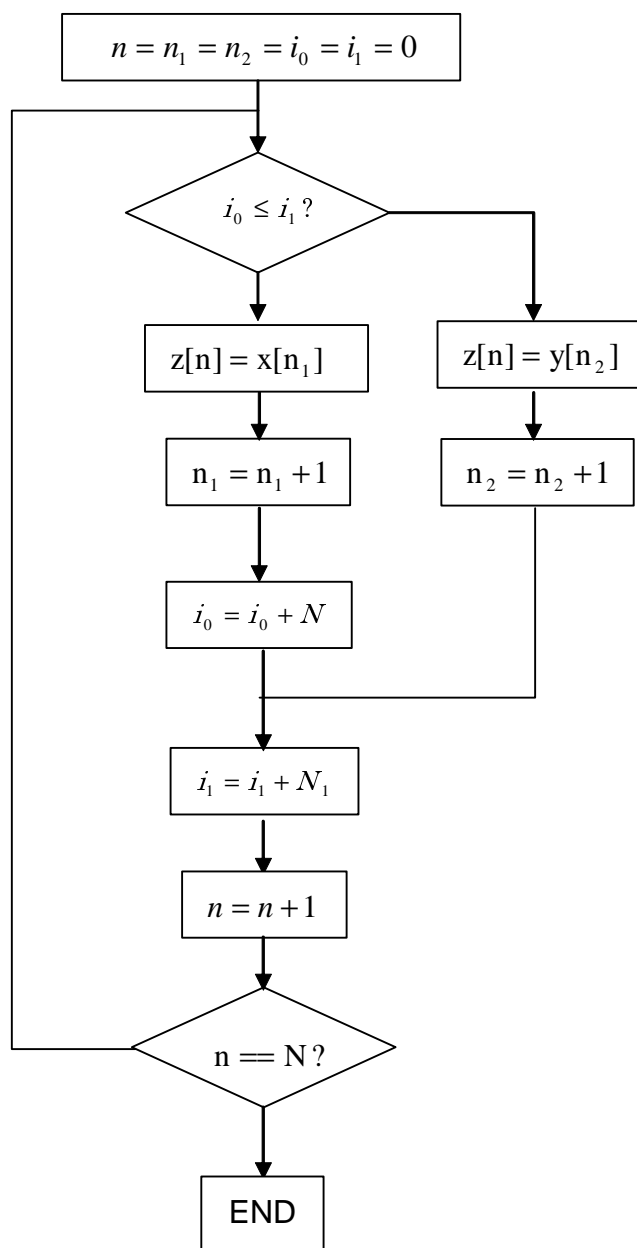


Figure 10: Flowchart of category interleaving method

5.5.6 HVXC error detection and concealment

5.5.6.1 Cyclic Redundancy Check (CRC)

The CRC parity bits are computed from the source bits in the same ESC. A schematic diagram of the CRC checker, the polynomials and initialization procedure is given in annex D.

5.5.6.2 Error concealment

In case a CRC error is detected, error concealment processing (bad frame masking) is carried out in the HVXC decoder. The state transition diagram of the concealment processing is shown in figure 11. A frame masking state of the current frame is updated based on the decoded CRC result of ESC0. If a CRC error is detected in ESC0, the frame is declared to be a "bad" frame. The initial state of the state transition diagram is state = 0. The arrow with a letter "1" denotes the transition for a bad frame, and that with a letter "0" the one for a good frame. At the 2,0 kbit/s rate ESC1 and ESC2 are protected by CRC bits; at the 4,0 kbit/s ESC1 to ESC3 are protected by CRC bits. The results of the CRC checks against these ESCs are used to maintain toll quality as described in detail below.

5.5.6.2.1 Parameter replacement

According to the state value, the following parameter replacement is done. In error free condition, the state value becomes 0, and received source coder bits are used without any concealment processing.

LSP parameters

At state = 1...6, LSP parameters are replaced with those of previous states. At state = 7, if LSP4 = 0 (LSP quantization mode without inter-frame prediction), then LSP parameters are calculated from all LSP indices received in the current frame. If LSP4 = 1 (LSP quantization mode with inter-frame coding), then LSP parameters are calculated with the following method, where LSP parameters belonging to the LSP1 index are interpolated with the previous LSPs.

$$LSP_{base}(n) = p \times LSP_{prev}(n) + (1 - p) \times LSP_{1st}(n) \text{ for } n = 1...10 \quad (1)$$

$LSP_{base}(n)$ is LSP parameters of the base layer, $LSP_{prev}(n)$ is the decoded LSPs of the previous frame, $LSP_{1st}(n)$ is the decoded LSPs from the current LSP1 index, and p is the interpolation factor. p is changed according to the number of previous bad frames as shown in table 28. LSP indices LSP2, LSP3 and LSP5 are not used, and $LSP_{base}(n)$ computed according to the Equation (1) is used as current LSP parameters.

Table 28: p factor

frame	p
0	0,7
1	0,6
2	0,5
3	0,4
4	0,3
5	0,2
6	0,1
7	0,0

Mute variable

According to the "state" value, a variable "mute" is set to control the output level of the reproduced speech. The "mute" values in table 29 are used. At state = 7, the average of 1,0 and "mute" value of the previous frame (= 0,5 (1,0 + previous "mute value")) is used. However, when this value is more than 0,8, "mute" value is replaced with 0,8.

Table 29: Mute value

state	mute value
0	1,0
1	0,8
2	0,7
3	0,5
4	0,25
5	0,125
6	0,0
7	average/0,8

Replacement and gain control of "voiced" parameters

At state = 1...6, spectrum parameter SE_shape1, SE_shape2, spectrum gain parameter SE_gain, spectrum parameter for 4,0 kbit/s mode SE_shape3 ... SE_shape6 are replaced with the corresponding parameters of the previous frame. Also, to control the volume of the output speech and the harmonic magnitude parameters of LPC residual signal, " $Am[0...127]$ " is gain controlled as shown in equation (2). In the equation, $Am_{(org)}[i]$ is computed from the received spectrum parameters of the latest error free frame.

$$Am[i] = mute \times Am_{(org)}[i] \text{ for } i = 0...127 \quad (2)$$

If the previous frame is unvoiced and the current state is state = 7, equation (2) is replaced with equation (3);

$$Am[i] = 0,6 \times mute \times Am_{(org)}[i] \text{ for } i = 0...127 \quad (3)$$

As described before, SE_shape1 and SE_shape2 are individually protected by 1 bit CRCs. ESC1 applies to SE_shape1 and ESC2 applies to SE_shape2 at 2,0 kbit/s. In the same way, ESC2 applies to SE_shape1 and ESC3 applies to SE_shape2 at 4,0 kbit/s. At state = 0 or 7, if both of the CRCs of SE_shape1 and SE_shape2 are in error at the same time, the quantized harmonic magnitudes with fixed dimension $Am_{qnt}[1...44]$ are gain suppressed as:

$$Am_{qnt}[i] = s[i] \times Am_{qnt(org)}[i] \text{ for } i = 1...44 \quad (4)$$

$s[i]$ is the factor for the gain suppression. $Am_{qnt(org)}[i]$ is the fixed dimension harmonic magnitudes generated using the SE_shape1 and SE_shape2 which contains bit errors. $Am_{qnt}[i]$ is then dimension converted to obtain $Am[i]$.

Table 30: Factor for gain suppression, s[0...44]

i	1	2	3	4	5	6	7...44
$s[i]$	0,10	0,25	0,40	0,55	0,70	0,85	1,00

At 4,0 kbit/s, SE_shape4, SE_shape5, and SE_shape6 are subject to CRC checking as ESC1. When a CRC error is detected, the spectrum parameter of the enhancement layer is not used.

Replacement and gain control of "unvoiced" parameters

At state = 1...6, the stochastic codebook gain parameters VX_gain1[0] and VX_gain1[1] are replaced with the previous frame's VX_gain1[1]. Also the stochastic codebook gain parameters for the 4,0 kbit/s mode VX_gain2[0]...VX_gain2[3] are replaced with the previous frame's VX_gain2[3].

The stochastic codebook shape parameters VX_shape1[0], VX_shape1[1] and the stochastic codebook shape parameters for the 4,0 kbit/s mode VX_shape2[0], VX_shape2[1], VX_shape2[2] and VX_shape2[3] are generated from randomly generated index values.

Finally, in order to control the volume of the output speech, the LPC residual signal $res[0...159]$ is gain controlled as shown in equation (5). $res_{(org)}[i]$ is the excitation signal obtained by using the stochastic codebook gain and the shape parameters generated as explained above.

$$res[i] = mute \times res_{(org)}[i] \quad (0 \leq i \leq 159) \quad (5)$$

Frame masking state transitions

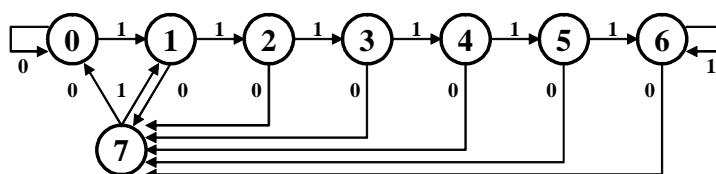


Figure 11: Frame masking state transitions

5.5.7 HVXC + SBR

Figure 12 outlines the audio super frame composition for HVXC + SBR. The HVXC core part is identical to the HVXC audio super frame defined by figure 12 and includes the possible 4 padding bits. The HVXC core has frame lengths of 20 ms, whereas SBR employs 40 ms frames. Thus, two HVXC frames are associated with every SBR frame.

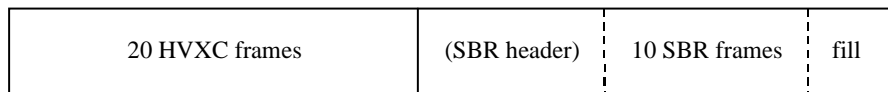


Figure 12: HVXC + SBR audio super frame overview

The number of bits per SBR frame is constant and no signalling of frame locations within the superframe is used. After the last SBR frame, additional bits for byte alignment follow. If the *sbr_header_flag* in the SDC is set (see clause 6.4.3.10) the SBR frames are preceded by a single SBR header. In this mode a wide range of SBR tunings, and correspondingly different overheads, are possible. If the flag is not set, no header is sent and predefined combinations of tuning parameters are used, as defined in clause I.2. The predefined SBR tunings are selected such that the combined data of HVXC + SBR fits in the Very Strong Protected Part of commonly used transmission channel configurations.

5.6 SBR coding

5.6.1 AAC + SBR

The combination of AAC with SBR uses the SBR tool as it is defined in the MPEG-4 Audio standard [2] and the location of the SBR data within the AAC audio super frames is described in clause 5.3.2. This clause only describes how to extract the SBR elements from the SBR data and thus describes the mapping to the format defined in the MPEG-4 standard.

Table 31: Syntax of *sbr_aac_frame()*

Syntax	No. of bits	Note
<pre> sbr_aac_frame(audio_mode) // audio_mode is located in the SDC { sbr_crc_bits if (audio_mode != stereo) sbr_extension_data(ID_SCE, 0); else sbr_extension_data(ID_CPE, 0); } </pre>	8	see annex D
NOTE 1: <i>sbr_extension_data()</i> is defined in the MPEG-4 Audio standard [2]. NOTE 2: <i>sbr_extension_data()</i> uses a variable <i>cnt</i> for the <i>num_align_bits</i> calculation. <i>cnt</i> is not available in the DRM bitstream format and <i>num_align_bits</i> is 0 if <i>bs_extended_data</i> =0 or <i>num_align_bits</i> is the value of <i>num_bits_left</i> after leaving the while (<i>num_bits_left</i> > 7) loop if <i>bs_extended_data</i> =1.		

sbr_crc_bits Cyclic redundancy checksum for the SBR bit stream part. The CRC algorithm is applied to all the *sbr_extension_data_bits*).

5.6.2 CELP/HVXC + SBR

MPEG-4 Audio defines the SBR tool for operation together with AAC. Within DRM, the SBR tool is applied to CELP and HVXC as well, however using a modified bitstream and a different frame length. The following clauses rely on the description of the AAC + SBR decoding process as given in MPEG-4 Audio, and only define the areas in which CELP/HVXC + SBR differs from AAC + SBR.

5.6.2.1 SBR Payload

This clause describes the SBR speech protocol by means of MPEG-style pseudo code. Overviews of the CELP + SBR and HVXC + SBR superframes were given in clauses 5.4.2 and 5.5.7 respectively. The first clause presents the bitstream syntax and the second clause defines the syntactical elements used therein.

5.6.2.1.1 Bitstream Syntax

The below syntax is a direct replacement for the one defined in MPEG-4 Audio, clause "Payloads for the audio object type SBR". It applies when the *SDC audio coding* field equals 1 (CELP) or 2 (HVXC) and the *SBR flag* field is 1. The *SDC audio mode* and *sbr_header_flag* fields are used as parameters (see clause 6.4.3.10).

Table 32: Syntax of SBR speech audio super frame

Syntax	No. of bits	Note
<pre>sbr_speech_asf(audio_mode, sbr_header_flag) { if (sbr_header_flag) { bs_crc_header; sbr_speech_header(); } for (frame = 0; frame < 10; frame++) { if (audio_mode == 1 audio_mode == 3) bs_crc_data; sbr_speech_data(); } bs_fill_bits; }</pre>	<p>6</p> <p>8</p> <p>0..7</p>	<p>see annex D</p> <p>see annex D</p>
<p>NOTE: CRC for the payload is used if the core uses CRC, i.e. the SDC audio mode field contains "01" or "11", regardless of speech codec.</p>		

Table 33: Syntax of SBR speech header

Syntax	No. of bits	Mnemonic
<pre>sbr_speech_header() { bs_start_freq; bs_stop_freq; bs_freq_scale; bs_alter_scale; bs_noise_bands; bs_num_level = 2 * tmp + 2; bs_num_shape = floor(2 ^ (tmp - 1)); bs_num_noise = floor(2 ^ (tmp - 1)); bs_num_invf; bs_num_rfield; bs_limiter_bands; bs_limiter_gains; bs_interpol_freq; if (bs_num_noise == 0) bs_data_noise; if (bs_num_invf == 0) bs_invf_mode; bs_reserved; }</pre>	<p>4</p> <p>4</p> <p>2</p> <p>1</p> <p>2</p> <p>1</p> <p>2</p> <p>2</p> <p>1</p> <p>2</p> <p>2</p> <p>2</p> <p>1</p> <p>3</p> <p>2</p> <p>4</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>
<p>NOTE 1: The condition $bs_num_level \geq bs_num_shape$ must be satisfied.</p> <p>NOTE 2: The condition $bs_num_level \geq bs_num_noise$ must be satisfied.</p>		

Table 35: Start frequency table ($F_s = 32\ 000$ Hz)

bs_start_freq	Frequency (Hz)	QMF start band (k_0)
0	3 000	12
1	3 250	13
2	3 500	14
3	3 750	15
4	4 000	16
5	4 250	17
6	4 500	18
7	4 750	19
8	5 000	20
9	5 250	21
10	5 500	22
11	6 000	24
12	6 500	26
13	7 000	28
14	7 500	30
15	8 000	32

NOTE: bs_start_freq 5 through 15 are only available for wide band CELP.

Table 36: Stop frequency table ($F_s = 32\ 000$ Hz)

bs_stop_freq	Frequency (Hz)	QMF stop band (k_2)
0	8 000	32
1	8 500	34
2	9 000	36
3	9 500	38
4	10 000	40
5	10 500	42
6	11 000	44
7	11 500	46
8	12 000	48
9	12 500	50
10	13 000	52
11	13 500	54
12	14 000	56
13	14 500	58
14	15 000	60
15	16 000	64

5.6.2.2 SBR decoding process

This clause is loosely modelled after MPEG-4 Audio [2], clause "Decoding process".

5.6.2.2.1 Notation, Constants and Defaults

In addition to the definitions given in MPEG-4 Audio, clause "SBR specific notation", the following notation is used:

$a \% b$ a modulo b

floor(x) rounding to the nearest integer towards minus infinity.

The following values should be used when evaluating the decoding equations:

$RATE = 1$ a constant indicating the number of QMF subband samples per time slot.

$numTimeSlots = 20$ number of time slots that exist within one SBR frame.

To a large extent, the SBR speech protocol is a subset of the protocol used for AAC + SBR. In table 37, values of MPEG-4 AAC + SBR bitstream signals that are absent in the speech protocol, yet needed in the decoding process, are defined. These values should be used for the SBR speech case in the evaluation of the SBR decoding equations given in MPEG-4 Audio, clause "Decoding process".

Table 37: Bitstream constants

Element	Value	Note
bs_amp_res	1	Amplitude resolution is always 3,0 dB
bs_xover_band	0	The frequency band table is always used from the lowest index
bs_freq_res	0	Only the low frequency resolution table is used, i.e. $r(l) = 0$ for all l
bs_smoothing_mode	1	Time smoothing of envelopes is not used
bs_add_harmonic_flag	0	Sinusoidal coding is not used

As evident from the pseudo code in table 34 in clause 5.6.2.1.1, the SBR speech protocol has a "layered" structure. Furthermore, the SBR header is optional (see clauses 5.4.2 and 5.5.7). Thus, for some tunings and modes one or more of the syntactical elements may be absent in the bitstream. In these cases, the decoder relies on default values for said elements, as defined in table 38 as well as in annex I.

Table 38: Bitstream defaults

Element	Value	Note
bs_alter_scale	1	extra wide bands in highest range
bs_limiter_bands	2	2.0 bands/octave
bs_limiter_gains	2	3 dB
bs_interpol_freq	1	On
bs_data_noise	0	(see note)
bs_invf_mode	2	intermediate inverse filtering (see note)
NOTE: Future generations of decoders may override those defaults by means of decoder side parameter estimation, or, in case of HVXC, guidance by the VUV signal.		

There is also provision for turning off the shape completely ($bs_num_shape = 0$). This mode is intended for a possible future generation of decoders, which will extrapolate the highband envelope from the lowband. Hereby the level signal will be used as a very low overhead guidance signal. Decoders without the extrapolation capabilities shall simply decode the core codec part only.

5.6.2.2.2 Frequency band tables

The speech decoder utilizes the same scheme for definition of the frequency grouping of the QMF subband samples as is used in AAC + SBR. However, neither a variable crossover frequency between core codec and SBR, nor an adaptive frequency resolution is employed. Thus, only one frequency band table is needed for the envelope coding. There are two different cases:

- 1) sbr_header_flag = 1:

Operation is nearly identical to the AAC + SBR case. The start and stop channels are derived from bs_start_freq and bs_stop_freq by means of tables 35 and 36 respectively (see clause 5.6.2.1.2).

The master frequency band table is calculated in accordance with the flowcharts given in MPEG-4 Audio, clause "Master Frequency Band Table", taking $k0$, $k2$, bs_freq_scale and bs_alter_scale as parameters. Hereby $temp2 = \{1.0, 1.5\}$ is used instead of $temp2 = \{1.0, 1.3\}$. The envelope frequency band table is then given by $f_{TableLow}$ which is derived from f_{Master} as given by MPEG-4 Audio, clause "Derived Frequency Band Tables". The noise floor table is also calculated in accordance with the description in said section, whereby bs_noise_bands is an input parameter.

- 2) sbr_header_flag = 0:

The number of bands, as well as start and stop QMF bands, are given explicitly by tables I.1 through I.4. Thus, the expressions for $numBands0$ and $numBands1$ in MPEG-4 Audio, clause "Master Frequency Band Table" are substituted according to:

$(k2/k0) > 2,2449$ is False:

$$numBands0 = num_shape_bands$$

(numBands1: N/A)

$(k2/k0) > 2,2449$ is True:

$$\text{numBands0} = \text{NINT}(\text{num_shape_bands} \times (1/(1 + \log(k2/k1)/(\log(2) \times 1,5))))$$

$$\text{numBands1} = \text{num_shape_bands} - \text{numBands0}$$

Furthermore, here the evaluation of the flowchart in MPEG-4 Audio, figure "Flowchart calculation of f_{Master} when $\text{bs_freq_scale} > 0$ " yields f_{TableLow} directly, i.e. the conversion from f_{Master} to $f_{\text{TableHigh}}$ to f_{TableLow} is bypassed. The noise floor table is derived from f_{TableLow} according to MPEG-4 Audio, clause "Derived Frequency Band Tables", however using $N_Q = \text{num_noise_bands}$.

In both the above cases the number of envelope frequency bands, num_shape_bands , and noise floor bands, num_noise_bands , are used for subsequent decoding of the SBR data frames, $\text{sbr_speech_data}()$.

5.6.2.2.3 Time / Frequency grid

Contrary to the AAC + SBR protocol, the speech codec version does not have variable frame boundaries. In terms of the frame classes defined in MPEG-4 Audio, clause "Time / Frequency Grid Generation", this corresponds to constant usage of $\text{bs_frame_class} = \text{FIXFIX}$. One frame comprises 20 QMF subband samples, which in this context corresponds to 20 time slots. Furthermore, adaptive switching of time and frequency resolution is not supported.

The envelope time boundaries (in "time slots") are given by:

$$\mathbf{t}_E = \begin{cases} [0, 10, 20] & \text{for } \text{bs_num_level} = 2 \\ [0, 5, 10, 15, 20] & \text{for } \text{bs_num_level} = 4 \end{cases}$$

Note that the envelope time boundaries are controlled by the bs_num_level parameter. When the bs_num_shape is set to a lower value, a given shape vector is applied to more than one envelope. This is further described in clause 5.6.2.2.4.

Similarly, the noise floor time boundaries (in "time slots") are given by:

$$\mathbf{t}_Q = \begin{cases} [0, 20] & \text{for } \text{bs_num_noise} = \{0, 1\} \\ [0, 10, 20] & \text{for } \text{bs_num_noise} = 2 \\ [0, 5, 10, 15, 20] & \text{for } \text{bs_num_noise} = 4 \end{cases}$$

5.6.2.2.4 Envelope and noise floor decoding and dequantization

Decoding of envelope scalefactors and noise floor scalefactors differs from the AAC + SBR case and is fully described below (replacing MPEG-4 Audio, clause "SBR Envelope and Noise Floor Decoding").

In order to calculate the envelope scalefactors from the level and shape bitstream elements, let:

$$\begin{aligned} L_E &= \text{bs_num_shape} \\ N_E &= \text{num_shape_bands} \end{aligned}$$

A level matrix, \mathbf{L} , is derived from the bitstream element bs_data_level according to:

$$\mathbf{L}(k, l) = \text{bs_data_level}[l], \begin{cases} 0 \leq k < N_E \\ 0 \leq l < L_E \end{cases}$$

Similarly, a shape matrix, \mathbf{C}_{read} , is derived from bs_data_shape according to:

$$\mathbf{C}_{\text{read}}(k, l) = \text{bs_data_shape}[l][k] - \text{SHAPE_LAV}, \begin{cases} 0 \leq k < N_E \\ 0 \leq l < \text{bs_num_shape} \end{cases}$$

where $\text{SHAPE_LAV} = 4$.

A spectral slope is applied to the shape matrix according to:

$$\mathbf{C} = \mathbf{C}_{read} + \mathbf{T}$$

where:

$$\mathbf{T}(k, l) = trend[k], \begin{cases} 0 \leq k < N_E \\ 0 \leq l < bs_num_shape \end{cases}$$

and $trend[k]$ is given by table 39.

Table 39: Definition of trend[k]

k	trend[k]
0	2
1	2
2	0
3	0
4	-1
5	-1
>5	-2

The shape matrix is mapped according to:

$$\mathbf{C}_{mapped}(k, l) = \mathbf{C}(k, i(l)), \begin{cases} 0 \leq k < N_E \\ 0 \leq l < L_E \end{cases}, i(l) = \begin{cases} \text{floor}((l-1)/r) & \text{if } l \% r > 0 \\ \text{floor}(l/r) & \text{if } l \% r = 0 \end{cases}, r = \frac{bs_num_level}{bs_num_shape}$$

and the envelope scalefactors, \mathbf{E} , are calculated from the level and shape matrices as:

$$\mathbf{E} = \mathbf{L} + \mathbf{C}_{mapped}$$

The envelope scalefactors are dequantized as described in MPEG-4 Audio, clause "Dequantization and Stereo Decoding" for the single channel element case.

In order to calculate the noise floor, let:

$$\begin{aligned} L_Q &= bs_num_noise \\ N_Q &= num_noise_bands \end{aligned}$$

A noise floor matrix, \mathbf{Q}_{read} , is derived from bs_data_noise according to:

$$\mathbf{Q}_{read}(k, l) = bs_data_noise[l][k] - NOISE_LAV, \begin{cases} 0 \leq k < N_Q \\ 0 \leq l < L_Q \end{cases}$$

where $NOISE_LAV = 4$. When $bs_num_noise = 0$, $bs_data_noise[l][k]$ equals the default value of bs_data_noise ($sbr_header_flag = 0$) or bs_data_noise as signalled in the SBR header ($sbr_header_flag = 1$) for every frame, l and k , or is estimated by other means, see clause 5.6.2.2.1.

The maximum values are expanded and a static offset is added according to:

$$\mathbf{Q}(k, l) = \begin{cases} NOISE_MUTED & \forall (k, l) : \mathbf{Q}_{read}(k, l) = 3 \\ \mathbf{Q}_{read}(k, l) + NOISE_OFFSET & \text{otherwise} \end{cases}$$

where $NOISE_MUTED = 11$ and $NOISE_OFFSET = 4$.

Like the envelope scalefactors, the noise floor scalefactors are dequantized as described in MPEG-4 Audio, clause "Dequantization and Stereo Decoding" for the single channel element case.

5.6.2.3 SBR filterbanks and tool overview

In principle the same QMF banks that are used for AAC + SBR, as described in MPEG-4 Audio, are used for CELP/HVXC + SBR. Furthermore, MPEG-4 Audio, clause "SBR tool overview" applies to CELP/HVXC + SBR with minor modifications: Clearly all references to the "AAC core" should be interpreted as references to the CELP or HVXC core, and "Huffman decoding" is substituted by the procedures defined in clause 5.6.2.2.4. A number of speech codec output frames are concatenated according to table 40, forming frames of length 320 samples at 8 kHz sampling rate or 640 samples at 16 kHz sampling rate, which are fed to the analysis filterbank. HF generation and HF adjustment operates at 32 kHz sampling rate, and upsampling by a factor 2 or 4 is performed in the analysis filterbank according to table 40. The analysis filterbank of size 32 is described in MPEG-4 Audio. The size 16 case is defined by the flowchart in figure 13 and the below description, where an array \mathbf{x} consisting of 160 time domain input samples is assumed. A higher index into the array corresponds to older samples.

- Shift the samples in the array \mathbf{x} by 16 positions. The oldest 16 samples are discarded and 16 new samples are stored in positions 0 to 15.
- Multiply the samples of array \mathbf{x} by every forth coefficient of window \mathbf{c} . The window coefficients can be found in the MPEG-4 Audio SBR Tables Annex, Table "Coefficients $c[i]$ of the QMF bank window".
- Sum the samples according to the formula in the flowchart to create the 32-element array \mathbf{u} .

Calculate 16 new subband samples by the matrix operation $\mathbf{M}\mathbf{u}$, where:

$$M(k, n) = 4 \cdot \exp\left(\frac{i \cdot \pi \cdot (k + 0,5) \cdot (2 \cdot n - 0,25)}{32}\right), \begin{cases} 0 \leq k < 16 \\ 0 \leq n < 32 \end{cases}$$

In the equation, $\exp()$ denotes the complex exponential function and i is the imaginary unit.

Every loop in the flowchart produces 16 complex-valued subband samples, each representing the output from one filterbank subband. For every SBR frame the filterbank will produce $numTimeSlots \cdot RATE$ subband samples for every subband, corresponding to a time domain signal of length $numTimeSlots \cdot RATE \cdot 16$ samples. In the flowchart $\mathbf{X}_{Low}[k][l]$ corresponds to subband sample l in QMF subband k .

The MPEG-4 Audio "Synchronization and timing" figure defines the buffers and synthesis windowing for CELP/HVXC + SBR, when using constants according to clause 5.6.2.2.1, and taking the output frame length of 1 280 time samples as well as the specific upsampling factor into account.

Table 40: Relations between core codec and SBR frames

Core	Core sampling frequency [kHz]	Core frame length [ms]	Core frames per SBR frame	Upsampling factor	Analysis Filterbank Size
CELP	16	10	4	2	32
CELP	16	20	2	2	32
CELP	8	10	4	4	16
CELP	8	20	2	4	16
CELP	8	40	1	4	16
HVXC	8	20	2	4	16

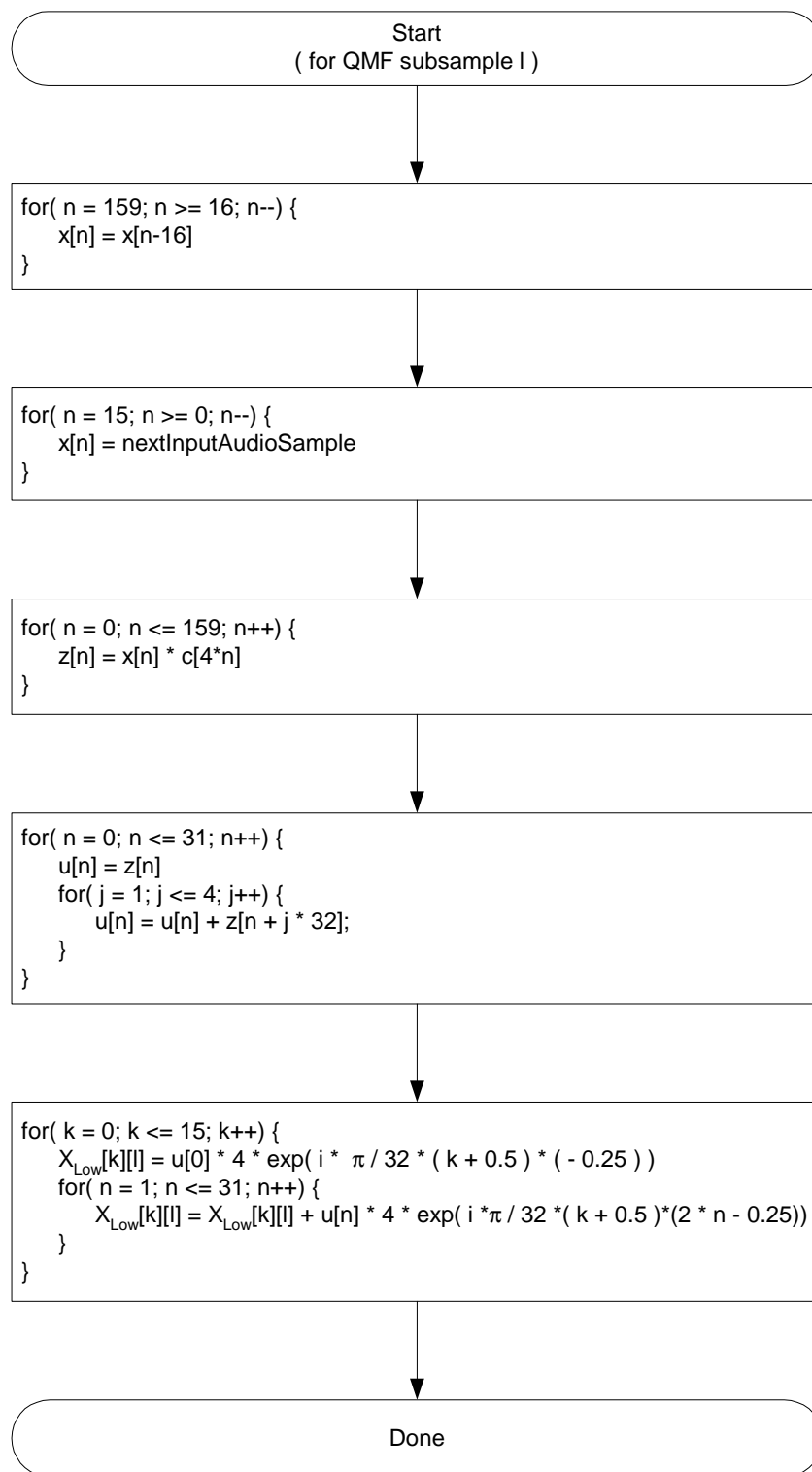


Figure 13: Flowchart of decoder 16-channel analysis QMF bank

5.6.2.4 HF Generation and HF adjustment

The high frequency generation, i.e. transposition and inverse filtering, is very similar to the scheme given in MPEG-4 Audio, clause "HF Generation". However, since only one inverse filtering parameter, **bs_invf_mode**, is signalled, the calculation of the chirp factors is slightly different. Thus **bwArray(i)** as given by MPEG-4 Audio is substituted by **bwArray(p)** where p denotes the patch number further defined below, and

$$\text{bwArray}(0) = \begin{cases} 0 & \text{if } \text{tempBw} < 0,015625 \\ \text{tempBw} & \text{if } \text{tempBw} \geq 0,015625 \end{cases}$$

where tempBw is calculated as:

$$\text{tempBw} = \begin{cases} 0,75000 \cdot \text{newBw} + 0,25000 \cdot \text{bw}' & \text{if } \text{newBw} < \text{bw}' \\ 0,90625 \cdot \text{newBw} + 0,09375 \cdot \text{bw}' & \text{if } \text{newBw} \geq \text{bw}' \end{cases}$$

and newBw is calculated in accordance with the MPEG-4 standard. The value for patches $p > 0$ is given by:

$$\text{bwArray}(p) = \begin{cases} 0 & \text{if } \text{bwArray}(0) = 0 \\ \min(\text{bwArray}(0) + 0,20, 0,99609375) & \text{if } \text{bwArray}(0) \neq 0 \end{cases}$$

The HF generator patch is built in accordance with the flowchart in figure 14, which replaces the one given in the MPEG-4 standard. The HF generation is obtained according to:

$$\begin{aligned} X_{High}(k, l + t_{HFAdj}) = & s \left[X_{Low}(p, l + t_{HFAdj}) + \text{bwArray}(p) \cdot \alpha_0(p) \cdot X_{Low}(p, l - 1 + t_{HFAdj}) \right. \\ & \left. + [\text{bwArray}(p)]^2 \cdot \alpha_1(p) \cdot X_{Low}(p, l - 2 + t_{HFAdj}) \right] \end{aligned}$$

where:

$$\begin{cases} k = k_x + x + \sum_{q=0}^{i-1} \text{patchNumSubbands}(q) \\ p = \text{patchStartSubband}(i) + x \end{cases}$$

for $0 \leq x < \text{patchNumSubbands}(i)$, $0 \leq i < \text{numPatches}$, $\text{RATE} \cdot \mathbf{t}_E(0) \leq l < \text{RATE} \cdot \mathbf{t}_E(L_E)$, and

$$s = \begin{cases} -1 & p+k : \text{odd} \text{ and } l : \text{odd} \\ 1 & \text{otherwise} \end{cases}.$$

Finally, HF adjustment differs from MPEG-4 Audio in that $\mathbf{G}_{Boost}(m, l) = 1$ and $\mathbf{Q}_{Lim}(m, l) = \mathbf{Q}_M(m, l)$ replace the corresponding expressions in the "Calculation of gain" clause.

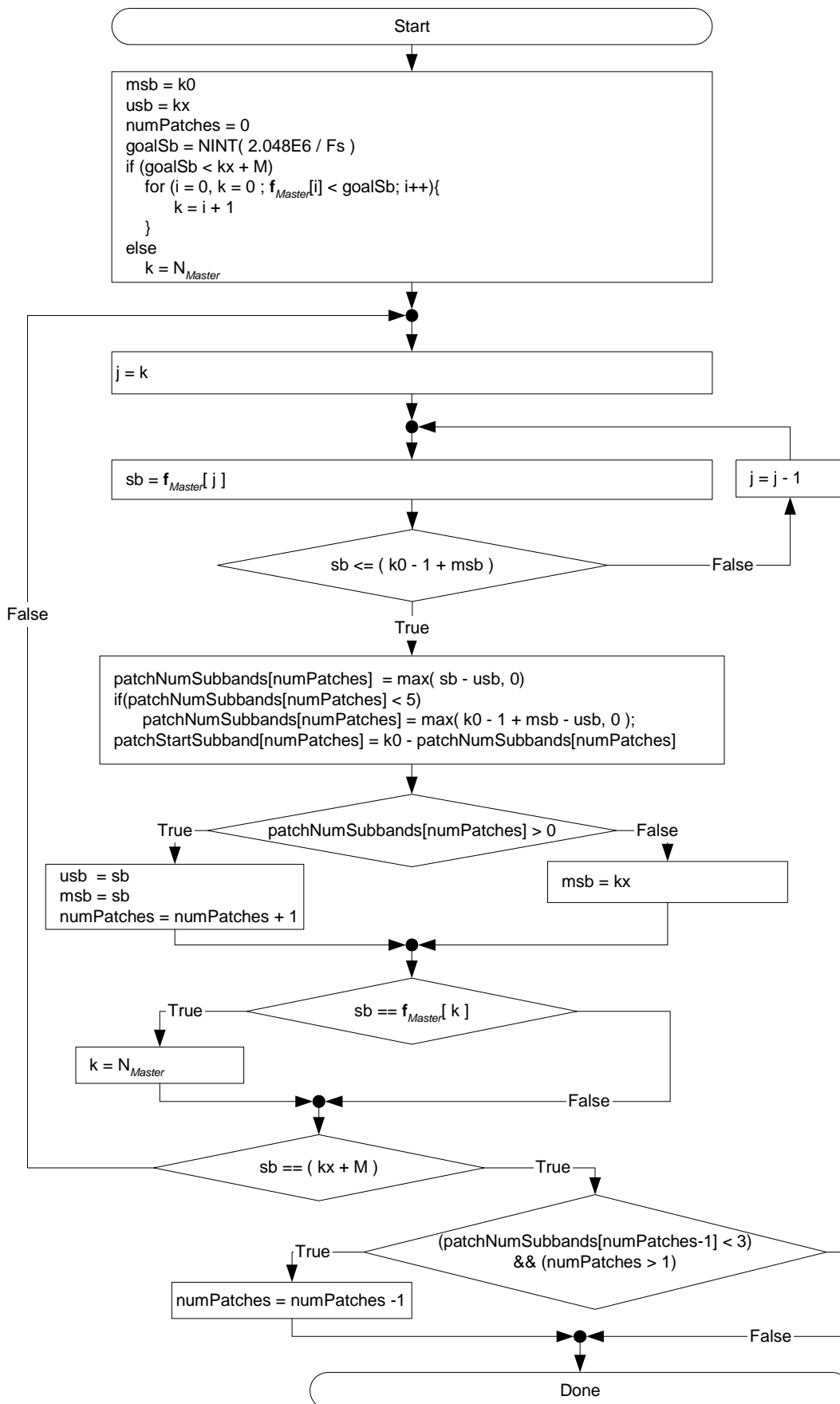


Figure 14: Flowchart of patch construction

5.7 Parametric Stereo coding

5.7.1 Introduction

In the present document the audio coder AAC + SBR, that combines AAC with the bandwidth extension tool SBR, is included according to clauses 5.3.2 and 5.6.1. For improved performance at low bitrate stereo coding, a Parametric Stereo (PS) coder partly based upon the SBR framework is available.

Clause 5.7.2 outlines the technical concept as well as the advantages of Parametric Stereo (PS) coding for AAC + SBR.

The general idea with PS coding is to send stereo image describing data as side information along with a downmixed mono signal. This stereo side information is very concise and only requires a small fraction of the total bitrate allowing the mono signal to have maximum quality for the total bitrate given.

The PS coding method here defined, combines frequency selective panorama technique with a stereo ambience reconstruction technique. This allows a stereo image reconstruction well suited for both loudspeaker and headphones playback.

The stereo synthesis at the decoder reconstructs spatial properties but does not affect the total spectral energy. Hence, there is no colorization of the frequency spectrum compared to the mono compatible core signal. Further advantage are the backward compatibility which allows a decoder not supporting PS to successfully decode the mono core. This is possible as the PS data is included in the extended data field in the bitstream which optionally can be ignored.

The design target bitrates for applying parametric stereo coding to AAC + SBR are in the range 18 kbit/s to 26 kbit/s, however the technique may be used at any bitrate.

5.7.2 Technical overview

5.7.2.1 Stereo parameters

In the Parametric Stereo (PS) tool two different stereo parameters are used for describing the stereo image, Panorama (Pan) and Stereo Ambience (SA). The Pan parameter contains information about the frequency selective level differences between left and right channels while SA parameter contains frequency selective information about the stereo ambience that cannot be described with just the left-right energy distribution.

When considering a mid-side representation of most stereo signals, reverberation is usually over-represented in the side channel compared to the mid channel. Due to that fact much ambience is lost after a mono downmix. The SA synthesis is a method to regenerate the lost ambience by adding a synthetic version of it to the side channel again.

For frequency domain analysis and synthesis, the native filterbank (QMF bank) included in the SBR system is used. Since no additional time-frequency transform is necessary, PS encoding and decoding gives very little complexity overhead. The QMF bank used in the SBR system is a 64 channel complex-valued filterbank with alias-free behaviour even when altering the gains of neighbouring subbands excessively.

The SA and Pan parameters are updated every SBR frame and interpolated between the frames to get smooth transitions. SA and Pan are treated as individual parameters. However, the impact of the SA modified signal is to some degree correlated with the values of the Pan parameters. At higher Pan values, i.e. at more uneven left-right power distribution, SA tends to be less significant. That motivates the co-existence of SA and Pan.

5.7.2.2 Signalling

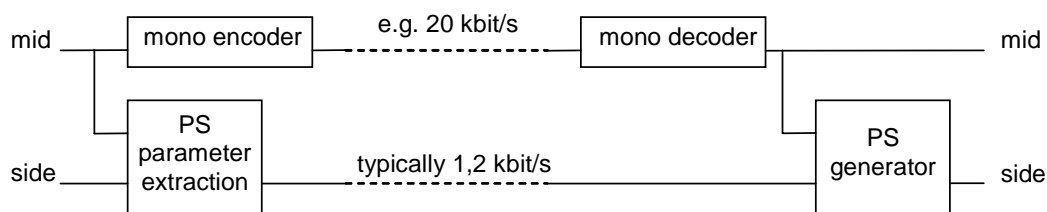


Figure 15: Topology of a codec system using parametric stereo

The technical concept of PS coding is easily depicted by using a mid-side, instead of a left-right, representation of the stereo signal. It is done by the simple transform:

- $\text{mid} = \text{left} + \text{right}$.
- $\text{side} = \text{left} - \text{right}$.

According to the topology view in figure 15, it is clear that the PS data only affects the signal at the very end of the decoding process. Furthermore, it can be concluded that PS encoding and decoding are independent of the actual audio codec used for the mono coding.

The PS encoding produces a small amount of data that is used to reconstruct the stereo image at the decoder. The data includes the SA and Pan parameters as described in clause 5.7.3 but also some few miscellaneous control data bits. For the AAC + SBR configuration using the bitstream syntax according to MPEG-4 Audio [2], the PS data is located in the extended data field. This assures backwards compatibility towards MPEG-4 Audio since any decoder not supporting PS will just ignore that part in the extended data. However, this PS tool is not available in combination with the low power SBR tool.

The signalling rate for PS is dynamic because of entropy coding but can be expected to have a bitrate distribution that rarely (e.g. less than 1 %) exceeds 2 kbit/s for stereo material. The PS data rate is ultimately chosen by the encoder and can therefore easily be limited to a maximum value if desired.

When applying PS on AAC + SBR at 24 kbit/s the average bitrate distributions will typically be:

- 22,8 kbit/s core data (AAC + SBR); and
- 1,2 kbit/s PS data for stereo signals.

For pure mono signals the PS data will only take 0,05 kbit/s and hence no significant degradation of the mono signal will occur. Also for stereo signals with a stereo width close to mono the PS data rate will be lower than average. However, for pure mono broadcasting PS is not signalled at all. Table 41 shows typical ratings for the PS bitrate.

Table 41: Typical bitrates of the PS data

Signal type	PS data rate
mono signals, extended data disabled	0 bit/s
mono signals, extended data enabled	50 bit/s
complex stereo music (average)	1 200 bit/s
arbitrary stereo signals (maximum peak)	2 500 bit/s (recommended maximum for encoder)

5.7.3 Bitstream protocol

5.7.3.1 Bitstream integration

The Parametric Stereo (PS) data is conveyed in the SBR extended data field. PS signals a unique ID in the SBR extended data field defined by the bitstream element **bs_extension_id**. To be successfully decoded, PS needs to receive data from single channel elements in the bitstream, i.e. from a mono bitstream.

5.7.3.2 PS syntax

The function `sbr_extension()` used in `sbr_channel_pair_base_element()` and `sbr_channel_pair_element()` described in MPEG-4 Audio, clause "Payloads for the audio object type SBR", is defined as follows.

Table 42: Syntax of sbr_extension()

Syntax	No. of bits	Note
<pre>sbr_extension(bs_extension_id, num_bits_left) { switch(bs_extension_id) { case PARAMETRIC_STEREO: bits_read = ps_data(); num_bits_left -= bits_read; break; case RESERVED_1: reserved; break; case RESERVED_2: reserved; break; case RESERVED_3: reserved; break; } }</pre>		1 2 3 3 3
<p>NOTE 1: ps_data() returns <i>bits_read</i> which is the total number of bits read by ps_data().</p> <p>NOTE 2: The variable <i>num_bits_left</i> is the same as used in the sbr_pair_base_element() and sbr_channel_pair_element().</p> <p>NOTE 3: ID:s for future use, see table 46 in clause 5.7.3.3.</p>		

Table 43: Syntax of ps_data()

Syntax	No. of bits	Note
<pre>ps_data() { if (bs_enable_sa) sbr_sa_element() if (bs_enable_pan) sbr_pan_element() }</pre>	1 1	

Table 44: Syntax of sbr_sa_element()

Syntax	No. of bits	Note
<pre>sbr_sa_element() { if (bs_sa_dt_flag) sa_huff = t_huffman_sa; else sa_huff = f_huffman_sa; for (band = 0 ; band < NUM_SA_BANDS ; band++) bs_sa_data[band] = huff_dec(sa_huff, bs_codeword); }</pre>	1 1..9	 1 2
<p>NOTE 1: NUM_SA_BANDS = $N_{SABands}$, defined in clause 5.7.4.1.</p> <p>NOTE 2: huff_dec() is explained further in annex I.</p>		

Table 45: Syntax of sbr_pan_element()

Syntax	No. of bits	Note
<pre>sbr_pan_element() { if (bs_pan_dt_flag) pan_huff = t_huffman_pan; else pan_huff = f_huffman_pan; for (band = 0 ; band < NUM_PAN_BANDS ; band++) bs_pan_data[band] = huff_dec(pan_huff, bs_codeword); }</pre>	1	
	1..18	1 2
NOTE 1: NUM_PAN_BANDS = $N_{PanBands}$, defined in clause 5.7.4.1.		
NOTE 2: huff_dec() is explained further in annex I.		

5.7.3.3 Bitstream element definitions

bs_extension_id Holds an ID of the extended data element.

Table 46: Definition of bs_extension_id

bs_extension_id	Meaning	Note
0	PARAMETRIC STEREO	
1	Reserved ID for future use	
2	Reserved ID for future use	
3	Reserved ID for future use	

bs_enable_sa Enables or disables Stereo Ambience (SA) if read as 1 or 0, respectively.

Table 47: Definition of bs_enable_sa

bs_enable_sa	Meaning	Note
0	Bypass Stereo Ambience (SA)	
1	Use Stereo Ambience (SA)	

bs_enable_pan Enables or disables Pan if read as 1 or 0, respectively.

Table 48: Definition of bs_enable_pan

bs_enable_pan	Meaning	Note
0	Bypass Pan	
1	Use Pan	

bs_sa_dt_flag Indicates whether to apply delta decoding in time or frequency direction on the SA data.

Table 49: Definition of bs_sa_dt_flag

bs_sa_dt_flag	Meaning	Note
0	Apply delta coding in frequency direction.	
1	Apply delta coding in time direction.	

bs_pan_dt_flag Indicates whether to apply delta decoding in time or frequency direction on the Pan data.

Table 50: Definition of bs_pan_dt_flag

bs_pan_dt_flag	Meaning	Note
0	Apply delta coding in frequency direction.	
1	Apply delta coding in time direction.	

5.7.4 PS Decoding

5.7.4.1 Decoding overview

The PS decoding process is divided into two independent parts, the SA process and the Pan process. Since the PS processing is operating on the subband samples in the QMF domain before the final QMF bank synthesis, synthesis on the two output channels has to be performed after the PS decoding process. SA uses a mono signal as input while Pan uses a stereo signal as input. The signal flow follows the scheme depicted in figure 16.

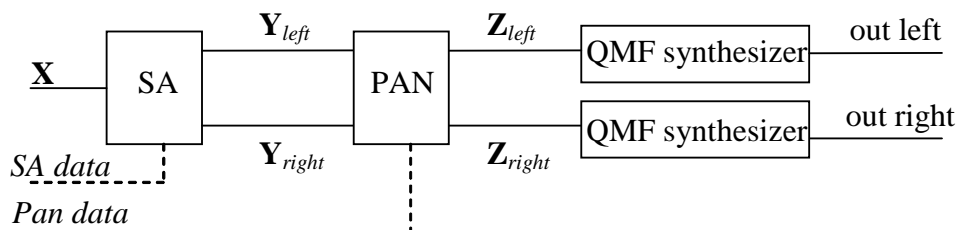


Figure 16: PS decoder module overview

PS decoder specific variables/notation:

F_s	sampling frequency of the SBR enhanced signal.
\mathbf{X}	is the input to the PS decoder, as a complex QMF bank subband matrix.
\mathbf{Y}_{left}	is the left output of the SA decoder, as a complex QMF bank subband matrix.
\mathbf{Y}_{right}	is the right output of the SA decoder, as a complex QMF bank subband matrix.
\mathbf{Z}_{left}	is the left output of the Pan decoder, as a complex QMF bank subband matrix.
\mathbf{Z}_{right}	is the right output of the Pan decoder, as a complex QMF bank subband matrix.
\mathbf{g}_{Sa}	is the SA gain for each SA band.
$\mathbf{g}_{SaMapped}$	is the SA gain for each QMF band.
$\mathbf{G}_{SaMapped}$	is the SA gain for each QMF band and subsample.
$\mathbf{g}_{SaDirMapped}$	is the SA direct gain compensation for each QMF band.
$\mathbf{G}_{SaDirMapped}$	is the SA direct gain compensation for each QMF band and subsample.
\mathbf{g}_{Pan}	is the Pan data for each SA band.
$\mathbf{g}_{PanMapped}$	is the Pan data for each QMF band.
$\mathbf{G}_{PanMapped}$	is the Pan for each QMF band and subsample.

$\mathbf{X}(m, n)$ is subband sample "n" in band "m". If a QMF bank subband matrix is indexed by a negative number as time index, e.g. $\mathbf{X}(m, -1)$, this refers to the sample $\mathbf{X}'(m, NUM_OF_SUBSAMPLES - 1)$ in the previous frame.

Constants:

$N_{SaBands} = 8$	is the number of SA scalefactor bands.
$N_{PanBands} = 20$	is the number of Pan scalefactor bands.
$DECAY_CUTOFF = 3$	is the start frequency band for the ambience decay slope.
$DECAY_SLOPE = 0,05$	is the ambience decay slope.

$NUM_OF_LINKS = 3$ is the number of filter links for the ambience generator.

$NUM_OF_QMF_CHANNELS = 64$ is the number of QMF channels.

$NUM_OF_SUBSAMPLES = 30$ is the number of QMF subsamples in each QMF channel for one frame.

Sampling Rate dependencies:

The following data from tables in annex I are sampling rate dependant according to below:

$$\text{SA frequency scale vector, } \mathbf{f}_{SaTable} = \begin{cases} \mathbf{f}_{SaTable\ 24\text{kHz}} & , F_s < 24\text{kHz} \\ \mathbf{f}_{SaTable\ 48\text{kHz}} & , F_s \geq 24\text{kHz} \end{cases}$$

$$\text{Delay length vector, } \mathbf{d} = \begin{cases} \mathbf{d}_{24\text{kHz}} & , F_s < 24\text{kHz} \\ \mathbf{d}_{48\text{kHz}} & , F_s \geq 24\text{kHz} \end{cases}$$

$$\text{Peak decay factor, } a = \begin{cases} a_{Decay\ 24\text{kHz}} & , F_s < 24\text{kHz} \\ a_{Decay\ 48\text{kHz}} & , F_s \geq 24\text{kHz} \end{cases}$$

$$\text{Smoothing coefficient, } a_{Smooth} = \begin{cases} 0,6 & , F_s < 24\text{kHz} \\ 0,25 & , F_s \geq 24\text{kHz} \end{cases}$$

5.7.4.2 Dequantization and mapping

Both SA and Pan parameters are delta coded in either time or frequency direction. The vectors bs_sa_data and bs_pan_data contain delta coded indices. $bs_sa_dt_flag$ and $bs_pan_dt_flag$ indicate whether they are delta coded in time or frequency direction. The absolute indices, denoted by $\mathbf{g}_{SaIndex}$ and $\mathbf{g}_{PanIndex}$, are extracted by the formulas below:

$$\mathbf{g}_{SaIndex}(i) = \begin{cases} \mathbf{g}'_{SaIndex}(i) + bs_sa_data(i) & , bs_sa_dt_flag = 1 \quad , 0 \leq i < N_{SaBands} \\ \mathbf{g}_{SaIndex}(i-1) + bs_sa_data(i) & , bs_sa_dt_flag = 0 \quad , 1 \leq i < N_{SaBands} \\ bs_sa_data(i) & , bs_sa_dt_flag = 0 \quad , i = 0 \end{cases}$$

$$\mathbf{g}_{PanIndex}(i) = \begin{cases} \mathbf{g}'_{PanIndex}(i) + bs_pan_data(i) & , bs_pan_dt_flag = 1 \quad , 0 \leq i < N_{PanBands} \\ \mathbf{g}_{PanIndex}(i-1) + bs_pan_data(i) & , bs_pan_dt_flag = 0 \quad , 1 \leq i < N_{PanBands} \\ bs_pan_data(i) & , bs_pan_dt_flag = 0 \quad , i = 0 \end{cases}$$

where ' denotes the values from the previous frame.

After Huffman and delta decoding of the SA and Pan data, the vectors $\mathbf{g}_{SaIndex}$ and $\mathbf{g}_{PanIndex}$ will contain indices used to receive the actual values for decoding. The tables associated to those indices are frequency selective and can be found in the quantization matrices $\mathbf{C}_{SaQMatrix}$ and $\mathbf{C}_{PanQMatrix}$ for SA and Pan, respectively. Which table to use in the matrices, for a certain SA or Pan scalefactor band, are defined by the quantization class vectors $\mathbf{c}_{SaQClass}$ and $\mathbf{c}_{PanQClass}$. The dequantization is done according to the following formulas:

$$\mathbf{g}_{Sa}(i) = \mathbf{C}_{SaQMatrix}(\mathbf{c}_{SaQClass}(i), \mathbf{g}_{SaIndex}(i)), \quad 0 \leq i < N_{SaBands}$$

$$\mathbf{g}_{Pan}(i) = \begin{cases} -\mathbf{C}_{PanQMatrix}(\mathbf{c}_{PanQClass}(i), -\mathbf{g}_{PanIndex}(i)) & , \mathbf{g}_{PanIndex}(i) < 0 \\ \mathbf{C}_{PanQMatrix}(\mathbf{c}_{PanQClass}(i), \mathbf{g}_{PanIndex}(i)) & , \mathbf{g}_{PanIndex}(i) \geq 0 \end{cases} , 0 \leq i < N_{PanBands}$$

where $\mathbf{c}_{SaQClass}$ and $\mathbf{c}_{PanQClass}$ are defined in tables I.10 and I.11.

The SA and Pan data extracted from the bitstream are vectors containing data elements representing a frequency range of several QMF bands. In order to simplify the explanation below, and sometimes out of necessity, this grouped data is mapped to the highest available frequency resolution for the envelope adjustment, i.e. the number of QMF channels. This means that several adjacent channels in the mapped vectors will have the same value.

The mapping of the SA and Pan data is outlined below:

$$\begin{aligned} \mathbf{g}_{SaMapped}(m) &= \mathbf{g}_{Sa}(i) & , \mathbf{f}_{SaTable}(i) \leq m < \mathbf{f}_{SaTable}(i+1) & , 0 \leq i < N_{SaBands} \\ \mathbf{g}_{PanMapped}(m) &= \mathbf{g}_{Pan}(i) & , \mathbf{f}_{PanTable}(i) \leq m < \mathbf{f}_{PanTable}(i+1) & , 0 \leq i < N_{PanBands} \end{aligned}$$

where $\mathbf{f}_{SaTable}$ and $\mathbf{f}_{PanTable}$ are defined in tables I.14 and I.15.

5.7.4.3 SA decoding

As is shown in figure 16, the SA processing block takes the input signal \mathbf{X} along with the SA parameters to create the SA output signals \mathbf{Y}_{Left} and \mathbf{Y}_{Right} . \mathbf{X} is the sum of the low- and high-band in the QMF domain, where the high-band is the signal in the SBR system that has just been high frequency adjusted (i.e. corresponding to the signal \mathbf{Y} defined in MPEG-4 Audio, clause "Assembling HF Signals").

Calculate SA gain matrix, $\mathbf{G}_{SaMapped}$

As a first step, the frame based SA gain matrices are calculated.

The SA gain factors, $\mathbf{g}_{SaDirMapped}$ and $\mathbf{g}_{SaMapped}$ are interpolated linearly within the frame according to:

$$\begin{aligned} \mathbf{k}_{SaMapped}(m) &= \frac{\mathbf{g}_{SaMapped}(m) - \mathbf{g}'_{SaMapped}(m)}{NUM_OF_SUBSAMPLES} & , 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands}) \\ \mathbf{k}_{SaDirMapped}(m) &= \frac{\mathbf{g}_{SaDirMapped}(m) - \mathbf{g}'_{SaDirMapped}(m)}{NUM_OF_SUBSAMPLES} & , 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands}) \end{aligned}$$

where $\mathbf{g}_{SaDirMapped}(m) = \sqrt{1 - \mathbf{g}_{SaMapped}^2(m)}$ and ' denotes the values from the previous frame. $\mathbf{k}_{SaMapped}(m)$ and $\mathbf{k}_{SaDirMapped}(m)$ are the slopes used for the interpolation. This gives the interpolated gain values for one frame according to:

$$\begin{aligned} \mathbf{G}_{SaMapped}(m, n) &= \mathbf{g}'_{SaMapped}(m) + n \cdot \mathbf{k}_{SaMapped}(m) & , \begin{cases} 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands}) \\ 0 \leq n < NUM_OF_SUBSAMPLES \end{cases} \\ \mathbf{G}_{SaDirMapped}(m, n) &= \mathbf{g}'_{SaDirMapped}(m) + n \cdot \mathbf{k}_{SaDirMapped}(m) & , \begin{cases} 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands}) \\ 0 \leq n < NUM_OF_SUBSAMPLES \end{cases} \end{aligned}$$

Calculate ambience side signal, $\mathbf{S}_{Ambience}(z, m)$

The ambience process is described in the Z-domain. Its transfer function for each QMF band is defined by:

$$\mathbf{H}_{Sa}(z, m) = \prod_{k=0}^{NUM_OF_LINKS-1} \frac{\mathbf{Q}_{Fract}(m, k) z^{-\mathbf{d}(k)} - \mathbf{a}(k) \mathbf{g}_{DecaySlope}(m)}{1 - \mathbf{a}(k) \mathbf{g}_{DecaySlope}(m) \mathbf{Q}_{Fract}(m, k) z^{-\mathbf{d}(k)}}, \quad 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$$

where the filter coefficient vector $\mathbf{a}(k)$ and the delay length vector $\mathbf{d}(k)$ are defined in tables I.16 and I.17, respectively. The fractional delay length matrix, $\mathbf{Q}_{Fract}(m, k)$ is defined by using the fractional delay length vector $\mathbf{q}(k)$, which also is defined in table I.18, by:

$$\mathbf{Q}_{Fract}(m, k) = \exp\left(-i\pi\mathbf{q}(k)\left(m + \frac{1}{2}\right)\right), \quad \begin{cases} 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands}) \\ 0 \leq k < NUM_OF_LINKS \end{cases}$$

where $i = \sqrt{-1}$ denotes the imaginary unit.

The vector $\mathbf{g}_{DecaySlope}$ contains time invariant factors for making the ambience decay frequency variant. It is defined by:

$$\mathbf{g}_{DecaySlope}(m) = \begin{cases} 1 - DECAFY_SLOPE \cdot (m - DECAFY_CUTOFF) & , m > DECAFY_CUTOFF \\ 1 & , otherwise \end{cases}$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$

Let $\mathbf{S}_{Ambience}(z, m)$ be the ambience signal and $\mathbf{X}(z, m)$ the mono input signal in the Z-domain for each QMF band.

Then $\mathbf{S}_{Ambience}(z, m)$ is defined according to below.

$$\mathbf{S}_{Ambience}(z, m) = 1,5625 \cdot z^{-1} \cdot \mathbf{H}_{Sa}(z, m) \cdot \mathbf{X}(z, m) \quad , \quad 0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$$

Perform transient detection

To be able to handle transients and other fast time-envelopes, the ambience has to be attenuated for those signals. It is done by the following scheme:

Apply peak decay on the input power signal according to:

$$\mathbf{X}_{PeakDecayNrg}(m, n) = \begin{cases} \alpha |\mathbf{X}_{PeakDecayNrg}(m, n-1)|^2 & , |\mathbf{X}(m, n)| < \alpha |\mathbf{X}_{PeakDecayNrg}(m, n-1)| \\ |\mathbf{X}(m, n)|^2 & , otherwise \end{cases}$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$, $1 \leq n < NUM_OF_SUBSAMPLES$

α is the peak decay factor defined in table I.19.

Subsequently, filter the power and peak decay power signals with the Z-domain transfer function, $H_{Smooth}(z)$:

$$\mathbf{X}_{SmoothNrg}(z, m) = H_{Smooth}(z) \mathbf{X}_{Nrg}(z, m),$$

$$\mathbf{X}_{SmoothPeakDecayDiffNrg}(z, m) = H_{Smooth}(z) (\mathbf{X}_{PeakDecayNrg}(z, m) - \mathbf{X}_{Nrg}(z, m))$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$, where:

$$H_{Smooth}(z) = \frac{a_{Smooth}}{1 + (a_{Smooth} - 1)z^{-1}},$$

$$\mathbf{X}_{Nrg}(m, n) = |\mathbf{X}(m, n)|^2$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$ and $0 \leq n < NUM_OF_SUBSAMPLES$

The transient attenuator, $\mathbf{G}_{TransientRatio}$ is then calculated as follows:

$$\mathbf{G}_{TransientRatio}(m, n) = \begin{cases} \frac{\mathbf{X}_{Nrg}(m, n)}{\gamma \mathbf{X}_{SmoothPeakDecayDiffNrg}(m, n)}, & \gamma \mathbf{X}_{SmoothPeakDecayDiffNrg}(m, n) > \mathbf{X}_{Nrg}(m, n) \\ 1, & \text{otherwise} \end{cases}$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$, $0 \leq n < NUM_OF_SUBSAMPLES$

where $\gamma = 1,5$ is a transient duck impact factor.

Add ambience to output signals, \mathbf{Y}_{Left} and \mathbf{Y}_{Right}

Finally, the SA process adds the gain-corrected ambience signal to the side signal where \mathbf{Y}_{Left} and \mathbf{Y}_{Right} are the output matrices of the SA process for the left and right channel, respectively. If SA is enabled by signalling the bitstream element $bs_enable_sa = 1$, the SA output should be:

$$\mathbf{Y}_{Left}(m, n) = \mathbf{G}_{SaDirMapped}(m, n) \cdot \mathbf{X}(m, n) + \mathbf{G}_{TransientRatio}(m, n) \cdot \mathbf{G}_{SaMapped}(m, n) \cdot \mathbf{S}_{Ambience}(m, n)$$

$$\mathbf{Y}_{Right}(m, n) = \mathbf{G}_{SaDirMapped}(m, n) \cdot \mathbf{X}(m, n) - \mathbf{G}_{TransientRatio}(m, n) \cdot \mathbf{G}_{SaMapped}(m, n) \cdot \mathbf{S}_{Ambience}(m, n)$$

for $0 \leq m < \mathbf{f}_{SaTable}(N_{SaBands})$, $0 \leq n < NUM_OF_SUBSAMPLES$

Set some qmf bands to mono for the region not covered by SA:

$$\mathbf{Y}_{Left}(m, n) = \mathbf{X}(m, n)$$

$$\mathbf{Y}_{Right}(m, n) = \mathbf{X}(m, n)$$

for $\mathbf{f}_{SaTable}(N_{SaBands}) \leq m < NUM_OF_QMF_CHANNELS$, $0 \leq n < NUM_OF_SUBSAMPLES$

otherwise (if $bs_enable_sa = 0$) SA is bypassed and hence:

$$\mathbf{Y}_{Left} = \mathbf{X} \text{ and}$$

$$\mathbf{Y}_{Right} = \mathbf{X}$$

5.7.4.4 Pan decoding

As can be seen in figure 16, the Pan processing block takes the input signals \mathbf{Y}_{Left} and \mathbf{Y}_{Right} , which are the output signals from the SA decoding, along with the Pan parameters to create the Pan output signals \mathbf{Z}_{Left} and \mathbf{Z}_{Right} . Those output signals are fed to the final QMF bank synthesis as described in the SBR system.

Calculate Pan matrix, $\mathbf{G}_{PanMapped}$

The Pan factor, $\mathbf{g}_{PanMapped}$ is interpolated linearly within the frame according to.

$$\mathbf{k}_{PanMapped}(m) = \frac{\mathbf{g}_{PanMapped}(m) - \mathbf{g}'_{PanMapped}(m)}{NUM_OF_SUBSAMPLES}, 0 \leq m < \mathbf{f}_{PanTable}(N_{PanBands})$$

where $'$ denotes the values from the previous frame. $\mathbf{k}_{PanMapped}(m)$ is the slope used for the interpolation. The interpolated Pan matrix, $\mathbf{G}_{PanMapped}$ is then calculated according to:

$$\mathbf{G}_{PanMapped}(m, n) = \mathbf{g}'_{PanMapped}(m) + n \cdot \mathbf{k}_{PanMapped}(m), \begin{cases} 0 \leq m < \mathbf{f}_{PanTable}(N_{PanBands}) \\ 0 \leq n < NUM_OF_SUBSAMPLES \end{cases}$$

Apply Pan matrix, $\mathbf{G}_{PanMapped}$ to get \mathbf{Z}_{Left} and \mathbf{Z}_{Right}

Finally, the Pan process gain adjusts the signal taken from the SA output signal where \mathbf{Z}_{Left} and \mathbf{Z}_{Right} becomes the output matrices of the Pan process for the left and right channel, respectively. If Pan is enabled by signalling the bitstream element $bs_enable_pan = 1$, the Pan output should be:

$$\mathbf{Z}_{Left}(m, n) = \frac{2 \cdot 2^{\mathbf{G}_{PanMapped}(m, n)}}{1 + 2^{\mathbf{G}_{PanMapped}(m, n)}} \cdot \mathbf{Y}_{Left}(m, n),$$

$$\mathbf{Z}_{Right}(m, n) = \frac{2}{1 + 2^{\mathbf{G}_{PanMapped}(m, n)}} \cdot \mathbf{Y}_{Left}(m, n)$$

for $0 \leq m < \mathbf{f}_{PanTable}(N_{PanBands})$, $0 \leq n < NUM_OF_SUBSAMPLES$

otherwise (if $bs_enable_pan = 0$) Pan is bypassed and hence:

$$\mathbf{Z}_{Left} = \mathbf{Y}_{Left} \text{ and}$$

$$\mathbf{Z}_{Right} = \mathbf{Y}_{Right}$$

6 Multiplex definition

6.1 Introduction

The DRM transmission super frame consists of three channels: the Main Service Channel (MSC), the Fast Access Channel (FAC), and the Service Description Channel (SDC). The MSC contains the data for the services. The FAC provides information on the channel width and other such parameters and also provides service selection information to allow for fast scanning. The SDC gives information on how to decode the MSC, how to find alternative sources of the same data, and gives the attributes of the services within the multiplex. It can include links to analogue simulcast services.

6.2 Main Service Channel (MSC)

6.2.1 Introduction

The Main Service Channel (MSC) contains the data for all the services contained in the DRM multiplex. The multiplex may contain between one and four services, and each service may be either audio or data. The gross bit rate of the MSC is dependent upon the DRM channel bandwidth and the transmission mode.

6.2.2 Structure

The MSC contains between one and four streams. Each stream is divided into logical frames each 400 ms long. Audio streams comprise compressed audio and optionally they can carry text messages. Data streams may be composed of up to four "sub-streams" consisting of data packets. A sub-stream carries packets for one service. An audio service comprises one audio stream and optionally one data stream or one data sub-stream. A data service comprises one data stream or one data sub-stream.

Each logical frame generally consists of two parts, each with its own protection level. The lengths of the two parts are independently assigned. Unequal error protection for a stream is provided by setting different protection levels to the two parts. The logical frames from all the streams are mapped together to form multiplex frames of 400 ms duration which are passed to the channel coder. Alternatively, the first stream may be carried in logical frames mapped in to hierarchical frames.

The multiplex configuration is signalled using the SDC. The multiplex may be reconfigured at transmission super frame boundaries.

Annex M contains some examples of different MSC configurations.

6.2.3 Building the MSC

The MSC consists of a sequence of multiplex frames, and if hierarchical modulation is in use a sequence of hierarchical frames also. The multiplex frames and hierarchical frames are passed separately to the channel coder.

6.2.3.1 Multiplex frames

The multiplex frames are built by placing the logical frames from each non-hierarchical stream together. The logical frames consist, in general, of two parts each with a separate protection level. The multiplex frame is constructed by taking the data from the higher protected part of the logical frame from the lowest numbered stream (stream 0 when hierarchical modulation is not used, or stream 1 when hierarchical modulation is used) and placing it at the start of the multiplex frame. Next the data from the higher protected part of the logical frame from the next lowest numbered stream is appended and so on until all streams have been transferred. The data from the lower protected part of the logical frame from the lowest numbered stream (stream 0 when hierarchical modulation is not used, or stream 1 when hierarchical modulation is used) is then appended, followed by the data from the lower protected part of the logical frame from the next lowest numbered stream, and so on until all streams have been transferred. The higher protected part is designated part A and the lower protected part is designated part B in the multiplex description.

The multiplex frame is larger than or equal to the sum of the logical frames from which it is formed. The remainder, if any, of the multiplex frame shall be filled with 0s. These bits shall be ignored by the receiver.

NOTE: No padding bits are inserted between the end of part A and the beginning of part B. The capacity of part A of the multiplex frame is equal to the sum of the higher protected parts of the logical frames, but as a result of restrictions introduced by the channel encoding procedure applied for DRM (see clause 7.2.1.1), some of the bits nominally belonging to the lower protected part B of a multiplex frame might in fact be protected at the higher level.

6.2.3.2 Hierarchical frames

The hierarchical frames only exist when hierarchical modulation is used. They are built by taking the data from the logical frame from stream 0 and placing it at the start of the hierarchical frame.

The hierarchical frame is larger than or equal to the logical frame from which it is formed. The remainder, if any, of the hierarchical frame shall be filled with 0s. These bits shall be ignored by the receiver.

6.2.4 Reconfiguration

A reconfiguration of the multiplex occurs when the channel parameters in the FAC are changed, or when the services in the multiplex are reorganized. The new configuration is signalled ahead of time in the SDC and the timing is indicated by the reconfiguration index in the FAC. Clause 6.4.6 describes the signalling of a reconfiguration.

6.3 Fast Access Channel (FAC)

6.3.1 Introduction

The FAC is used to provide service selection information for fast scanning. It contains information about the channel parameters (for example the spectrum occupancy and interleaving depth) such that a receiver is able to begin to decode the multiplex effectively. It also contains information about the services in the multiplex to allow the receiver to either decode this multiplex or change frequency and search again.

6.3.2 Structure

Each transmission frame contains an FAC block. An FAC block contains parameters that describe the channel and parameters to describe one service along with a CRC. When more than one service is carried in the multiplex, a number of FAC blocks are required to describe all the services.

6.3.3 Channel parameters

The channel parameters are as follows:

- Base/Enhancement flag 1 bit
- Identity 2 bits
- Spectrum occupancy 4 bits
- Interleaver depth flag 1 bit
- MSC mode 2 bits
- SDC mode 1 bit
- Number of services 4 bits
- Reconfiguration index 3 bits
- Rfu 2 bits

The following definitions apply:

Base/Enhancement flag: this 1-bit flag indicates whether the transmission is the base or enhancement layer as follows:

0: Base layer - decodable by all DRM receivers.

1: Enhancement layer - only decodable by receivers with enhancement layer capabilities.

Identity: this 2-bit field identifies the current frame and also validates the SDC AFS index (see clause 6.4) as follows:

- 00: first FAC of the transmission super frame and AFS index is valid.
- 01: second FAC of the transmission super frame.
- 10: third FAC of the transmission super frame.
- 11: first FAC of the transmission super frame and AFS index is invalid.

Spectrum occupancy: this 4-bit field specifies the configuration of the digital signal. See clause 8.

Interleaver depth flag: this 1-bit flag indicates the depth of the time interleaving as follows:

- 0: 2 s (long interleaving).
- 1: 400 ms (short interleaving).

MSC mode: this 2-bit field indicates the modulation mode in use for the MSC as follows:

- 00: 64-QAM, no hierarchical.
- 01: 64-QAM, hierarchical on I.
- 10: 64-QAM, hierarchical on I&Q.
- 11: 16-QAM, no hierarchical.

SDC mode: this 1-bit field indicates the modulation mode in use for the SDC as follows:

- 0: 16-QAM.
- 1: 4-QAM.

Number of services: this 4-bit field indicates the number of audio and data services as follows:

- 0000: 4 audio services.
- 0001: 1 data service.
- 0010: 2 data services.
- 0011: 3 data services.
- 0100: 1 audio service.
- 0101: 1 audio service and 1 data service.
- 0110: 1 audio service and 2 data services.
- 0111: 1 audio service and 3 data services.
- 1000: 2 audio services.
- 1001: 2 audio services and 1 data service.
- 1010: 2 audio services and 2 data services.
- 1011: reserved.
- 1100: 3 audio services.
- 1101: 3 audio services and 1 data service.
- 1110: reserved.
- 1111: 4 data services.

Reconfiguration index: this 3-bit field indicates the status and timing of a multiplex reconfiguration. A non-zero value indicates the number of transmission super frames of the old configuration that are transmitted before the new configuration takes effect, see clause 6.4.6.

Rfu: these 2 bits are reserved for future use and shall be set to zero until they are defined.

6.3.4 Service parameters

The service parameters are as follows:

- Service identifier 24 bits
- Short identifier 2 bits
- Audio CA indication 1 bit
- Language 4 bits
- Audio/Data flag 1 bit
- Service descriptor 5 bits
- Data CA indication 1 bit
- Rfa 6 bits

The following definitions apply:

Service identifier: this 24-bit field indicates the unique identifier for this service.

Short Id: this 2-bit field indicates the short identifier assigned to this service and used as a reference in the SDC. The Short Id is assigned for the duration of the service and is maintained through multiplex reconfigurations.

Audio CA indication: this 1-bit flag indicates whether the service uses conditional access as follows:

0: No CA system is used for the audio stream (or the service has no audio stream).

1: CA system is used for the audio stream.

NOTE 1: The details are provided by the SDC data entity type 2.

Every DRM receiver shall check the "Audio CA indication" bit before presenting the audio stream of the audio service. A non-CA capable DRM receiver shall not try to decode the audio stream if the "Audio CA indication" is set to 1.

Language: this 4-bit field indicates the language of the target audience as defined in table 51.

NOTE 2: Further languages are also indicated by SDC data entity type 12.

Audio/Data flag: this 1-bit flag indicates whether the service is audio or data as follows:

0: Audio service.

1: Data service.

Service descriptor: this 5-bit field depends upon the value of the Audio/Data flag as follows:

0: Programme type.

1: Application identifier.

Regardless of the value of the Audio/Data flag, the value 31 (all bits set to 1) indicates that a standard DRM receiver should skip this broadcast and continue to scan for services.

NOTE 3: This is to allow for engineering test transmissions to be ignored by standard receivers.

Programme type: this 5-bit field indicates the programme type of an audio service as defined in table 52.

Application identifier: this 5-bit field indicates the application identifier of a data service as defined in TS 101 968 [8].

Data CA indication: this 1-bit flag indicates whether the service uses conditional access as follows:

0: No CA system is used for the data stream/sub-stream (or the service has no data stream/sub-stream).

1: CA system is used for the data stream/sub-stream.

NOTE 4: The details are provided by the SDC data entity type 2.

Every DRM receiver shall check the "Data CA indication" bit before presenting the data stream/sub-stream of the audio or data service. A non-CA capable DRM receiver shall not try to decode the data stream/sub-stream if the "Data CA indication" is set to 1.

Rfa: these 6 bits are reserved for future additions and shall be set to zero until they are defined.

Table 51: Language codes

Decimal number	Language	Decimal number	Language
0	No language specified	8	Hindi
1	Arabic	9	Japanese
2	Bengali	10	Javanese
3	Chinese (Mandarin)	11	Korean
4	Dutch	12	Portuguese
5	English	13	Russian
6	French	14	Spanish
7	German	15	Other language

Table 52: Programme type codes

Decimal number	Programme type	Decimal number	Programme type
0	No programme type	16	Weather/meteorology
1	News	17	Finance/Business
2	Current Affairs	18	Children's programmes
3	Information	19	Social Affairs
4	Sport	20	Religion
5	Education	21	Phone In
6	Drama	22	Travel
7	Culture	23	Leisure
8	Science	24	Jazz Music
9	Varied	25	Country Music
10	Pop Music	26	National Music
11	Rock Music	27	Oldies Music
12	Easy Listening Music	28	Folk Music
13	Light Classical	29	Documentary
14	Serious Classical	30	<i>Not used</i>
15	Other Music	31	<i>Not used - skip indicator</i>

The **data field** carries a variable number of data entities. It may contain an end marker and padding. The length of the data field depends upon the robustness mode, SDC mode and spectrum occupancy, and is given in table 54.

Table 54: Length of SDC data field

Robustness mode	SDC mode	Length of data field (bytes)					
		Spectrum occupancy					
		0	1	2	3	4	5
A	0	37	43	85	97	184	207
	1	17	20	41	47	91	102
B	0	28	33	66	76	143	161
	1	13	15	32	37	70	79
C	0	-	-	-	68	-	147
	1	-	-	-	32	-	72
D	0	-	-	-	33	-	78
	1	-	-	-	15	-	38

The **CRC** (Cyclic Redundancy Check) field shall contain a 16-bit CRC calculated over the AFS index coded in an 8-bit field (4 msbs are 0) and the data field. It shall use the generator polynomial $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$. See annex D.

6.4.3 Data entities

The data field is filled with data entities. Every data entity has a 12-bit header and a variable length body. The header has the following format:

- length of body 7 bits.
- version flag 1 bit.
- data entity type 4 bits.

The following definitions apply:

The **length of body** gives the number of whole bytes occupied by the data entity body.

The **version flag** controls the management of data in the receiver.

The **data entity type** is a number that determines the identity of the data entity.

The version flag allows three different mechanisms to control data management in the receiver, as specified below. The actual mechanism used is specified for each data entity.

- Reconfiguration:** For data entities using this mechanism, the version flag indicates whether the data is for the current (= 0) or next (= 1) configuration.
- List:** For data entities using this mechanism, the version flag indicates the version of the list. When any of the data in the list changes, the flag is inverted and the existing data in the receiver is discarded. The version flag applies to all the data delivered using the data entity type.
- Unique:** For data entities using this mechanism, the version flag has no meaning and shall be set to 0. These data entities carry data that is unique and therefore do not require any change mechanism.

The body of the data entities shall be at least 4 bits long. The length of the body, excluding the initial 4 bits, shall be signalled by the header.

When there is space remaining in the data field, a data end marker shall be sent. The data end marker shall be 0x00. Any remaining space shall be filled with padding. The padding shall take the value 0x00.

6.4.3.1 Multiplex description data entity - type 0

Each SDC block should contain a multiplex description entity. This data entity uses the reconfiguration mechanism for the version flag. The current configuration can always be signalled. During a reconfiguration (i.e. when the FAC reconfiguration index is non-zero) the next configuration shall be signalled. This data entity describes the multiplex of streams within the MSC. The information is as follows:

- protection level for part A 2 bits.
- protection level for part B 2 bits.
- stream description for stream 0 24 bits.

and optionally, dependent upon the number of streams in the multiplex:

- stream description for stream 1 24 bits.
- stream description for stream 2 24 bits.
- stream description for stream 3 24 bits.

The stream description for stream 0 depends on whether the MSC mode field of the FAC indicates that the hierarchical frame is present or not.

If the hierarchical frame is not present then the stream description is as follows:

- data length for part A 12 bits.
- data length for part B 12 bits.

If the hierarchical frame is present then the stream description is as follows:

- protection level for hierarchical 2 bits.
- rfu 10 bits.
- data length for hierarchical 12 bits.

The stream descriptions for streams 1, 2 and 3, when present, are as follows:

- data length for part A 12 bits.
- data length for part B 12 bits.

The following definitions apply:

protection level for part A: this field gives the overall coding rate for data in part A (see clause 7.5.1).

protection level for part B: this field gives the overall coding rate for data in part B (see clause 7.5.1).

data length for part A: this field gives the net length of data in bytes in part A of the MSC frame used by this stream.

data length for part B: this field gives the net length of data in bytes in part B of the MSC frame used by this stream.

protection level for hierarchical: this field gives the overall coding rate for data in the hierarchical frame (see clause 7.5.1).

rfu: these 10 bits shall be reserved for future use by the stream description field and shall be set to zero until they are defined.

data length for hierarchical: this field gives the net length of data in bytes in the hierarchical part of the MSC frame used by this stream.

When equal error protection is allocated to the multiplex frame (i.e. only one protection level is used) then the data length for the part A fields shall be set to 0 and the protection level for part A fields shall be set to 0.

When unequal error protection is allocated to the multiplex frame then part A is the higher protected part and part B is the lower protected part.

NOTE 1: If more than one service is carried in the multiplex, a service may be carried in both parts (some data in part A and some data in part B), or it may be carried only in one part (part A or part B). In this way, different services can be transported using unequal error protection, equal error protection at the higher level or equal error protection at the lower level in the same multiplex.

NOTE 2: The receiver may determine the number of streams present in the multiplex by dividing the value of the length field of the header by three.

6.4.3.2 Label data entity - type 1

Services may be labelled. The label should be sent in every SDC block to enable fast display, although for data services the repetition rate can be lowered. This data entity uses the unique mechanism for the version flag. The information is as follows:

- Short Id 2 bits.
- rfu 2 bits.
- label n bytes.

The following definitions apply:

Short Id: this field contains the short Id that relates the information to the Service Id provided by the FAC.

rfu: these 2 bits are reserved for future use and shall be set to zero until they are defined.

label: this is a variable length field of up to 64 bytes containing character data for up to 16 characters using UTF-8 coding (ISO/IEC 10646-1 [4]).

NOTE: The length of the label (in bytes) is given by the length field of the header.

6.4.3.3 Conditional access parameters data entity - type 2

This data entity allows the conditional access parameters to be sent. This data entity uses the reconfiguration mechanism for the version flag.

- Short Id 2 bits.
- Audio CA flag 1 bit.
- Data CA flag 1 bit.
- CA system identifier 8 bits.
- CA system specific information n bytes.

The following definitions apply:

Short Id: this field contains the short Id that relates the information to the Service Id provided by the FAC.

Audio CA flag: this 1-bit flag indicates whether the conditional access parameters refer to an audio stream as follows:

0: Parameters do not refer to an audio stream.

1: Parameters refer to an audio stream.

NOTE 1: In case of a data service this flag shall be 0.

Data CA flag: this 1-bit flag indicates whether the conditional access parameters refer to a data stream/sub-stream as follows:

0: Parameters do not refer to a data stream/sub-stream.

1: Parameters refer to a data stream/sub-stream.

NOTE 2: In case of an audio service that does not have a data stream/sub-stream this flag shall be 0.

rfu: these 2 bits are reserved for future use and shall be set to zero until they are defined.

CA system identifier: this field indicates the CA system used by this service.

CA system specific information: this is a variable length field containing CA system specific data.

NOTE 3: An audio service can have a scrambled audio stream and a scrambled data stream/sub-stream and the conditional access parameters can be different for each. In this case two *Conditional access parameters data entity - type 2* are needed. If the audio stream and the data stream/sub-stream use identical conditional access parameters then one *Conditional access parameters data entity - type 2* is sufficient; both the Audio CA flag and the Data CA flag are set to 1.

6.4.3.4 Alternative frequency signalling: Multiple frequency network information data entity - type 3

This data entity is used to provide receivers with information about the DRM frequencies on which:

- the whole multiplex of services can be found;
- some of the services of the multiplex can be found;
- the base/enhancement layer of the whole multiplex of services can be found;
- the base/enhancement layer of some of the services of the multiplex can be found.

It is also used to provide receivers with information as to whether the frequencies can be used for seamless alternate frequency checking and switching, see annex G.

This data entity uses the list mechanism for the version flag.

NOTE 1: If the same audio programme or data application is broadcast using a different DRM service identifier, or if it is broadcast using another broadcast system, then data entity type 11 is used to indicate the frequency.

The frequencies may be restricted to certain times and/or geographic areas in combination with data entities type 4 and 7 respectively.

The information is as follows:

- Synchronous Multiplex flag 1 bit.
- Layer flag 1 bit.
- Service Restriction flag 1 bit.
- Region/Schedule flag 1 bit.
- Service Restriction field 0 or 8 bits.
- Region/Schedule field 0 or 8 bits.
- n frequencies $n \times 16$ bits.

Region Id: this field indicates whether the region is unspecified (value 0) or whether the alternative frequencies are valid just in certain geographic areas, in which case it carries the Region Id (value 1 to 15). The region may be described by one or more "Alternative frequency signalling: Region definition data entity - type 7" with this Region Id.

Schedule Id: this field indicates whether the schedule is unspecified (value 0) or whether the alternative frequencies are valid just at certain times, in which case it carries the Schedule Id (value 1 to 15). The schedule is described by one or more "Alternative frequency signalling: Schedule definition data entity - type 4" with this Schedule Id.

***n* frequencies:** this field carries *n* 16 bit fields. *n* is in the range 1 to 16. Each 16 bit field contains the following information:

- rfu 1 bit.
- frequency value 15 bits.

NOTE 2: The number of frequencies, *n*, is determined from the length field of the header and the value of the Service Restriction flag and the Region/Schedule flag.

rfu: this 1 bit field is reserved for future use of the frequency value field and shall be set to zero until defined.

frequency value: this 15 bit field is coded as an unsigned integer and gives the frequency in kHz.

rfa: these 1-bit or 4-bit fields (if present) are reserved for future additions and shall be set to zero until they are defined.

Additional information on Alternative Frequency Signalling is provided in annexes F and G.

6.4.3.5 Alternative frequency signalling: Schedule definition data entity - type 4

This entity allows a frequency schedule to be transmitted. This data entity uses the list mechanism for the version flag. This information is as follows:

- Schedule Id 4 bits
- Day Code 7 bits
- Start Time 11 bits
- Duration 14 bits

The following definitions apply:

Schedule Id: this field indicates the Schedule Id for the defined schedule. Up to 15 different schedules with an individual Schedule Id (values 1 to 15) can be defined; the value 0 shall not be used, since it indicates "unspecified schedule" in data entity type 3 and 11.

Day Code: this field indicates which days the frequency schedule (the following Start Time and Duration) applies to. The msb indicates Monday, the lsb Sunday. Between one and seven bits may be set to 1.

Start Time: this field indicates the time from when the frequency is valid. The time is expressed in minutes since midnight UTC. Valid values range from 0 to 1 439 (representing 00:00 to 23:59).

Duration: this field indicates how long the frequency is valid starting from the indicated Start Time. The time is expressed in minutes. Valid values range from 1 to 16 383.

NOTE: The Duration may signal a time interval of more than one week. See Annex O for interpretation rules and examples.

Additional information on Alternative Frequency Signalling is provided in annexes F and G.

data unit indicator: this field indicates whether the data stream is composed of single packets or data units as follows:

0: single packets.

1: data units.

packet Id: this field indicates the Packet Id carried in the header of packets intended for this service.

enhancement flag: this field indicates whether enhancement data is available in another channel as follows:

0: no enhancement available.

1: enhancement is available.

application domain: this field indicates the source of the data application specification. The interpretation of this field is given in TS 101 968 [8].

packet length: this field indicates the length in bytes of the data field of each packet specified as an unsigned binary number (the total packet length is three bytes longer as it includes the header and CRC fields).

NOTE: All packets contained in one data stream shall have the same length (see clause 6.6.4).

application data: this field of variable length is defined by the data service (or data application) specification. The interpretation of this field is given in TS 101 968 [8].

6.4.3.7 Announcement support and switching data entity - type 6

This data entity indicates which types of announcements are supported by the tuned multiplex or by another DRM multiplex/another broadcast system. It also indicates which of the services of the tuned multiplex should switch in case of an active announcement.

This data entity uses the list mechanism for the version flag. The version flag status shall remain the same if only the "Announcement switching flags" are changed. The version flag shall be inverted in case of changes to any other announcement information.

The following information is necessary:

- Short Id flags 4 bits.
- Same Multiplex/Other Service flag 1 bit.
- Short Id/Announcement Id 2 bits.
- rfa 1 bit.
- Announcement support flags 10 bits.
- Announcement switching flags 10 bits.

The following definitions apply:

Short Id flags: this 4 bit field indicates to which services (identified by their Short Id) of the tuned DRM multiplex the announcement definition applies to by setting the corresponding bit to 1. The first bit (msb) refers to Short Id 3, while the last bit (lsb) refers to Short Id 0 of the tuned DRM multiplex.

Same Multiplex/Other Service flag: this field indicates if the announcements are carried in the tuned DRM multiplex or not, as follows:

0: announcements are carried in the tuned multiplex.

1: announcements are carried elsewhere.

Short Id/Announcement Id: the content of this 2 bit field depends upon the value of the Same Multiplex/Other Service flag as follows:

Same Multiplex/Other Service flag = 0:

- Short Id 2 bits.

Same Multiplex/Other Service flag = 1:

- Announcement Id 2 bits.

Short Id: this field signals the Short Id of the service within the tuned DRM multiplex which carries the announcement content.

Announcement Id: this field carries the Announcement Id (value 0 to 3). The Other Service Ids (and potentially frequencies) carrying the programme with the announcement content are described by one or multiple "Alternative frequency signalling: Other services data entity - type 11" (with the "Short Id/Announcement Id flag" bit set to 1, indicating that the Other Service Id and/or the list of frequencies belong to a programme carrying announcement content).

rfa: this 1 bit is reserved for future additions and shall be set to zero until it is defined.

Announcement support flags: this 10-bit field specifies the types of announcements that are described by this data entity and provided either by one service in the tuned DRM multiplex or by another service on another frequency as follows:

B_i ($i = 0, \dots, 9$).

0: Announcement type not provided.

1: Announcement type provided.

The meaning of each bit is as follows:

b_0 : Travel.

b_1 : News flash.

b_2 : Weather flash.

b_3 : Warning/Alarm.

$b_4 - b_9$: reserved for future definition.

Announcement switching flags: the individual bits of this 10-bit field indicate whether or not a particular announcement type is currently active. The flags are coded as follows:

B_i ($i = 0, \dots, 9$).

0: Announcement type not valid (currently not active).

1: Announcement type valid (currently active).

The meaning of each bit is as defined for the Announcement support flags above.

Additional information on announcements is provided in annex F.

6.4.3.8 Alternative frequency signalling: Region definition data entity - type 7

This entity allows the definition of geographic areas for which a set of alternative frequencies is provided. This data entity uses the list mechanism for the version flag.

The following definitions apply:

Modified Julian Date: this field indicates the date in MJD format.

UTC: this field specifies the current UTC time expressed in hours (5 bits) and minutes (6 bits).

When the time and date are signalled, this data entity shall be sent in the first SDC block after the minute's edge.

6.4.3.10 Audio information data entity - type 9

Each audio service needs a detailed description of the parameters needed for audio decoding. This data entity uses the reconfiguration mechanism for the version flag.

- Short Id 2 bits.
- Stream Id 2 bits.
- audio coding 2 bits.
- SBR flag 1 bit.
- audio mode 2 bits.
- audio sampling rate 3 bits.
- text flag 1 bit.
- enhancement flag 1 bit.
- coder field 5 bits.
- rfa 1 bit.

The following definitions apply:

short Id: this field indicates the short Id for the service concerned.

Stream Id: this field indicates the stream Id of the stream that carries the service concerned.

audio coding: this field indicated the source coding system as follows:

- 00: AAC.
- 01: CELP.
- 10: HVXC.
- 11: reserved.

SBR flag: this field indicates whether SBR is used or not as follows:

- 0: SBR not used.
- 1: SBR used.

audio mode: this field depends upon the value of the audio coding field and SBR flag as follows:

audio coding field = 00 (AAC):

- 00: mono.
- 01: parametric stereo.
- 10: stereo.
- 11: reserved.

audio coding field = 01 (CELP) and SBR flag = 0:

- rfa 1 bit.
- CELP_CRC 1 bit.

audio coding field = 01 (CELP) and SBR flag = 1:

- sbr_header_flag 1 bit.
- CELP_CRC 1 bit.

audio coding field = 10 (HVXC):

- HVXC_rate 1 bit.
- HVXC_CRC 1 bit.

CELP_CRC: this field indicates whether the CRC is used or not:

0: CRC not used.

1: CRC used.

sbr_header_flag: this field indicates whether an SBR header is present in the audio super frame or not:

0: header absent, use predefined SBR configuration as given by annex I.

1: header present, derive SBR configuration from header in SBR bitstream.

HVXC_rate: this field indicates the rate of the HVXC:

0: 2 kbit/s.

1: 4 kbit/s.

HVXC_CRC: this field indicates whether the CRC is used or not:

0: CRC not used.

1: CRC used.

audio sampling rate: this field indicates the audio sampling rate as follows:

000: 8 kHz.

001: 12 kHz.

010: 16 kHz.

011: 24 kHz.

100: reserved.

101: reserved.

110: reserved.

111: reserved.

text flag: this field indicates whether a text message is present or not as follows:

0: no text message is carried.

1: text message is carried (see clause 6.5).

enhancement flag: this field indicates whether audio enhancement data is available in another channel as follows:

0: no enhancement available.

1: enhancement is available.

coder field: this field depends upon the value of the audio coding field and SBR flag as follows:

audio coding field = 00 (AAC):

- rfa 5 bits.

audio coding field = 01 (CELP):

- CELP_index 5 bits.

audio coding field = 10 (HVXC) and SBR flag = 0:

- rfa 5 bits.

audio coding field = 10 (HVXC) and SBR flag = 1:

- sbr_header_flag 1 bit.

if sbr_header_flag = 0:

- sbr_hvxc_preset 4 bits.

if sbr_header_flag = 1:

- rfa 4 bits.

CELP_index: this field indicates the CELP bit rate index, as defined in tables 10 and 11 (see clause 5.4.1.1).

sbr_header_flag: this field indicates whether an SBR header is present in the audio super frame or not:

0: header absent, use predefined SBR configuration as given by annex I.

1: header present, derive SBR configuration from header in SBR bitstream.

sbr_hvxc_preset: this field indicates which SBR preset to use at a given bitrate as given by clause I.2.

rfa: these 1-bit, 4-bit and 5-bit fields are reserved for future additions and shall be set to zero until they are defined.

6.4.3.11 FAC channel parameters data entity - type 10

This data entity permits the next configuration FAC channel parameters to be specified in advance for service following across reconfigurations. This data entity uses the reconfiguration mechanism for the version flag. The fields are as follows:

- Base/Enhancement flag 1 bit.
- Robustness mode 2 bits.
- Spectrum occupancy 4 bits.
- Interleaver depth flag 1 bit.
- MSC mode 2 bits.
- SDC mode 1 bit.
- number of services 4 bits.
- rfa 3 bits.
- rfu 2 bits.

The following definitions apply:

Base/Enhancement flag: this 1-bit flag indicates whether the transmission is the base or enhancement layer as follows:

0: Base layer - decodable by all DRM receivers.

1: Enhancement layer - only decodable by receivers with enhancement layer capabilities.

Robustness mode: this field indicates the robustness mode of the new configuration as follows:

00: A.

01: B.

10: C.

11: D.

Spectrum occupancy: specifies the configuration of the digital signal (see clause 8).

Interleaver depth flag: indicates the depth of the time interleaving as follows:

0: 2 s (long interleaving).

1: 400 ms (short interleaving).

MSC mode: indicates the modulation mode in use for the MSC as follows:

00: 64-QAM, no hierarchical.

01: 64-QAM, hierarchical on I.

10: 64-QAM, hierarchical on I&Q.

11: 16-QAM, no hierarchical.

SDC mode: indicates the modulation mode in use for the SDC as follows:

0: 16-QAM.

1: 4-QAM.

Number of services: indicates the number of audio and data services as follows:

0000: 4 audio services.

0001: 1 data service.

0010: 2 data services.

0011: 3 data services.

0100: 1 audio service.

0101: 1 audio service and 1 data service.

0110: 1 audio service and 2 data services.

0111: 1 audio service and 3 data services.

1000: 2 audio services.

1001: 2 audio services and 1 data service.

1010: 2 audio services and 2 data services.

1011: reserved.

1100: 3 audio services.

1101: 3 audio services and 1 data service.

1110: reserved.

1111: 4 data services.

rfa: these 3 bits are reserved for future additions and shall be set to zero until they are defined.

rfu: these 2 bits are reserved for future use and shall be set to zero until they are defined.

If the DRM transmission is being discontinued at the reconfiguration, then this data entity shall be sent with the length field of the header set to 0, and the first four bits of the body field set to 0.

6.4.3.12 Alternative frequency signalling: Other services data entity - type 11

For every service of the tuned DRM multiplex alternative sources can be signalled. These alternative sources can be DRM services (using different DRM service identifiers) or services on other broadcast systems like AM, FM, FM-RDS or DAB. For every service of the tuned DRM multiplex, this type of data entity lists the corresponding service identifier within a DRM multiplex or another broadcast system, optionally along with the list of alternative frequencies. It uses the list mechanism for the version flag. At least one other service identifier or one frequency must be provided in this data entity.

NOTE: The list of alternative DRM frequencies for the entire DRM multiplex or some of its services (i.e. using the same service identifiers) can be derived from "Alternative frequency signalling: Multiple frequency network information data entity - type 3".

The alternative frequencies may be scheduled to certain times and/or restricted to certain geographic areas.

The information is as follows:

- Short Id/Announcement Id flag 1 bit.
- Short Id/Announcement Id field 2 bits.
- Region/Schedule flag 1 bit.
- Same Service flag 1 bit.
- rfa 2 bits.
- System Id 5 bits.
- Region/Schedule field 0 or 8 bits.
- Other Service Id 0 or 16 or 24 or 32 bits.
- n frequencies $n \times (8 \text{ or } 16)$ bits.

The following definitions apply:

Short Id/Announcement Id flag: this flag specifies the content of the Short Id/Announcement Id field as follows:

0: Short Id.

1: Announcement Id.

Short Id/Announcement Id field: the content of this field depends upon the value of the Short Id/Announcement Id flag as follows:

Short Id/Announcement Id flag = 0:

- Short Id 2 bits.

Short Id/Announcement Id flag = 1:

- Announcement Id 2 bits.

Short Id: this field carries the Short Id of the service in the tuned DRM multiplex to which the alternative frequencies apply.

Announcement Id: this field carries the Announcement Id (value 0 to 3). The information for which services in the tuned DRM multiplex announcements are signalled (and which type of announcements) is described by one or more "Announcement support and switching data entity - type 6" (with the "Same Multiplex/Other Service flag" bit set to 1, indicating that the announcement is provided by another service outside the tuned DRM multiplex).

Region/Schedule flag: this field indicates whether the list of frequencies is restricted by region and/or schedule or not as follows:

0: no restriction.

1: region and/or schedule applies to this list of frequencies.

Same Service flag: this flag indicates whether the specified other service should be considered the "same service" (e.g. carrying the identical audio programme) or an "alternative service" (e.g. a different audio programme from the same broadcaster offering a similar programme - see clauses F.1.1 and G.1 for details):

0: alternate service.

1: same service.

rfa: these 2 bits are reserved for future additions and shall be set to zero until they are defined.

System Id: this field indicates which broadcast system the Other Service Id and frequency information applies to as follows:

00000: DRM service

Other Service Id: 24 bits (DRM service identifier).

00001: reserved for future definition

(AM service with AM service identifier).

00010: AM service

Other Service Id: not present (AM service identifier not specified)

00011: FM-RDS service (Europe and North America grid)

Other Service Id: 24 bits (ECC+PI code).

00100: FM-RDS service (Europe and North America grid)

Other Service Id: 16 bits (PI code only).

00101: FM service (Europe and North America grid)

Other Service Id: not present (PI code not specified).

00110: FM-RDS service (Asia grid)

Other Service Id: 24 bits (ECC+PI code).

00111: FM-RDS service (Asia grid)

Other Service Id: 16 bits (PI code only).

01000: FM service (Asia grid)

Other Service Id: not present (PI code not specified).

01001: DAB service

Other Service Id: 24 bits (ECC + audio service identifier).

01010: DAB service

Other Service Id: 16 bits (audio service identifier only).

01011: DAB service

Other Service Id: 32 bits (data service identifier).

all other values: reserved for future definition.

Additional information on Alternative Frequency Signalling is provided in annexes F and G.

6.4.3.13 Language and country data entity - type 12

The language and country data entity allows additional language and country information to be signalled. This data entity uses the unique mechanism for the version flag. The information is as follows:

- Short Id 2 bits.
- rfu 2 bits.
- language code 24 bits.
- country code 16 bits.

The following definitions apply:

Short Id: this field indicates the short Id for the service concerned.

rfu: these 2 bits are reserved for future use and shall be set to zero until they are defined.

Language code: this 24-bit field identifies the language of the target audience of the service according to ISO 639-2 [5] using three lower case characters as specified by ISO/IEC 8859-1 [7]. If the language is not specified, the field shall contain three "-" characters.

Country code: this 16-bit field identifies the country of origin of the service (the site of the studio) according to ISO 3166 [6] using two lower case characters as specified by ISO/IEC 8859-1 [7]. If the country code is not specified, the field shall contain two "-" characters.

6.4.3.14 Other data entities

Other data entities are reserved for future definition.

6.4.4 Summary of data entity characteristics

Tables 56 and 57 summarize the version flag mechanism, repetition rate and transmission status of each data entity. The standard repetition rate is that all information for that data entity type should be transmitted within one cycle of the entire database. Individual SDC blocks may carry changed information (e.g. time and date) by use of the FAC identity field.

Table 56: Summary of data entity characteristics

Data entity	Name	Version flag mechanism	Repetition rate
0	Multiplex description	reconfiguration	every SDC block
1	Label	unique	every SDC block
2	Conditional Access Parameters	reconfiguration	as required
3	AFS - Multiple frequency network information	list	standard
4	AFS - Schedule definition	list	standard
5	Application information	reconfiguration	as required
6	Announcement support and switching	list	standard
7	AFS - Region definition	list	standard
8	Time and date information	unique	once per minute
9	Audio information	reconfiguration	every SDC block
10	FAC channel parameters	reconfiguration	every SDC block when FAC reconfiguration index is non-zero
11	AFS - Other services	list	standard
12	Language and country	unique	standard

Table 56 gives the recommended repetition rate for fast access to services. However, when the SDC capacity (see clause 6.4.2) is low, lower repetition rates are permitted for every data entity.

Table 57: Summary of data entity characteristics

Data entity	Transmission status	Entity occurrence and limits (normal)	Entity occurrence and limits (during reconfiguration)
0	mandatory	one entity	as (normal) for each configuration; two entities in total
1	optional	zero or one entity per service; zero to four entities in total	as (normal)
2	mandatory for each service for which the FAC CA indication flag = 1	zero, one or two entities per audio service; zero or one entity per data service; zero to seven entities in total	as (normal) for each configuration; zero to 14 entities in total
3	optional	zero to 16 frequencies per entity; zero to 64 entities in total	as (normal)
4	optional	zero to 32 entities per Schedule Id; zero to 128 entities in total	as (normal)
5	mandatory for each data service and data application	zero or one entity per audio service; one entity per data service; zero to four entities in total	as (normal) for each configuration; zero to eight entities in total
6	optional	zero to eight entities in total	as (normal)
7	optional	zero to four entities per Region Id; up to 16 CIRAF zones per Region Id; zero to 32 entities in total	as (normal)
8	optional	zero or one entity per minute	as (normal)
9	mandatory for each audio service	one entity per audio service; zero to four entities in total	as (normal) for each configuration; zero to eight entities in total
10	mandatory when FAC reconfiguration index is non-zero	zero	one entity
11	optional	zero to 16 frequencies per entity; zero to 256 entities in total	as (normal)
12	optional	zero or one entity per service; zero to four entities in total	as (normal)

Table 57 lists the transmission characteristics of each SDC entity type. The total number of entities is limited in some cases to ensure consistent receiver operation (by defining the maximum memory size needed).

6.4.5 Changing the content of the SDC

The content of the SDC is important for the operation of Alternative Frequency checking and Switching (AFS). For AFS to function, the receiver must know what the content of the SDC is in advance so that a correlation may be performed. For this purpose, the AFS index is provided in the SDC and the FAC validates the index by use of the Identity field.

On transmissions with no alternative frequencies, the content of the SDC can be fully dynamic and changed at will: no AFS function is required. In this case it is recommended that the AFS index should be set to 0, and the Identity field in the FAC should then indicate the sequence 11, 01, 10, etc. to indicate that the AFS function cannot be performed.

On transmissions with alternative frequencies, the assignment of data entities to SDC blocks should be carefully designed in order that the content of the SDC can be as static as possible thereby permitting use of the AFS function. In this case it is recommended that the AFS index is chosen such that all required information can be sent in one cycle of SDC blocks. If the content is completely static then the Identity field in the FAC indicates the sequence 00, 01, 10, etc. which indicates that the AFS function can be performed at every position, provided the receiver has stored the data for all the SDC blocks in the cycle.

When the Time and date data entity or announcement support and switching data entity is included in the SDC, and alternative frequencies are signalled, then a semi-dynamic use of the SDC is recommended. In this case one or more SDC blocks in the cycle defined by the AFS index are signalled to be invalid by use of the FAC Identity field thereby allowing the content of those blocks to be changed continuously, whilst other SDC blocks are always signalled as valid by use of the FAC Identity field thereby allowing the AFS function to be performed. An example of changing the SDC content and of using the semi-dynamic scheme with the AFS index = 1 is given in annex G.

A change of the AFS index is only allowed at reconfiguration.

6.4.6 Signalling of reconfigurations

Reconfiguration of the DRM multiplex shall be signalled in advance in order to permit receivers to make the best decisions about how to handle the changes. There two types of reconfiguration: a service reconfiguration, which concerns the reallocation of the data capacity between the services of the MSC; and a channel reconfiguration, which concerns changes to the overall capacity of the MSC.

Both types of reconfiguration are signalled by setting the FAC reconfiguration index to a non-zero value. The index then counts down on each subsequent transmission super frame. The reconfiguration index shall be identical for all three transmission frames of a transmission super frame. The final transmission super frame corresponding to the current configuration shall be that in which the reconfiguration index = 1. The new configuration takes effect for the next transmission super frame and in which the reconfiguration index = 0.

All data entity types that use the reconfiguration mechanism for the version flag that are present in the current configuration, and all data entity types that use the reconfiguration mechanism for the version flag that are required in the new configuration, shall be sent during the period when the reconfiguration index is non-zero with the version flag indicating the next configuration. This shall include data entity type 10 that signals the FAC channel parameters for the new configuration.

6.4.6.1 Service reconfigurations

A service reconfiguration is one in which the data capacity of the MSC is reallocated between services. This happens when the number of services in the multiplex is changed or the size of data streams is changed. A service reconfiguration shall also be signalled if any of the content of the data entity types using the reconfiguration mechanism of the version flag changes. The reconfiguration shall be signalled as far in advance as possible in order to provide the greatest chance that the receiver gets all the information necessary for the next configuration. Therefore the reconfiguration index shall first take the value 7.

When a new service is introduced, and the overall capacity of the MSC is not changed, then the receiver shall follow the currently selected service through the reconfiguration. To facilitate this, the Service Identity and Short Id of all continuing services shall remain the same. The new service shall use a Short Id that is not used in the current configuration. The one exception to this rule is if there are four services in the current configuration and four services in the new configuration. In this case, if the currently selected service is discontinued, then the receiver follows to the new service with the same Short Id if it is of the same type (e.g. both are audio services).

If the currently selected service is discontinued at the reconfiguration, then the receiver may try to find another source of that service on another frequency and/or system by using the information from data entity types 3 and 11.

6.4.6.2 Channel reconfigurations

A channel reconfiguration is one in which the following FAC channel parameters are altered: spectrum occupancy, interleaver depth, MSC mode; and when the robustness mode is changed. In this case the receiver is unable to follow the currently selected service without disruption to the audio output. The reconfiguration should be signalled as far in advance as possible in order to provide the greatest chance that the receiver gets all the information necessary for the next configuration. Ideally the reconfiguration index should first take the value 7, although a lower starting value may be necessary for operational reasons.

If the transmission is discontinued on the tuned frequency, then a reconfiguration shall be signalled with data entity type 10 taking a special value (see clause 6.4.3.11). In this specific case, the other data entity types that use the reconfiguration mechanism for the version flag shall not be signalled.

6.5 Text message application

Text messages can provide a highly valuable additional element to an audio service without consuming much data capacity. The text message is a basic part of DRM and consumes only 80 bits/s. This capacity can be saved if the service provider does not use text messaging.

6.5.1 Structure

The text message (when present) shall occupy the last four bytes of the lower protected part of each logical frame carrying an audio stream. The message is divided into a number of segments and UTF-8 character coding is used. The beginning of each segment of the message is indicated by setting all four bytes to the value 0xFF.

When no text message is available for insertion all four bytes shall be set to 0x00.

The text message may comprise up to 8 segments. Each segment consists of a header, a body and a CRC. The body shall contain 16 bytes of character data unless it is the last segment in which case it may contain less than 16 bytes.

Each segment is further divided into four-byte pieces which are placed into each successive frame. If the length of the last segment is not a multiple of four then the incomplete frame shall be padded with 0x00 bytes.

The structure of the segment is as follows:

- Header 16 bits.
- Body $n \times 8$ bits.
- CRC 16 bits.

The Header is made up as follows:

- toggle bit 1 bit.
- first flag 1 bit.
- last flag 1 bit.
- command flag 1 bit.
- field 1 4 bits.
- field 2 4 bits.
- rfa 4 bits.

The following definitions apply:

Toggle bit: this bit shall be maintained in the same state as long as segments from the same message are being transmitted. When a segment from a different text message is sent for the first time, this bit shall be inverted with respect to its previous state. If a text message, which may consist of several segments, is repeated, then this bit shall remain unchanged.

First flag, Last flag: these flags are used to identify particular segments which form a succession of segments in a text message. The flags are assigned as follows:

First flag	Last flag	The segment is:
0	0	: an intermediate segment;
0	1	: the last segment;
1	0	: the first segment;
1	1	: the one and only segment.

Command flag: this 1-bit flag signals whether Field 1 contains the length of the body of the segment or a special command, as follows:

0: Field 1 signals the **length** of the body of the segment.

1: Field 1 contains a special **command**.

Field 1:

- **Length:** this 4-bit field, expressed as an unsigned binary number, specifies the number of bytes in the body minus 1. It shall normally take the value 15 except in the last segment.
- **Command:** this 4-bit field contains a special command, as follows (all other codes are reserved for future use):
 - 0 0 0 1: the message shall be removed from the display.

Field 2:

- if **First flag** = "1":
 - this field contains the value "1111";
- if **First flag** = "0":
 - **Rfa:** this 1-bit field is reserved for future additions. The bit shall be set to zero until it is defined.
 - **SegNum** (Segment number): this 3-bit field, expressed as an unsigned binary number, specifies the sequence number of the current segment minus 1. (The second segment of a label corresponds to SegNum = 1, the third segment to SegNum = 2, etc.) The value 0 is reserved for future use.

Rfa: this 4-bit field is reserved for future additions. These bits shall be set to zero until they are defined.

Body: this field shall be coded as a string of characters (maximum 16). If the last character of a message segment is a multibyte character and not all bytes fit into the body then the character shall continue in the next message segment. This field shall be omitted when the C flag = "1" (special command).

The following additional codes may be used:

- Code 0x0A may be inserted to indicate a preferred line break.
- Code 0x0B may be inserted to indicate the end of a headline. Headlines shall be restricted to a maximum length of 2×16 displayable characters (including hyphens introduced as a result of a control code 0x1F) and may contain 0 or 1 preferred line breaks codes (Code 0x0A). There may not be more than sixteen characters before any line-break and there may not be more than sixteen characters after any line-break.
- Code 0x1F (hex) may be inserted to indicate a preferred word break. This code may be used to display long words comprehensibly.

Cyclic Redundancy Check (CRC): this 16-bit CRC shall be calculated on the header and the body. It shall use the generator polynomial $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$.

6.6 Packet mode

Data services generally consist of either streams of information, in either synchronous or asynchronous form, or files of information. A generalized packet delivery system allows the delivery of asynchronous streams and files for various services in the same data stream and allows the bit rate of the (synchronous) data stream to be shared on a frame-by-frame basis between the various services. Services can be carried by a series of single packets or as a series of data units. A data unit is a series of packets that are considered as one entity with regard to error handling - one received errored packet within a data unit causes the whole data unit to be rejected. This mechanism can be used to transfer files and also to allow simpler synchronization of asynchronous streams. The carriage of data applications is described in TS 101 968 [8].

The size of a packet mode data logical frame shall be a multiple of the packet size. The maximum length of a data unit is 8 215 bytes.

6.6.1 Packet structure

The packet is made up as follows:

- header 8 bits.
- data field n bytes.
- CRC 16 bits.

The **header** contains information to describe the packet.

The **data field** contains the data intended for a particular service. The length of the data field is indicated by use of data entity 5, see clause 6.4.3.6.

Cyclic Redundancy Check (CRC): this 16-bit CRC shall be calculated on the header and the data field. It shall use the generator polynomial $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$ (see annex D).

6.6.1.1 Header

The header consists of the following fields:

- first flag 1 bit.
- last flag 1 bit.
- packet Id 2 bits.
- Padded Packet Indicator (PPI) 1 bit.
- Continuity Index (CI) 3 bits.

The following definitions apply:

First flag, Last flag: these flags are used to identify particular packets which form a succession of packets. The flags are assigned as follows:

First flag	Last flag	The packet is:
0	0	: an intermediate packet;
0	1	: the last packet of a data unit;
1	0	: the first packet of a data unit;
1	1	: the one and only packet of a data unit.

Packet Id: this 2-bit field indicates the Packet Id of this packet.

Padded Packet Indicator: this 1-bit flag indicates whether the data field carries padding or not, as follows:

0: no padding is present: all data bytes in the data field are useful;

1: padding is present: the first byte gives the number of useful data bytes in the data field.

Continuity index: this 3-bit field shall increment by one modulo-8 for each packet with this packet Id.

6.6.1.2 Data field

The data field contains the useful data intended for a particular service.

If the Padded Packet Indicator (PPI) field of the header is 0, then all bytes of the data field are useful bytes.

If the PPI is 1 then the first byte indicates the number of useful bytes that follow, and the data field is completed with padding bytes of value 0x00.

Packets with no useful data are permitted if no packet data is available to fill the logical frame. The PPI shall be set to 1 and the first byte of the data field shall be set to 0 to indicate no useful data. The first and last flags shall be set to 1. The continuity index shall be incremented for these empty packets. If less than 4 sub-streams are used within the data stream then an unused packet id shall be used. Empty packets using a packet id of <p> shall not be inserted during the transmission of a DRM data unit using the same packet id <p>.

6.6.2 Asynchronous streams

Asynchronous streams can be used to transport byte-oriented information. Both single packets and data units can be used to transport asynchronous streams.

Applications that use the single packet transport mechanism shall be able to deal with missing data packets. The first and last flags indicate intermediate packets.

Applications that use the data unit transport mechanism can carry a collection of bytes that are related in a data unit and then make use of the error handling of data units for synchronization purposes.

6.6.3 Files

The file may be carried in a data unit.

Applications that use this transport mechanism shall provide a mechanism to identify each object.

The first and last flags are used to indicate the series of packets that make up the data unit. The continuity index is used to determine whether any intermediate packets have been lost.

6.6.4 Choosing the packet length

A data stream for packet mode may contain one or more packets per logical frame, and the packets may belong to one or more services. However, all packets contained in the stream shall have the same length to minimize the propagation of errors. The choice of the packet length depends on various factors, but the following should be taken into account:

- The overhead of signalling the header and CRC is fixed per packet. Therefore the larger the packet, the lower the ratio of overhead to useful data.
- The amount of padding carried in packets is related to the size of the files compared to the packet size or the transit delay requirements for asynchronous streams. Large packets are less efficient at transporting many small objects.

7 Channel coding and modulation

7.1 Introduction

The DRM system consists of three different channels, the MSC, SDC and FAC. Because of the different needs of these channels different coding and mapping schemes shall be applied. An overview of the encoding process is shown in figure 17.

The coding is based on a multilevel coding scheme for which the principle is explained in clause 7.3. Due to different error protection needs within one service or for different services within one multiplex different mapping schemes and combinations of code rates are applicable: Unequal Error Protection (UEP) and Equal Error Protection (EEP) are available and can be combined with hierarchical modulation. Equal error protection uses a single code rate to protect all the data in a channel. EEP is mandatory for the FAC and SDC. Instead of EEP, unequal error protection can be used with two code rates to allow the data in the Main Service Channel to be assigned to the higher protected part and the lower protected part. When using hierarchical modulation three mapping strategies are applicable to the MSC: the Standard Mapping (SM), the symmetrical Hierarchical Mapping (HMsym) and a mixture of the previous two mappings (HMmix) that results in the real component of the constellation following a Hierarchical Mapping and the imaginary part following a standard one. The Hierarchical Mappings split the decodable data stream into two parts: a Very Strongly Protected Part (VSPP) and a Standard Protected Part (SPP). The SM method only consists of a SPP. In any case, up to two different overall code rates shall be applied to the SPP of the MSC. For the FAC and SDC only SM is allowed. The application of the coding to the different channels is described in clause 7.5.

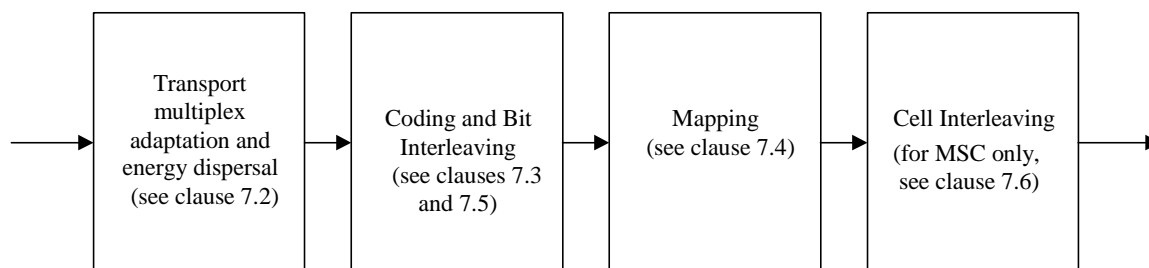


Figure 17: Functional block diagram of the coding and interleaving

7.2 Transport multiplex adaptation and energy dispersal

7.2.1 Transport multiplex adaptation

The different channels (MSC, SDC, FAC) are processed in the channel coding independently. The vector length L for processing equals one FAC block for the FAC, one SDC block for the SDC or one multiplex frame for the MSC.

7.2.1.1 MSC

The number of bits L_{MUX} per multiplex frame is dependent on the robustness mode, spectrum occupancy and constellation:

- when using one protection level (EEP) it is given by:

$$L_{MUX} = L_2$$

- when using two protection levels (UEP) it is given by:

$$L_{MUX} = L_1 + L_2$$

where the number of bits of the higher protected part is L_1 and the number of bits of the lower protected part is L_2 .

- when using HMsym or HMmix the number of very strongly protected bits is given by L_{VSPP} .

L_1 , L_2 and L_{VSPP} are calculated as follows:

SM:

$$L_1 = \sum_{p=0}^{P_{\max}-1} 2N_1 R_p$$

$$L_2 = \sum_{p=0}^{P_{\max}-1} RX_p \left\lfloor \frac{2N_2 - 12}{RY_p} \right\rfloor$$

$$L_{VSPP} = 0$$

P_{\max} is the number of levels (4-QAM: $P_{\max} = 1$; 16-QAM: $P_{\max} = 2$; 64-QAM: $P_{\max} = 3$).

RX_p is the numerator of the code rate of each individual level, see table 60.

RY_p is the denominator of the code rate of each individual level, see table 60.

R_p is the code rate of each individual level, see table 60.

HMsym:

$$L_1 = \sum_{p=1}^2 2N_1 R_p$$

$$L_2 = \sum_{p=1}^2 RX_p \left\lfloor \frac{2N_2 - 12}{RY_p} \right\rfloor,$$

$$L_{VSPP} = RX_0 \left\lfloor \frac{2(N_1 + N_2) - 12}{RY_0} \right\rfloor$$

$P_{\max} = 3$ is the number of levels for 64-QAM using HMsym.

NOTE: A hierarchical mapping scheme can only be used in a 64-QAM signal constellation.

RX_p is the numerator of the code rate of each individual level, see table 60.

RY_p is the denominator of the code rate of each individual level, see table 60.

R_p is the code rate of each individual level, see table 60.

HMmix:

$$L_1 = N_1 R_0^{\text{Im}} + \sum_{p=1}^2 N_1 (R_p^{\text{Re}} + R_p^{\text{Im}})$$

$$L_2 = RX_0^{\text{Im}} \left\lfloor \frac{N_2 - 12}{RY_0} \right\rfloor + \sum_{p=1}^2 RX_p^{\text{Re}} \left\lfloor \frac{N_2 - 12}{RY_p} \right\rfloor + RX_p^{\text{Im}} \left\lfloor \frac{N_2 - 12}{RY_p} \right\rfloor$$

$$L_{VSPP} = RX_o^{\text{Re}} \left\lfloor \frac{N_1 + N_2 - 12}{RY_0^{\text{Re}}} \right\rfloor$$

$P_{\max} = 3$ is the number of levels for 64-QAM using HMmix.

$RX_p^{\text{Re}}, RX_p^{\text{Im}}$ are the numerators of the code rates of each individual level (see table 60) for the real and imaginary component respectively.

$RY_p^{\text{Re}}, RY_p^{\text{Im}}$ are the denominators of the code rates of each individual level (see table 60) for the real and imaginary component respectively.

$R_p^{\text{Re}}, R_p^{\text{Im}}$ are the code rates of each individual level (see table 60) for the real and imaginary component respectively and $\lfloor \cdot \rfloor$ means round towards minus infinity.

The total number N_{MUX} of MSC OFDM cells per multiplex frame is given in clause 7.7.

The total number N_{MUX} of MSC OFDM cells per multiplex frame when using one protection level (EEP) equals N_2 .

The total number N_{MUX} of MSC OFDM cells per multiplex frame when using two protection levels (UEP) equals the addition of the cells of the higher protected part and the lower protected part:

$$N_{MUX} = N_1 + N_2$$

N_1 is the number of OFDM cells used for the higher protected part.

N_2 is the number of OFDM cells used for the lower protected part including the tailbits.

To calculate the number N_1 of OFDM cells in the higher protected part (part A) the following formulae apply:

SM:

$$N_1 = \left\lfloor \frac{8X}{2RY_{lcm} \sum_{p=0}^{P_{\max}-1} R_p} \right\rfloor RY_{lcm}$$

HMsym:

$$N_1 = \left\lfloor \frac{8X}{2RY_{lcm} \sum_{p=1}^2 R_p} \right\rfloor RY_{lcm}$$

HMmix:

$$N_1 = \left\lfloor \frac{8X}{RY_{lcm} \left(R_0^{\text{Im}} + \sum_{p=1}^2 (R_p^{\text{Re}} + R_p^{\text{Im}}) \right)} \right\rfloor RY_{lcm}$$

where:

X is the number of bytes in part A (as signalled in the SDC);

RY_{lcm} is taken from tables 62 and 63 for SM; from tables 64 and 65 for HMsym; and from tables 63, 65 and 66 for HMmix.

$\lceil \cdot \rceil$ means round towards plus infinity.

To calculate the number N_2 of OFDM cells in the lower protected part (part B) the following formula applies:

$$N_2 = N_{MUX} - N_1$$

The following restrictions shall be taken into account:

$$N_1 \in \{0, \dots, N_{MUX} - 20\}$$

$$N_2 \in \{20, \dots, N_{MUX}\}$$

7.2.1.2 FAC

The number of bits L_{FAC} per FAC block equals 72 bits in every mode.

The total number N_{FAC} of FAC OFDM cells per FAC block equals 65 in every mode.

7.2.1.3 SDC

The number of bits L_{SDC} per SDC block is dependent on the robustness mode, spectrum occupancy and constellation.

The total number N_{SDC} of SDC OFDM cells per SDC block are given in table 58.

The formulas given in clause 7.2.1.1 for the MSC are valid also for the SDC under the constraint of EEP and SM (only 4-QAM: $P_{\max} = 1$, 16-QAM: $P_{\max} = 2$), i.e. $L_{SDC} + L_2$ and $N_{SDC} = N_2$.

Table 58: Number of QAM cells N_{SDC} for SDC

Robustness mode	Spectrum occupancy					
	0	1	2	3	4	5
A	167	190	359	405	754	846
B	130	150	282	322	588	662
C	-	-	-	288	-	607
D	-	-	-	152	-	332

7.2.2 Energy dispersal

The purpose of the energy dispersal is to avoid the transmission of signal patterns which might result in an unwanted regularity in the transmitted signal.

For the SDC and FAC, the output of the energy dispersal shall form the input stream u_i to the corresponding multilevel coding process.

The output of the energy dispersal acting on the MSC multiplex frame shall form the standard protected input stream u_i to the multilevel coding process for the MSC. The output of the energy dispersal acting on the hierarchical frame (if present) shall form the very strongly protected input stream u'_i to the same multilevel coding process.

Energy dispersal shall be applied on the different channels (MSC, SDC, FAC) in order to reduce the possibility that systematic patterns result in unwanted regularity in either the transmitted signal or in any digital processing, this by providing a deterministic selective complementing of bits.

The individual inputs of the energy dispersal scramblers shown in figure 18 shall be scrambled by a modulo-2 addition with a Pseudo-Random Binary Sequence (PRBS), prior to channel encoding.

The PRBS is defined as the output of the feedback shift register of figure 18. It shall use a polynomial of degree 9, defined by:

$$P(X) = X^9 + X^5 + 1$$

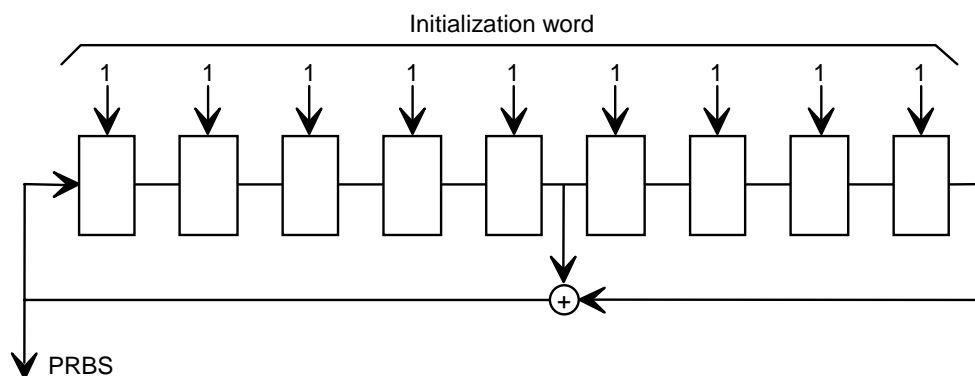


Figure 18: PRBS generator

The initialization word shall be applied in such a way that the first bit of the PRBS is obtained when the outputs of all shift register stages are set to value "1"; the first 16 bits of the PRBS are given in table 59.

Table 59: First 16 bits of the PRBS

bit index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
bit value	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0

The FAC, SDC and MSC shall be processed by the energy dispersal scramblers as follows:

- The vector length for processing equals one FAC block for the FAC, one SDC block for the SDC and one multiplex frame and one hierarchical frame for the MSC.
- Each FAC block consists of 72 bits, the block lengths for the SDC and MSC are dependent on the robustness mode, spectrum occupancy and constellation, see clause 7.2.1.
- The four blocks shall be processed independently. The input vector shall be scrambled with the PRBS, the first bit of the vector being added modulo 2 to the PRBS bit of index 0.

The scramblers of the different channels are reset as follows:

- FAC: every FAC block;
- SDC: every SDC block;
- MSC: every multiplex frame for the standard protected part, every hierarchical frame for the very strongly protected part.

7.3 Coding

Clause 7.3.1 explains the structure of multilevel coding for the different mappings, clause 7.3.2 defines the component code of the multilevel coding scheme and clause 7.3.3 the corresponding bit-wise interleaving.

7.3.1 Multilevel coding

The channel encoding process is based on a multilevel coding scheme. The principle of multilevel coding is the joint optimization of coding and modulation to reach the best transmission performance. This denotes that more error prone bit positions in the QAM mapping get a higher protection. The different levels of protection are reached with different component codes which are realized with punctured convolutional codes, derived from the same mother code.

The decoding in the receiver can be done either straightforwardly or through an iterative process. Consequently the performance of the decoder with errored data can be increased with the number of iterations and hence is dependent on the decoder implementation.

Depending on the signal constellation and mapping used, five different schemes are applicable. The 1-level scheme shall be considered as a special case of the multilevel coding scheme. Different mapping schemes are only applicable to the 64-QAM constellation as depicted in figures 25, 26 and 27. For the standard mapping and symmetrical hierarchical modulation (SM and HMsym), identical mappings shall be used for the real and imaginary components of the signal constellation. For the mixed mapping hierarchical modulation (HMmix) separate mappings shall be used for the real and imaginary components of the signal constellation.

3-level coding for SM

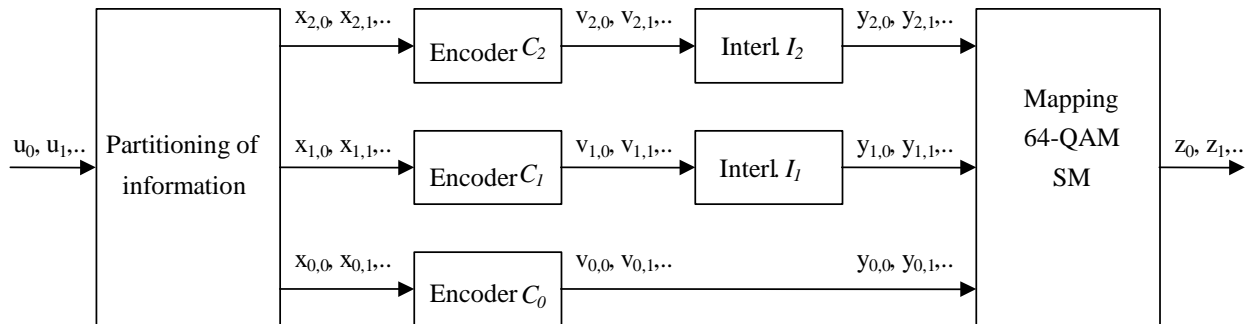


Figure 19: Multilevel coding with 3 levels for SM

3-level coding for HMsym

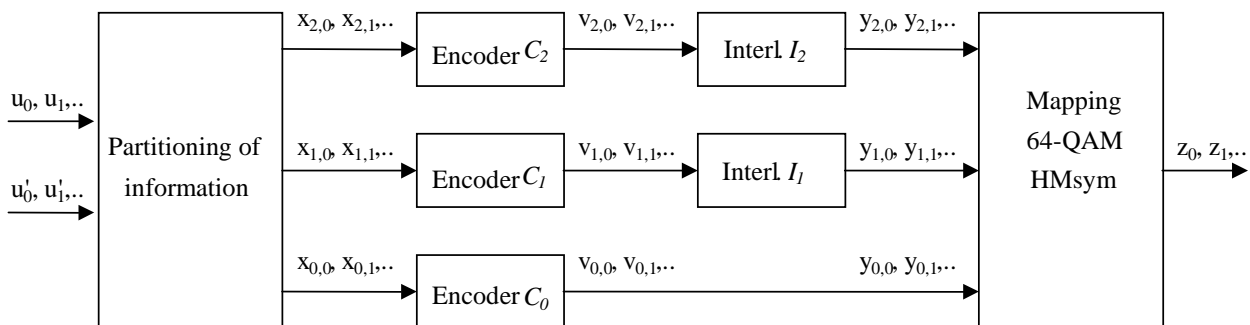


Figure 20: Multilevel coding with 3 levels for HMsym

3-level coding for HMix

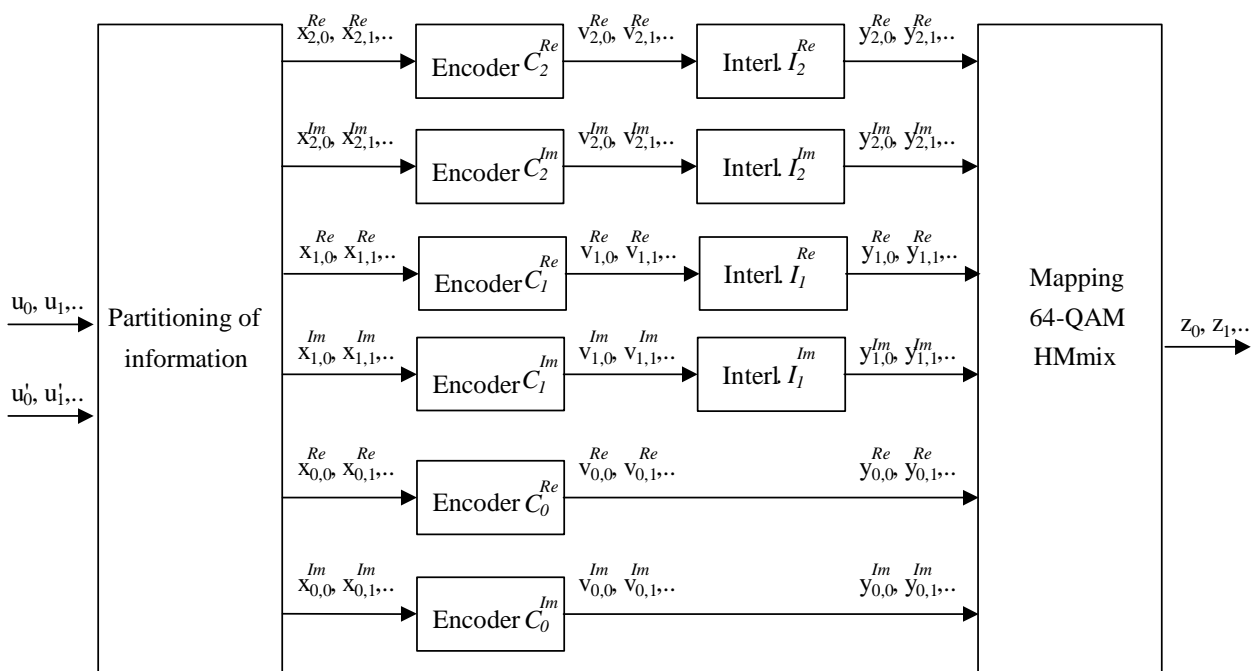


Figure 21: Multilevel coding with 3 levels for HMix

2-level coding

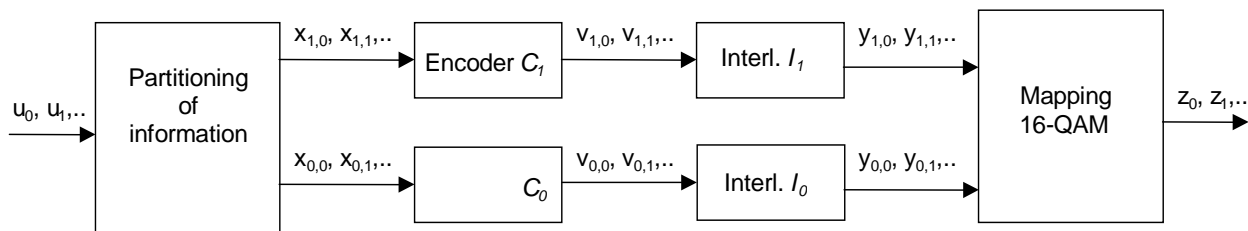


Figure 22: Multilevel coding with 2 levels (SM)

1-level coding

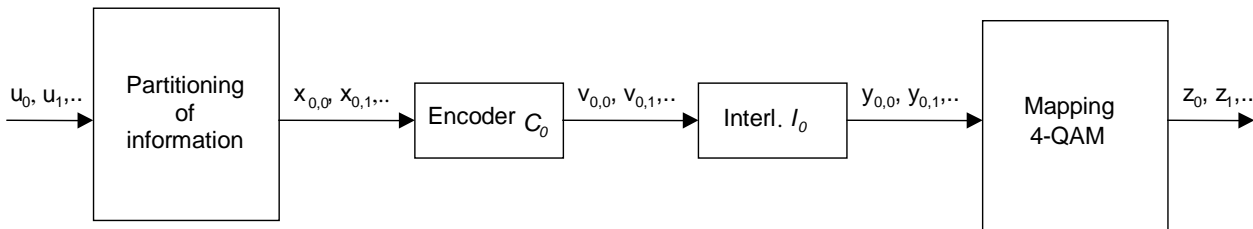


Figure 23: Multilevel coding with 1 level (SM)

7.3.1.1 Partitioning of bitstream in SM

The bitstream (u_i) shall be partitioned into several streams ($x_{p,i}$) according the number of levels. The bits of the higher protected part shall be fed to the encoders on $p = 0, \dots, P_{\max}-1$, then the bits of the lower protected part shall be fed to the encoders on $p = 0, \dots, P_{\max}-1$. This results in:

$$(x_{0,0}, x_{0,1}, \dots, x_{0,M_{0,1}-1}, x_{1,0}, x_{1,1}, \dots, x_{1,M_{1,1}-1}, x_{2,0}, x_{2,1}, \dots, x_{2,M_{2,1}-1}, x_{0,M_{0,1}}, x_{0,M_{0,1}+1}, \dots, x_{0,M_{0,1}+M_{0,2}-1}, \\ x_{1,M_{1,1}}, x_{1,M_{1,1}+1}, \dots, x_{1,M_{1,1}+M_{1,2}-1}, x_{2,M_{2,1}}, x_{2,M_{2,1}+1}, \dots, x_{2,M_{2,1}+M_{2,2}-1}) = (u_0, u_1, \dots, u_{L_{VSP}+L_1+L_2-1})$$

for the 3-level coding,

$$(x_{0,0}, x_{0,1}, \dots, x_{0,M_{0,1}-1}, x_{1,0}, x_{1,1}, \dots, x_{1,M_{1,1}-1}, \\ x_{0,M_{0,1}}, x_{0,M_{0,1}+1}, \dots, x_{0,M_{0,1}+M_{0,2}-1}, x_{1,M_{1,1}}, x_{1,M_{1,1}+1}, \dots, x_{1,M_{1,1}+M_{1,2}-1}) = (u_0, u_1, \dots, u_{L_1+L_2-1})$$

for the 2-level coding,

$$(x_{0,0}, x_{0,1}, \dots, x_{0,M_{0,1}-1}, x_{0,M_{0,1}}, x_{0,M_{0,1}+1}, \dots, x_{0,M_{0,1}+M_{0,2}-1}) = (u_0, u_1, \dots, u_{L_1+L_2-1})$$

for the 1-level coding.

When using only one protection level (EEP) the elements with negative indexes shall not be taken into account.

The number of bits on each level p is calculated for the higher protected part and lower protected part by:

$$M_{p,1} = 2N_1R_p \text{ where } p \in \{0,1,2\}$$

$$M_{p,2} = RX_p \left\lfloor \frac{2N_2-12}{RY_p} \right\rfloor \text{ where } p \in \{0,1,2\}$$

NOTE: The actual number of bits in the higher protected part (L_1) can be greater than the number signalled in the SDC. This means that some bits belonging to part B of the multiplex frame are in fact protected at the higher level.

The total number of bits on each level p is:

$$M_p = M_{p,1} + M_{p,2}$$

From these formulas it can be derived that the input bitstreams ($x_{p,i}$) to the encoders C_p have different lengths according to their code rate so that all the encoder output bitstreams ($v_{p,i}$) have the same length.

The overall code rate for each protection part for the SM is approximately:

$$R_{all} = \frac{\sum_{p=0}^{P_{\max}-1} R_p}{P_{\max}},$$

when using P_{\max} levels.

7.3.1.2 Partitioning of bitstream in HMsym

The bitstream of the SPP (u_i) shall be partitioned into two streams ($x_{p,i}$). The bits of the higher protected part shall be fed to the encoders on $p = 1$ then $p = 2$, then the bits of the lower protected part shall be fed to the encoders on $p = 1$ then $p = 2$. This results in:

$$\begin{aligned} & (x_{1,0}, x_{1,1}, \dots, x_{1,M_{1,1}-1}, x_{2,0}, x_{2,1}, \dots, x_{2,M_{2,1}-1}, x_{1,M_{1,1}}, x_{1,M_{1,1}+1}, \dots, x_{1,M_{1,1}+M_{1,2}-1}, \\ & x_{2,M_{2,1}}, x_{2,M_{2,1}+1}, \dots, x_{2,M_{2,1}+M_{2,2}-1}) = (u_0, u_1, \dots, u_{L_1+L_2-1}) \end{aligned}$$

When using only one protection level (EEP) the elements with negative indexes shall not be taken into account.

The bitstream of the VSPP (u'_i) shall be sent to the encoder on level 0:

$$(x_{0,0}, x_{0,1}, \dots, x_{0,M_{0,2}-1}) = (u'_0, u'_1, \dots, u'_{L_{VSPP}-1})$$

The number of bits on each level p is calculated for the higher protected part and lower protected part by:

$$M_{p,1} = 2N_1R_p \text{ where } p \in \{1,2\}$$

$$M_{p,2} = RX_p \left\lfloor \frac{2N_2 - 12}{RY_p} \right\rfloor \text{ where } p \in \{1,2\}$$

and

$$M_{0,1} = 0$$

$$M_{0,2} = RX_0 \left\lfloor \frac{2(N_1 + N_2) - 12}{RY_0} \right\rfloor = L_{VSPP}$$

The total number of bits on each level p is:

$$M_p = M_{p,1} + M_{p,2}$$

From these formulas it can be derived that the input bitstreams ($x_{p,i}$) to the encoders C_p have different lengths according to their code rate so that all the encoder output bitstreams ($v_{p,i}$) have the same length.

The overall code rate for each protection part for the HMsym is approximately:

$$R_{VSPP} = R_0$$

$$R_{SPP,all} = (R_1 + R_2) / 2$$

7.3.1.3 Partitioning of bitstream in HMMix

The bitstream of the SPP (u_i) shall be partitioned into five streams ($x_{p,i}^{Re}, x_{p,i}^{Im}$). The bits of the higher protected part shall be fed to the encoders on $p = 0, \dots, 2$, then the bits of the lower protected part shall be fed to the encoders on $p = 0, \dots, 2$. This results in:

$$\begin{aligned} & (x_{0,0}^{Im}, x_{0,1}^{Im}, \dots, x_{0,M_{0,1}^{Im}-1}^{Im}, x_{1,0}^{Re}, x_{1,1}^{Re}, \dots, x_{1,M_{1,1}^{Re}-1}^{Re}, x_{1,0}^{Im}, x_{1,1}^{Im}, \dots, x_{1,M_{1,1}^{Im}-1}^{Im}, x_{2,0}^{Re}, x_{2,1}^{Re}, \dots, x_{2,M_{2,1}^{Re}-1}^{Re}, x_{2,0}^{Im}, x_{2,1}^{Im}, \dots, x_{2,M_{2,1}^{Im}-1}^{Im}, \\ & x_{0,M_{0,1}^{Im}}^{Im}, x_{0,M_{0,1}^{Im}+1}^{Im}, \dots, x_{0,M_{0,1}^{Im}+M_{0,2}^{Im}-1}^{Im}, x_{1,M_{1,1}^{Re}}^{Re}, x_{1,M_{1,1}^{Re}+1}^{Re}, \dots, x_{1,M_{1,1}^{Re}+M_{1,2}^{Re}-1}^{Re}, x_{1,M_{1,1}^{Im}}, x_{1,M_{1,1}^{Im}+1}^{Im}, \dots, x_{1,M_{1,1}^{Im}+M_{1,2}^{Im}-1}^{Im}, \\ & x_{2,M_{2,1}^{Re}}^{Re}, x_{2,M_{2,1}^{Re}+1}^{Re}, \dots, x_{2,M_{2,1}^{Re}+M_{2,2}^{Re}-1}^{Re}, x_{2,M_{2,1}^{Im}}, x_{2,M_{2,1}^{Im}+1}^{Im}, \dots, x_{2,M_{2,1}^{Im}+M_{2,2}^{Im}-1}^{Im}) = (u_0, u_1, \dots, u_{L_1+L_2-1}) \end{aligned}$$

The bits of the VSPP (u'_i) shall be fed to the encoder for the real part on level $p = 0$:

$$(x_{0,0}^{\text{Re}}, x_{0,1}^{\text{Re}}, \dots, x_{0,M_{0,2}^{\text{Re}}-1}^{\text{Re}}) = (u'_0, u'_1, \dots, u'_{L_{VSPP}-1})$$

When using only one protection level (EEP) the elements with negative indexes shall not be taken into account.

The number of bits on each level p is calculated for the higher protected and lower protected parts for the real and imaginary component by:

$$M_{0,1}^{\text{Re}} = 0, \quad M_{0,1}^{\text{Im}} = N_1 R_0^{\text{Im}}$$

$$M_{0,2}^{\text{Re}} = RX_0^{\text{Re}} \left\lfloor \frac{N_1 + N_2 - 12}{RY_0^{\text{Re}}} \right\rfloor = L_{VSPP}, \quad M_{0,2}^{\text{Im}} = RX_0^{\text{Im}} \left\lfloor \frac{N_2 - 12}{RY_0^{\text{Im}}} \right\rfloor$$

$$M_{p,1}^{\text{Re}} = N_1 R_p^{\text{Re}} \quad \text{and} \quad M_{p,1}^{\text{Im}} = N_1 R_p^{\text{Im}} \quad \text{for } p \in \{1, 2\}$$

$$M_{p,2}^{\text{Re}} = RX_p^{\text{Re}} \left\lfloor \frac{N_2 - 12}{RY_p^{\text{Re}}} \right\rfloor \quad \text{and} \quad M_{p,2}^{\text{Im}} = RX_p^{\text{Im}} \left\lfloor \frac{N_2 - 12}{RY_p^{\text{Im}}} \right\rfloor \quad \text{for } p \in \{1, 2\}$$

The total number of bits on each level p in the real and imaginary component results in:

$$M_p^{\text{Re}} = M_{p,1}^{\text{Re}} + M_{p,2}^{\text{Re}} \quad \text{and} \quad M_p^{\text{Im}} = M_{p,1}^{\text{Im}} + M_{p,2}^{\text{Im}} \quad \text{for } p \in \{0, 1, 2\}$$

From these formulas it can be derived that the input bitstreams ($x_{p,i}^{\text{Re}}$) and ($x_{p,i}^{\text{Im}}$) to the encoders C_p^{Re} and C_p^{Im} respectively have different lengths according to their code rate so that all the encoder output bitstreams $p \in \{0, 1, 2\}$ have the same length.

The overall code rate for the HMmix schemes of each protection part is approximately:

$$R_{VSPP} = R_0^{\text{Re}}$$

$$R_{SPP,all} = (R_0^{\text{Im}} + R_1^{\text{Re}} + R_1^{\text{Im}} + R_2^{\text{Re}} + R_2^{\text{Im}}) / 5$$

7.3.2 Component code

The component code C_p is based on punctured convolutional coding with a mother code of rate 1/4 and constraint length 7. The mother convolutional encoder generates from the vector $(x_{p,i})_{i=0}^{M_p-1} = (a_i)_{i=0}^{I-1}$ a codeword

$\{(b_{0,i}, b_{1,i}, b_{2,i}, b_{3,i})\}_{i=0}^{I+5}$. This codeword is defined by:

$$b_{0,i} = a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6};$$

$$b_{1,i} = a_i \oplus a_{i-1} \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-6};$$

$$b_{2,i} = a_i \oplus a_{i-1} \oplus a_{i-4} \oplus a_{i-6};$$

$$b_{3,i} = a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6};$$

for $i = 0, 1, 2, \dots, I + 5$.

When i does not belong to the set $\{0, 1, 2, \dots, I-1\}$, a_i is equal to zero by definition.

The encoding can be achieved using the convolutional encoder presented in figure 24.

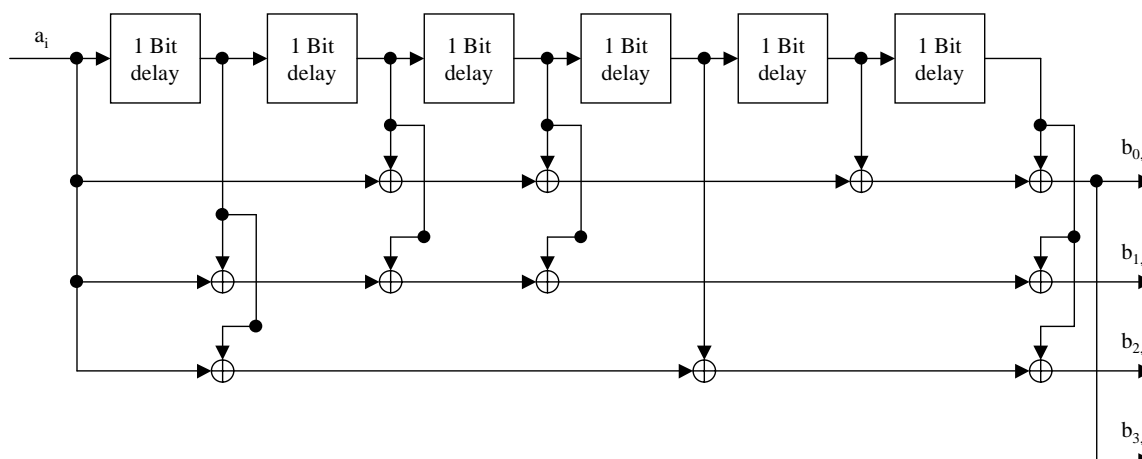


Figure 24: Convolutional encoder

The octal forms of the generator polynomials are 133, 171, 145 and 133, respectively.

The vector $(a_{-6}, a_{-5}, a_{-4}, a_{-3}, a_{-2}, a_{-1})$ corresponds to the all-zero initial state of the shift register and the vector $(a_I, a_{I+1}, a_{I+2}, a_{I+3}, a_{I+4}, a_{I+5})$ corresponds to the all-zero final state of the shift register.

In addition to the mother code the system shall allow punctured rates. Table 60 shows the puncturing patterns.

Table 60: Puncturing patterns

Code rates R_p	Numerator RX_p	Denominator RY_p	Puncturing pattern	Transmitted sequence
1/4	1	4	B ₀ : 1 B ₁ : 1 B ₂ : 1 B ₃ : 1	b _{0,0} b _{1,0} b _{2,0} b _{3,0} etc.
3/10	3	10	B ₀ : 1 1 1 B ₁ : 1 1 1 B ₂ : 1 1 1 B ₃ : 1 0 0	b _{0,0} b _{1,0} b _{2,0} b _{3,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} etc.
1/3	1	3	B ₀ : 1 B ₁ : 1 B ₂ : 1 B ₃ : 0	b _{0,0} b _{1,0} b _{2,0} etc.
4/11	4	11	B ₀ : 1 1 1 1 B ₁ : 1 1 1 1 B ₂ : 1 1 1 0 B ₃ : 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} etc.
1/2	1	2	B ₀ : 1 B ₁ : 1 B ₂ : 0 B ₃ : 0	b _{0,0} b _{1,0} etc.
4/7	4	7	B ₀ : 1 1 1 1 B ₁ : 1 0 1 0 B ₂ : 0 1 0 0 B ₃ : 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{2,1} b _{0,2} b _{1,2} b _{0,3} etc.

Code rates R_p	Numerator RX_p	Denominator RY_p	Puncturing pattern	Transmitted sequence
3/5	3	5	B ₀ : 1 1 1 B ₁ : 1 0 1 B ₂ : 0 0 0 B ₃ : 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} b _{1,2} etc.
2/3	2	3	B ₀ : 1 1 B ₁ : 1 0 B ₂ : 0 0 B ₃ : 0 0	b _{0,0} b _{1,0} b _{0,1} etc.
8/11	8	11	B ₀ : 1 1 1 1 1 1 1 1 B ₁ : 1 0 0 1 0 0 1 0 B ₂ : 0 0 0 0 0 0 0 0 B ₃ : 0 0 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} b _{0,3} b _{1,3} b _{0,4} b _{0,5} b _{0,6} b _{1,6} b _{0,7} etc.
3/4	3	4	B ₀ : 1 1 1 B ₁ : 1 0 0 B ₂ : 0 0 0 B ₃ : 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} etc.
4/5	4	5	B ₀ : 1 1 1 1 B ₁ : 1 0 0 0 B ₂ : 0 0 0 0 B ₃ : 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} b _{0,3} b _{0,4} etc.
7/8	7	8	B ₀ : 1 1 1 1 1 1 1 1 B ₁ : 1 0 0 0 0 0 0 0 B ₂ : 0 0 0 0 0 0 0 0 B ₃ : 0 0 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} b _{0,3} b _{0,4} b _{0,5} b _{0,6} etc.
8/9	8	9	B ₀ : 1 1 1 1 1 1 1 1 1 B ₁ : 1 0 0 0 0 0 0 0 0 B ₂ : 0 0 0 0 0 0 0 0 0 B ₃ : 0 0 0 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{0,2} b _{0,3} b _{0,4} b _{0,5} b _{0,6} b _{0,7} etc.

For the FAC, all bits are punctured according to table 60. For the MSC and the SDC, the last 24 bits (the tailbits) of the serial mother codeword shall be punctured as follows. The index r_p shall be used with table 61 to find the puncturing vector for the tailbits for each level. This index is calculated with the following formula:

SM:

$$r_p = (2N_2 - 12) - RY_p \left\lfloor \frac{2N_2 - 12}{RY_p} \right\rfloor \text{ for } p \in \{0,1,2\}$$

HMsym:

$$r_0 = (2(N_1 + N_2) - 12) - RY_0 \left\lfloor \frac{2(N_1 + N_2) - 12}{RY_0} \right\rfloor$$

$$r_p = (2N_2 - 12) - RY_p \left\lfloor \frac{2N_2 - 12}{RY_p} \right\rfloor \text{ for } p \in \{1,2\}$$

HMmix:

$$r_0^{\text{Re}} = (N_1 + N_2 - 12) - RY_0^{\text{Re}} \left\lfloor \frac{N_1 + N_2 - 12}{RY_0^{\text{Re}}} \right\rfloor,$$

$$r_p^{\text{Re}} = (N_2 - 12) - RY_p^{\text{Re}} \left\lfloor \frac{N_2 - 12}{RY_p^{\text{Re}}} \right\rfloor \text{ for } p \in \{1, 2\}$$

$$r_p^{\text{Im}} = (N_2 - 12) - RY_p^{\text{Im}} \left\lfloor \frac{N_2 - 12}{RY_p^{\text{Im}}} \right\rfloor \text{ for } p \in \{0, 1, 2\}$$

Table 61: Puncturing patterns of the tailbits

r_p	Puncturing pattern	Transmitted sequence
0	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 0 0 0 0 0 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{0,1} b _{1,1} b _{0,2} b _{1,2} b _{0,3} b _{1,3} b _{0,4} b _{1,4} b _{0,5} b _{1,5} etc.
1	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 0 0 0 0 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{0,2} b _{1,2} b _{0,3} b _{1,3} b _{0,4} b _{1,4} b _{0,5} b _{1,5} etc.
2	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 0 0 1 0 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{0,2} b _{1,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{0,5} b _{1,5} etc.
3	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 0 1 0 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{0,5} b _{1,5} etc.
4	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 0 1 1 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} etc.
5	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 1 1 1 0 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} etc.
6	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 1 1 1 1 B ₃ : 0 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} b _{2,5} etc.
7	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 1 1 1 1 B ₃ : 1 0 0 0 0 0	b _{0,0} b _{1,0} b _{2,0} b _{3,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} b _{2,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} b _{2,5} etc.
8	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 1 1 1 1 B ₃ : 1 0 0 1 0 0	b _{0,0} b _{1,0} b _{2,0} b _{3,0} b _{0,1} b _{1,1} b _{2,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} b _{2,3} b _{3,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} b _{2,5} etc.
9	B ₀ : 1 1 1 1 1 1 B ₁ : 1 1 1 1 1 1 B ₂ : 1 1 1 1 1 1 B ₃ : 1 1 0 1 0 0	b _{0,0} b _{1,0} b _{2,0} b _{3,0} b _{0,1} b _{1,1} b _{2,1} b _{3,1} b _{0,2} b _{1,2} b _{2,2} b _{0,3} b _{1,3} b _{2,3} b _{3,3} b _{0,4} b _{1,4} b _{2,4} b _{0,5} b _{1,5} b _{2,5} etc.

r_p	Puncturing pattern	Transmitted sequence
10	$B_0: 1\ 1\ 1\ 1\ 1\ 1$ $B_1: 1\ 1\ 1\ 1\ 1\ 1$ $B_2: 1\ 1\ 1\ 1\ 1\ 1$ $B_3: 1\ 1\ 0\ 1\ 0\ 1$	$b_{0,0}\ b_{1,0}\ b_{2,0}\ b_{3,0}\ b_{0,1}\ b_{1,1}\ b_{2,1}\ b_{3,1}\ b_{0,2}\ b_{1,2}\ b_{2,2}\ b_{0,3}\ b_{1,3}\ b_{2,3}$ $b_{3,3}\ b_{0,4}\ b_{1,4}\ b_{2,4}\ b_{0,5}\ b_{1,5}\ b_{2,5},\ b_{3,5},\ \text{etc.}$
11	$B_0: 1\ 1\ 1\ 1\ 1\ 1$ $B_1: 1\ 1\ 1\ 1\ 1\ 1$ $B_2: 1\ 1\ 1\ 1\ 1\ 1$ $B_3: 1\ 1\ 1\ 1\ 0\ 1$	$b_{0,0}\ b_{1,0}\ b_{2,0}\ b_{3,0}\ b_{0,1}\ b_{1,1}\ b_{2,1}\ b_{3,1}\ b_{0,2}\ b_{1,2}\ b_{2,2},\ b_{3,2},\ b_{0,3}$ $b_{1,3}\ b_{2,3}\ b_{3,3}\ b_{0,4}\ b_{1,4}\ b_{2,4}\ b_{0,5}\ b_{1,5}\ b_{2,5},\ b_{3,5}\ \text{etc.}$

The puncturing shall be performed as follows:

SM:

The higher protected part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,0} \dots v_{p,2N_1-1} \right).$$

The lower protected part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,2N_1} \dots v_{p,2(N_1+N_2)-13-r_p} \right).$$

The tailbits of the transmitted sequence are punctured according to table 61 resulting in:

$$\left(v_{p,2(N_1+N_2)-12-r_p} \dots v_{p,2(N_1+N_2)-1} \right).$$

NOTE 1: If there is only one protection level the higher protected part is absent.

HMSym:

The VSPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{0,0} \dots v_{0,2(N_1+N_2)-13-r_0} \right).$$

The tailbits for the VSPP part of the transmitted sequence are punctured according to table 61 resulting in:

$$\left(v_{0,2(N_1+N_2)-12-r_0} \dots v_{0,2(N_1+N_2)-1} \right).$$

The higher protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,0} \dots v_{p,2N_1-1} \right) \text{ for } p \in \{1,2\}.$$

The lower protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,2N_1} \dots v_{p,2(N_1+N_2)-13-r_p} \right) \text{ for } p \in \{1,2\}.$$

The tailbits for the SPP part of the transmitted sequence is punctured according to table 61 resulting in:

$$\left(v_{p,2(N_1+N_2)-12-r_p} \dots v_{p,2(N_1+N_2)-1} \right) \text{ for } Y(p) = (y_{p,0}, y_{p,1}, y_{p,2}, \dots, y_{p,2(N_1+N_2)-1}).$$

NOTE 2: If there is only one protection level the higher protected part is absent.

HMmix:

The VSPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{0,0}^{\text{Re}} \dots v_{0,2(N_1+N_2)-13-r_0}^{\text{Re}} \right).$$

The tailbits for the VSPP part of the transmitted sequence are punctured according to table 61 resulting in:

$$\left(v_{0,2(N_1+N_2)-12-r_0}^{\text{Re}} \dots v_{0,2(N_1+N_2)-1}^{\text{Re}} \right).$$

The real component of the higher protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,0}^{\text{Re}} \dots v_{p,2N_1-1}^{\text{Re}} \right) \text{ for } X_{in} = 2N_2.$$

The real component of the lower protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,2N_1}^{\text{Re}} \dots v_{p,2(N_1+N_2)-13-r_p}^{\text{Re}} \right) \text{ for } p \in \{1,2\}.$$

The tailbits for the SPP part of the transmitted sequence are punctured according to table 61 resulting in:

$$\left(v_{p,2(N_1+N_2)-12-r_p}^{\text{Re}} \dots v_{p,2(N_1+N_2)-1}^{\text{Re}} \right).$$

NOTE 3: If there is only one protection level the higher protected part is absent.

The imaginary component of the higher protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,0}^{\text{Im}} \dots v_{p,2N_1-1}^{\text{Im}} \right) \text{ for } p \in \{0,1,2\}.$$

The imaginary component of the lower protected part of the SPP part of the transmitted sequence is punctured according to table 60 resulting in:

$$\left(v_{p,2N_1}^{\text{Im}} \dots v_{p,2(N_1+N_2)-13-r_p}^{\text{Im}} \right) \text{ for } p \in \{0,1,2\}.$$

The tailbits for the SPP part of the transmitted sequence are punctured according to table 61 resulting in:

$$\left(v_{p,2(N_1+N_2)-12-r_p}^{\text{Im}} \dots v_{p,2(N_1+N_2)-1}^{\text{Im}} \right) \text{ for } p \in \{0,1,2\}.$$

NOTE 4: If there is only one protection level the higher protected part is absent.

7.3.3 Bit interleaving

Bit-wise interleaving shall be applied for some of the levels of the coding scheme according to figures 19 to 23. The same basic algorithm which results in a pseudo random bit ordering shall be used independently for the FAC, SDC and MSC.

The permutation $\Pi_p(i)$ is obtained from the following relations:

for 64-QAM: $t_1 = 13$, $t_2 = 21$

for 16-QAM: $t_0 = 13$, $t_1 = 21$

for 4-QAM: $t_0 = 21$

$$p \in \{0,1,2\}$$

$$s = 2^{\lceil \log_2(x_{in}) \rceil}$$

$$q = s / 4 - 1$$

the number of input bits x_{in} is defined below and $\lceil \cdot \rceil$ means round towards plus infinity

$$\Pi_p(0) = 0;$$

for $i = 1, 2, \dots, x_{in} - 1$:

$$\Pi_p(i) = (t_p \Pi_p(i-1) + q) \pmod{s};$$

while $\Pi_p(i) \geq x_{in}$:

$$\Pi_p(i) = (t_p \Pi_p(i) + q) \pmod{s}.$$

7.3.3.1 FAC

The block size shall be in every case the same for the interleaver I_p with $p = 0$ only. The number of elements per bit interleaver x_{in} equals $2N_{FAC}$. The input vector is defined by:

$$V(p) = (v_{p,0}, v_{p,1}, v_{p,2}, \dots, v_{p,2N_{FAC}-1})$$

The interleaved output vector is the subset of the permutations $\Pi_p(i)$ defined by:

$$Y(p) = (y_{p,0}, y_{p,1}, y_{p,2}, \dots, y_{p,2N_{FAC}-1})$$

The output elements are selected from the input elements according to:

$$y_{p,i} = v_{p,\Pi_p(i)}.$$

7.3.3.2 SDC

The block size shall be the same for each interleaver I_p . The number of elements per bit interleaver x_{in} equals $2N_{SDC}$. For each bit interleaver, the input vector is defined by:

$$V(p) = (v_{p,0}, v_{p,1}, v_{p,2}, \dots, v_{p,2N_{SDC}-1})$$

The interleaved output vector is the subset of the permutations $\Pi_p(i)$ defined by:

$$Y(p) = (y_{p,0}, y_{p,1}, y_{p,2}, \dots, y_{p,2N_{SDC}-1})$$

The output elements are selected from the input elements according to:

$$y_{p,i} = v_{p,\Pi_p(i)}.$$

7.3.3.3 MSC

SM and HMsym:

The block size shall be the same for each interleaver I_p , but shall be dependent on the robustness mode, spectrum occupancy and the constellation. The number of elements per bit interleaver equals $2(N_1 + N_2)$. For each bit interleaver, the input vector is defined by:

$$V(p) = (v_{p,0}, v_{p,1}, v_{p,2}, \dots, v_{p,2(N_1+N_2)-1}) = (v_{1,p,0}, v_{1,p,1}, \dots, v_{1,p,2N_1-1}, v_{2,p,0}, v_{2,p,1}, \dots, v_{2,p,2N_2-1})$$

The interleaved output vector is the subset of the two permutations $\Pi_p(i)$ defined by:

$$Y(p) = (y_{p,0}, y_{p,1}, y_{p,2}, \dots, y_{p,2(N_1+N_2)-1}) = (y_{1,p,0}, y_{1,p,1}, \dots, y_{1,p,2N_1-1}, y_{2,p,0}, y_{2,p,1}, \dots, y_{2,p,2N_2-1})$$

The two parts with different protection levels shall not overlap in the interleaving process. Therefore the interleaved lower protected part shall be appended to the interleaved higher protected part where the output elements are selected from the input elements according to:

$$y_{1,p,i} = v_{1,p,\Pi_p(i)} \text{ and } y_{2,p,i} = v_{2,p,\Pi_p(i)}$$

for each part respectively.

The number of input bits used for the permutation for the higher protected part is $x_{in} = 2N_1$, and for the lower protected part is $x_{in} = 2N_2$.

HMmix:

The block size shall be the same for each interleaver I_p^{Re} and I_p^{Im} but shall be dependent on the robustness mode, spectrum occupancy and the constellation. The number of elements per bit interleaver equals $(N_1 + N_2)$. For each bit interleaver, the input vectors for the real and imaginary components are defined by:

$$V^{Re}(p) = (v_{p,0}^{Re}, v_{p,1}^{Re}, v_{p,2}^{Re}, \dots, v_{p,N_1+N_2-1}^{Re}) = (v_{1,p,0}^{Re}, v_{1,p,1}^{Re}, \dots, v_{1,p,N_1-1}^{Re}, v_{2,p,0}^{Re}, v_{2,p,1}^{Re}, \dots, v_{2,p,N_2-1}^{Re}) \text{ or}$$

$$V^{Im}(p) = (v_{p,0}^{Im}, v_{p,1}^{Im}, v_{p,2}^{Im}, \dots, v_{p,N_1+N_2-1}^{Im}) = (v_{1,p,0}^{Im}, v_{1,p,1}^{Im}, \dots, v_{1,p,N_1-1}^{Im}, v_{2,p,0}^{Im}, v_{2,p,1}^{Im}, \dots, v_{2,p,N_2-1}^{Im}) \text{ respectively.}$$

The interleaved output vectors for the real and imaginary components are the subsets of the two permutations $\Pi_p(i)$ defined by:

$$Y^{Re}(p) = (y_{p,0}^{Re}, y_{p,1}^{Re}, y_{p,2}^{Re}, \dots, y_{p,N_1+N_2-1}^{Re}) = (y_{1,p,0}^{Re}, y_{1,p,1}^{Re}, \dots, y_{1,p,N_1-1}^{Re}, y_{2,p,0}^{Re}, y_{2,p,1}^{Re}, \dots, y_{2,p,N_2-1}^{Re}) \text{ or}$$

$$Y^{Im}(p) = (y_{p,0}^{Im}, y_{p,1}^{Im}, y_{p,2}^{Im}, \dots, y_{p,N_1+N_2-1}^{Im}) = (y_{1,p,0}^{Im}, y_{1,p,1}^{Im}, \dots, y_{1,p,N_1-1}^{Im}, y_{2,p,0}^{Im}, y_{2,p,1}^{Im}, \dots, y_{2,p,N_2-1}^{Im}) \text{ respectively.}$$

The two parts with different protection levels shall not overlap in the interleaving process. Therefore the interleaved lower protected part shall be appended to the interleaved higher protected part where the output elements are selected from the input elements according to:

$$y_{1,p,i}^{Re} = v_{1,p,\Pi(i)}^{Re}, y_{2,p,i}^{Re} = v_{2,p,\Pi(i)}^{Re}, y_{1,p,i}^{Im} = v_{1,p,\Pi(i)}^{Im} \text{ and } y_{2,p,i}^{Im} = v_{2,p,\Pi(i)}^{Im}$$

for each part respectively.

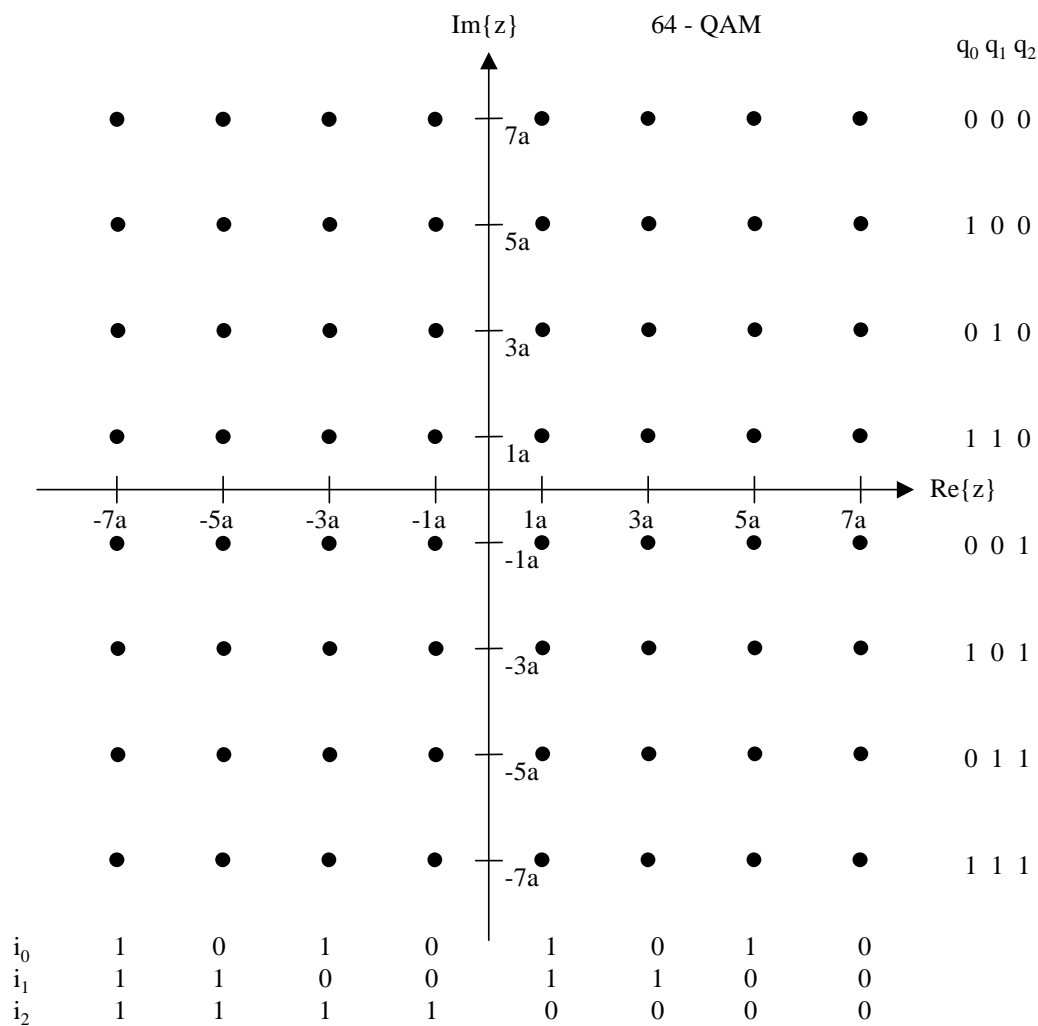
The number of input bits used for the permutation for the higher protected parts is $x_{in} = N_1$, and for the lower protected parts is $x_{in} = N_2$.

7.4 Signal constellations and mapping

The mapping strategy for each OFDM cell is dependent of the assignment to the channel (FAC, SDC, MSC) and the robustness mode. All data cells are either 4-QAM, 16-QAM or 64-QAM.

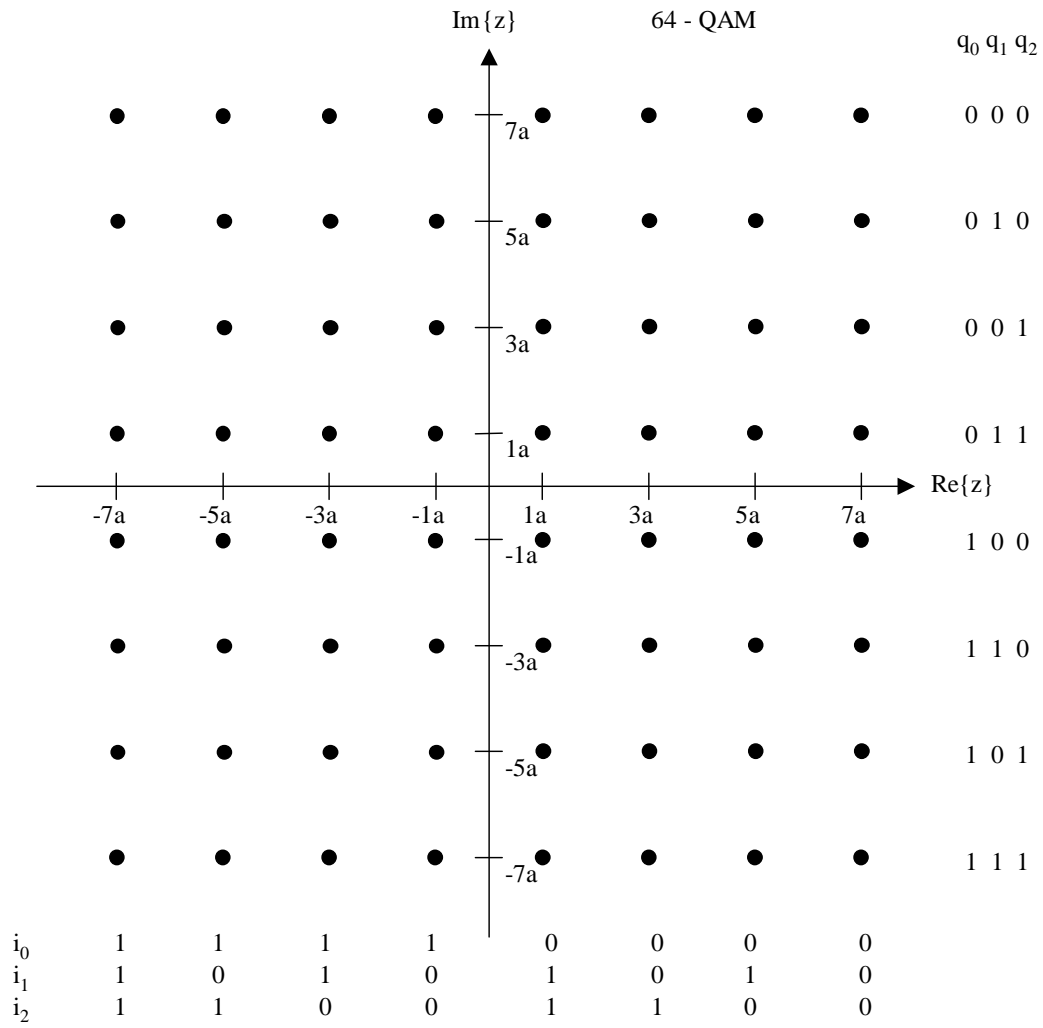
The default method for mapping shall be performed according to figures 25 to 29.

The y'_i denote the bits representing a complex modulation symbol z .



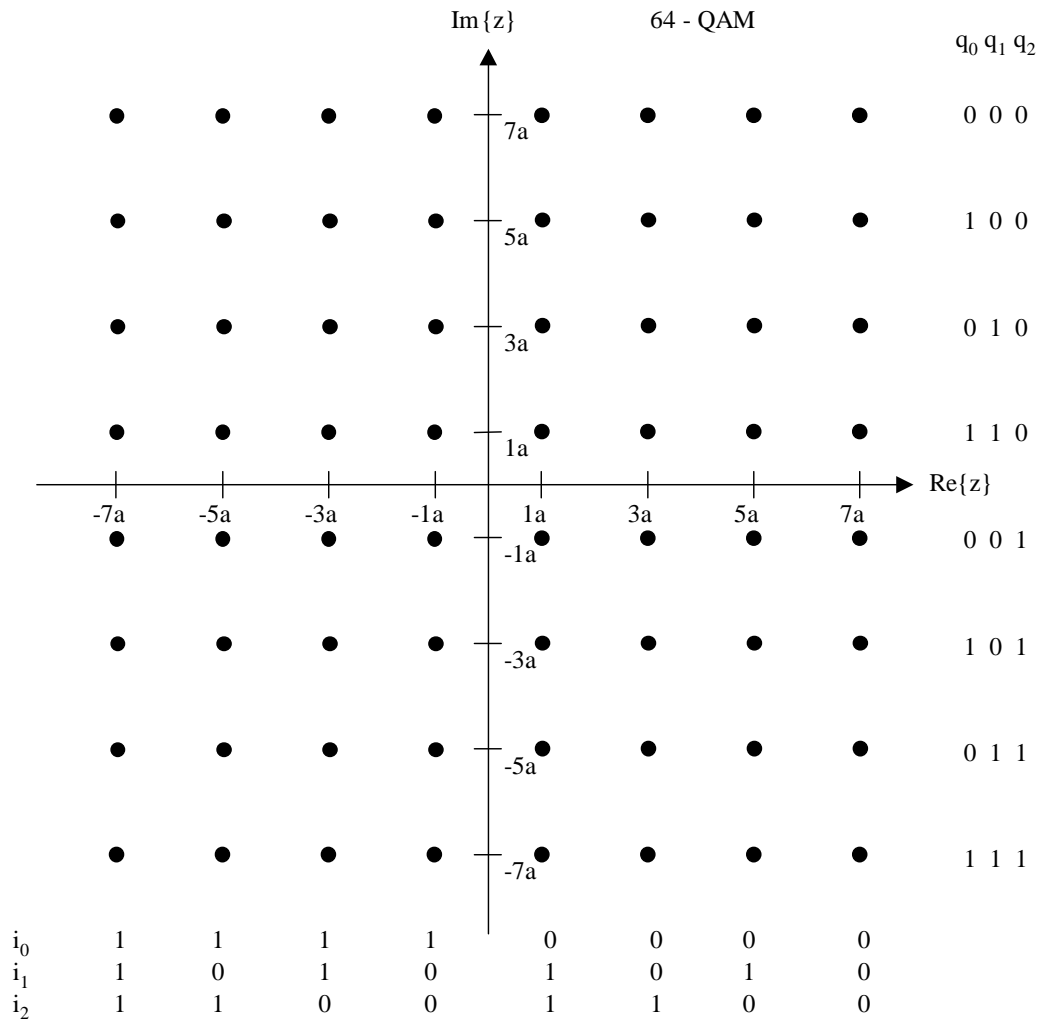
Bit ordering: $\{i_0 i_1 i_2 q_0 q_1 q_2\} = \{y'_0 y'_1 y'_2 y'_3 y'_4 y'_5\}$

Figure 25: SM 64-QAM mapping with corresponding bit pattern



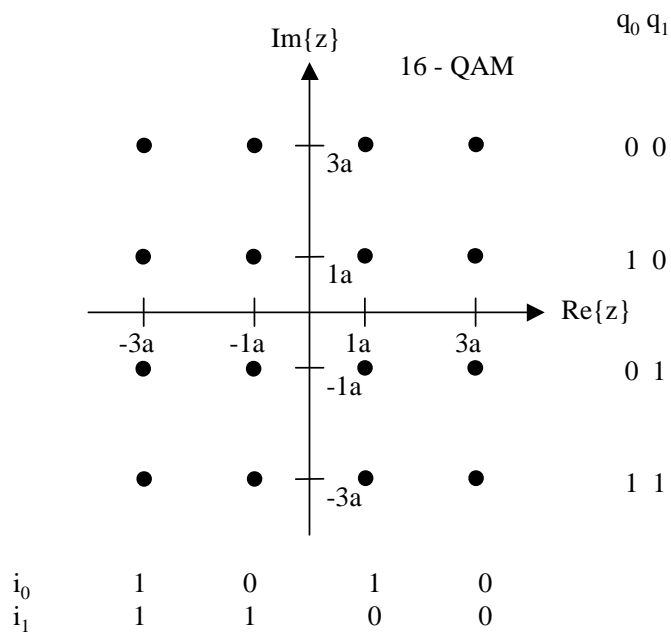
Bit ordering: $\{i_0 i_1 i_2 q_0 q_1 q_2\} = \{y'_0 y'_1 y'_2 y'_3 y'_4 y'_5\}$

Figure 26: HMsym 64-QAM mapping with corresponding bit pattern



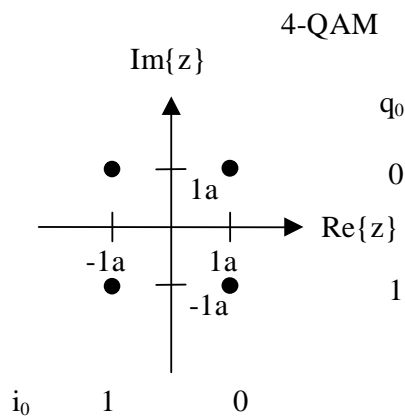
Bit ordering: $\{i_0 i_1 i_2 q_0 q_1 q_2\} = \{y'_0 y'_1 y'_2 y'_3 y'_4 y'_5\}$

Figure 27: HMmix 64-QAM mapping with corresponding bit pattern



Bit ordering: $\{i_0 i_1 q_0 q_1\} = \{y'_0 y'_1 y'_2 y'_3\}$

Figure 28: SM 16-QAM mapping with corresponding bit pattern



Bit ordering: $\{i_0 q_0\} = \{y'_0 y'_1\}$

Figure 29: SM 4-QAM mapping with corresponding bit pattern

NOTE: Left hand bit is the first in time.

For 64-QAM, the normalization factor is $a = \frac{1}{\sqrt{42}}$.

For 16-QAM, the normalization factor is $a = \frac{1}{\sqrt{10}}$.

For 4-QAM, the normalization factor is $a = \frac{1}{\sqrt{2}}$.

The data stream at the output of the interleaver consists of a number of bit words. These are mapped onto one signal point in the signal diagram according to a complex number z . For SM and HMsym the 64-QAM diagram shall be used according to figure 25 and 26 respectively. The bits shall be mapped accordingly:

$$\left(\overset{\cdot}{y}_0 \overset{\cdot}{y}_1 \overset{\cdot}{y}_2 \overset{\cdot}{y}_3 \overset{\cdot}{y}_4 \overset{\cdot}{y}_5 \right) = (y_{0,0} y_{1,0} y_{2,0} y_{0,1} y_{1,1} y_{2,1})$$

For HMmix the 64-QAM diagram shall be used according to figure 27. The bits shall be mapped accordingly:

$$\left(\overset{\cdot}{y}_0 \overset{\cdot}{y}_1 \overset{\cdot}{y}_2 \overset{\cdot}{y}_3 \overset{\cdot}{y}_4 \overset{\cdot}{y}_5 \right) = \left(y_{0,0}^{\text{Re}} y_{1,0}^{\text{Re}} y_{2,0}^{\text{Re}} y_{0,0}^{\text{Im}} y_{1,0}^{\text{Im}} y_{2,0}^{\text{Im}} \right)$$

The 16-QAM diagram shall be used according figure 28. The bits shall be mapped accordingly:

$$\left(\overset{\cdot}{y}_0 \overset{\cdot}{y}_1 \overset{\cdot}{y}_2 \overset{\cdot}{y}_3 \right) = (y_{0,0} y_{1,0} y_{0,1} y_{1,1})$$

The 4-QAM diagram shall be used according figure 29. The bits shall be mapped accordingly:

$$\left(\overset{\cdot}{y}_0 \overset{\cdot}{y}_1 \right) = (y_{0,0} y_{0,1})$$

7.5 Application of coding to the channels

7.5.1 Coding the MSC

The MSC may use either 64-QAM or 16-QAM mapping. 64-QAM provides high spectral efficiency whereas 16-QAM provides a more robust error performance. In each case, a range of code rates is available to provide the most appropriate level of error correction for a given transmission. The available combinations of constellation and code rate provide a large degree of flexibility over a wide range of transmission channels. Unequal error protection can be used to provide two levels of protection for the MSC. For the case of 64-QAM, hierarchical modulation may be used to provide a third level of error robustness for a part of the MSC.

7.5.1.1 SM

Two protection levels within one multiplex frame are possible resulting in the use of two overall code rates. The number of input bits L_{MUX} per multiplex frame are calculated with the formulas of clause 7.2.

The MSC shall be encoded according to clause 7.3. The overall code rates and code rates for each level are defined in tables 62 and 63. The protection level is signalled in the multiplex description data entity of the SDC (see clause 6.4.3.1).

Two overall code rates are defined for 16-QAM as follows:

Table 62: Code rate combinations for the MSC with 16-QAM

Protection level	R_{all}	R_0	R_1	RY_{lcm}
0	0,5	1/3	2/3	3
1	0,62	1/2	3/4	4

Four overall code rates are defined for 64-QAM as follows:

Table 63: Code rate combinations for the MSC with 64-QAM

Protection level	R_{all}	R_0	R_1	R_2	RY_{lcm}
0	0,5	1/4	1/2	3/4	4
1	0,6	1/3	2/3	4/5	15
2	0,71	1/2	3/4	7/8	8
3	0,78	2/3	4/5	8/9	45

NOTE: These code rates are also used for the imaginary part of HMmix.

One or two overall code rates shall be applied to one multiplex frame. When using two overall code rates, both shall belong to the same constellation.

Annex J provides the number of input bits per multiplex frame for EEP.

7.5.1.2 HMsym

Two protection levels are possible resulting in the use of two overall code rates. The number of input bits L_{MUX} per multiplex frame is calculated with the formulas of clause 7.2.

The MSC shall be encoded according to clause 7.3. The overall code rates and code rates for each level for the SPP are defined in table 64 and for the VSPP in table 65. The protection level is signalled in the multiplex description data entity of the SDC (see clause 6.4.3.1).

Four overall code rates are defined for the SPP as follows.

Table 64: Code rate combinations for the SPP of MSC with HMsym 64-QAM

Protection level	R_{all}	R_1	R_2	RY_{lcm}
0	0,45	3/10	3/5	10
1	0,55	4/11	8/11	11
2	0,72	4/7	7/8	56
3	0,78	2/3	8/9	9

NOTE: These code rates are also used for the real part of HMmix.

Four overall code rates are defined independently for the VSPP as follows.

Table 65: Code rate combinations for the VSPP of MSC with HMsym 64-QAM

Protection level	R_0
0	1/2
1	4/7
2	3/5
3	2/3

NOTE: These code rates are also used for the real part of HMmix.

Annex J provides the number of input bits per multiplex frame for EEP.

7.5.1.3 HMmix

Two protection levels are possible resulting in the use of two overall code rates. The number of input bits L_{MUX} per multiplex frame is calculated with the formulas of clause 7.2.

The MSC shall be encoded according to clause 7.3. The protection level is signalled in the multiplex description data entity of the SDC (see clause 6.4.3.1).

Four overall code rates are defined for the SPP as shown in table 66 and the four possible code rates for the VSPP are defined independently as shown in table 65.

Table 66: Code rate combinations for the SPP of MSC with HMmix 64-QAM

Protection level	R_{all}	R_0^{Im}	R_1^{Re}	R_1^{Im}	R_2^{Re}	R_2^{Im}	RY_{lcm}
0	0,48	1/4	3/10	1/2	3/5	3/4	20
1	0,58	1/3	4/11	2/3	8/11	4/5	165
2	0,71	1/2	4/7	3/4	7/8	7/8	56
3	0,78	2/3	2/3	4/5	8/9	8/9	45

Annex J provides the number of input bits per multiplex frame for EEP.

7.5.2 Coding the SDC

The SDC may use either 16-QAM or 4-QAM mapping. 16-QAM provides greater capacity whereas 4-QAM provides a more robust error performance. In each case, a fixed code rate is applied.

The constellation should be chosen with respect to the MSC parameters to provide more robustness for the SDC than for the MSC. When using hierarchical modulation, the SDC shall be coded using 4-QAM.

The number of input bits L_{SDC} per SDC block is calculated as given in clause 7.2.

For 16-QAM the combination given in table 67 shall be used.

Table 67: Code rate combinations for the SDC with 16-QAM

R_{all}	R_0	R_1
0,5	1/3	2/3

For 4-QAM the code rate given in table 68 shall be used.

Table 68: Code rate for the SDC with 4-QAM

R_{all}	R_0
0,5	1/2

Annex J provides the number of input bits per SDC block.

Error detection with a CRC is described in clause 6.

7.5.3 Coding the FAC

The FAC shall use 4-QAM mapping. A fixed code rate shall be applied.

The number of input bits L_{FAC} per FAC block is calculated as given in clause 7.2.

The code rate given in table 69 shall be used.

Table 69: Code rate for the FAC

R_{all}	R_0
0,6	3/5

Error detection with a CRC is described in clause 6.

7.6 MSC cell interleaving

A cell-wise interleaving shall be applied to the QAM symbols (cells) of the MSC after multilevel encoding with the possibility to choose low or high interleaving depth (denoted here as short or long interleaving) according to the predicted propagation conditions. The basic interleaver parameters are adapted to the size of a multiplex frame which corresponds to N_{MUX} cells.

For propagation channels with moderate time-selective behaviour (typical ground wave propagation in LF and MF) the short interleaving provides sufficient time- and frequency diversity for proper operation of the decoding process in the receiver (spreading of error bursts). The same block interleaving scheme as used for bit interleaving in the multilevel encoder (see clause 7.3.3) is always applied to the N_{MUX} cells of a multiplex frame.

The input vector of the block interleaver corresponding to the N_{MUX} QAM cells $z_{n,i}$ of multiplex frame n is given by:

$$\mathbf{Z}_n = (z_{n,0}, z_{n,1}, z_{n,2}, \dots, z_{n, N_{MUX}-1})$$

The output vector with the same number of cells or elements, respectively, is given by:

$$\hat{\mathbf{Z}}_n = (\hat{z}_{n,0}, \hat{z}_{n,1}, \hat{z}_{n,2}, \dots, \hat{z}_{n, N_{MUX}-1})$$

where the output elements are selected from the input elements according to:

$$\hat{z}_{n,i} = z_{n, \Pi(i)}$$

The permutation $\Pi(i)$ is obtained from the following relations:

$$s = 2^{\lceil \log_2(N_{MUX}) \rceil}, \lceil \cdot \rceil \text{ means round towards plus infinity;}$$

$$q = s / 4 - 1;$$

$$t_0 = 5;$$

$$\Pi(0) = 0;$$

for $i = 1, 2, \dots, N_{MUX} - 1$:

$$\Pi(i) = (t_0 \Pi(i-1) + q) \pmod{s};$$

while $\Pi(i) \geq N_{MUX}$:

$$\Pi(i) = (t_0 \Pi(i) + q) \pmod{s}.$$

For severe time- and frequency-selective channels as being typical for signal transmissions in the HF short wave frequency bands the interleaving depth can be increased by an additional simple convolutional interleaving scheme. For this the interleaving depth D is defined in integer multiples of multiplex frames. As a good trade-off between performance and processing delay a value of $D = 5$ has been chosen.

The output vector for long interleaving with N_{MUX} cells carrying complex QAM symbols is computed in almost the same way as for short interleaving. The only exception is that the permutation is based not only on the current but also on the last $D-1$ multiplex frames. The permutation $\Pi(i)$ as defined before is used again to determine the relation between the indices within the output vector $\hat{\mathbf{Z}}_n$ and the D input vectors $\mathbf{Z}_n, \mathbf{Z}_{n-1}, \dots, \mathbf{Z}_{n-D+1}$.

The output elements are selected from the input elements according to:

$$\hat{z}_{n,i} = z_{n-\Gamma(i), \Pi(i)}$$

For given value i the selection of the input vector number $n - \Gamma(i)$ for the correspondent element $\Pi(i)$ is done with the following formula:

$$\Gamma(i) = i \pmod{D} \text{ for } i = 0, 1, 2, \dots, N_{MUX} - 1.$$

Taking into consideration the transmission of the full content of a multiplex frame the overall delay of the pure interleaving/deinterleaving process is given by approximately 2×400 ms, i.e. 800 ms, for the short interleaving. In the case of the long interleaving it corresponds to about 2,4 s. In addition to that the delay is increased during transmission due to the fact that the SDC block is inserted at the beginning of a transmission super frame. With a depth of $D = 5$ multiplex frames for long interleaving the maximum additional increase in delay is given by the time duration of two SDC blocks.

7.7 Mapping of MSC cells on the transmission super frame structure

The content of three consecutive interleaved multiplex frames (with N_{MUX} QAM cells each) is mapped onto a transmission super frame, i.e. the corresponding number N_{SFU} of useful MSC cells is fixed as an integer multiple of 3. Due to the fact that the number of FAC and synchronization cells is varying from OFDM symbol to OFDM symbol a small loss N_L of 1 or 2 cells can occur compared with the number of available cells in a transmission super frame which is given by:

$$N_{SFA} = N_{SFU} + N_L = 3 \times N_{MUX} + N_L$$

Tables 70 to 73 give the values for the different robustness modes and bandwidths.

Table 70: Number of QAM cells for MSC for Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
Number of available MSC cells per super frame N_{SFA}	3 778	4 268	7 897	8 877	16 394	18 354
Number of useful MSC cells per super frame N_{SFU}	3 777	4 266	7 896	8 877	16 392	18 354
Number of MSC cells per multiplex frame N_{MUX}	1 259	1 422	2 632	2 959	5 464	6 118
Cell loss per super frame N_L	1	2	1	0	2	0

Table 71: Number of QAM cells for MSC for Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
Number of available MSC cells per super frame N_{SFA}	2 900	3 330	6 153	7 013	12 747	14 323
Number of useful MSC cells per super frame N_{SFU}	2 898	3 330	6 153	7 011	12 747	14 322
Number of MSC cells per multiplex frame N_{MUX}	966	1 110	2 051	2 337	4 249	4 774
Cell loss per super frame N_L	2	0	0	2	0	1

Table 72: Number of QAM cells for MSC for Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
Number of available MSC cells per super frame N_{SFA}	-	-	-	5 532	-	11 603
Number of useful MSC cells per super frame N_{SFU}	-	-	-	5 532	-	11 601
Number of MSC cells per multiplex frame N_{MUX}	-	-	-	1 844	-	3 867
Cell loss per super frame N_L	-	-	-	0	-	2

Table 73: Number of QAM cells for MSC for Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
Number of available MSC cells per super frame N_{SFA}	-	-	-	3 679	-	7 819
Number of useful MSC cells per super frame N_{SFU}	-	-	-	3 678	-	7 818
Number of MSC cells per multiplex frame N_{MUX}	-	-	-	1 226	-	2 606
Cell loss per super frame N_L	-	-	-	1	-	1

So the overall data vector for the useful MSC cells in transmission super frame m can be described by:

$$\begin{aligned}
\mathbf{S}_m &= (s_{m,0}, s_{m,1}, s_{m,2}, \dots, s_{m,N_{SFU}-1}) \\
&= (\hat{\mathbf{Z}}_{3^*m}, \hat{\mathbf{Z}}_{3^*m+1}, \hat{\mathbf{Z}}_{3^*m+2}) \\
&= (\hat{z}_{3^*m,0}, \hat{z}_{3^*m,1}, \dots, \hat{z}_{3^*m,N_{MUX}-1}, \hat{z}_{3^*m+1,0}, \hat{z}_{3^*m+1,1}, \dots, \hat{z}_{3^*m+1,N_{MUX}-1}, \hat{z}_{3^*m+2,0}, \hat{z}_{3^*m+2,1}, \dots, \hat{z}_{3^*m+2,N_{MUX}-1}).
\end{aligned}$$

In the case that N_L is unequal to 0 one or two dummy cells, i.e. $(\tilde{z}_{m,0})$ or $(\tilde{z}_{m,0}, \tilde{z}_{m,1})$, are attached at the end of \mathbf{S}_m . Their complex values (i.e. the corresponding QAM symbols) are as defined in table 74.

Table 74: QAM symbols for MSC dummy cells

Number of dummy cells N_L per transmission super frame	Complex values of the dummy cells (QAM symbols)	
	$\tilde{z}_{m,0}$	$\tilde{z}_{m,1}$
1	$a \times (1 + j 1)$	
2	$a \times (1 + j 1)$	$a \times (1 - j 1)$

The value of a in table 74 is dependent on the signal constellation chosen for the MSC (see clause 7.4).

8 Transmission structure

8.1 Transmission frame structure and modes

The transmitted signal is organized in transmission super frames.

Each transmission super frame consists of three transmission frames.

Each transmission frame has duration T_f , and consists of N_s OFDM symbols.

Each OFDM symbol is constituted by a set of K carriers and transmitted with a duration T_s .

The spacing between adjacent carriers is $1/T_u$.

The symbol duration is the sum of two parts:

- a useful part with duration T_u ;
- a guard interval with duration T_g .

The guard interval consists in a cyclic continuation of the useful part, T_u , and is inserted before it.

The OFDM symbols in a transmission frame are numbered from 0 to $N_s - 1$.

All symbols contain data and reference information.

Since the OFDM signal comprises many separately modulated carriers, each symbol can in turn be considered to be divided into cells, each cell corresponding to the modulation carried on one carrier during one symbol.

An OFDM frame contains:

- pilot cells;
- control cells;
- data cells.

The pilot cells can be used for frame, frequency and time synchronization, channel estimation, and robustness mode identification.

The transmitted signal is described by the following expression:

$$x(t) = \text{Re} \left\{ e^{j2\pi f_R t} \sum_{r=0}^{\infty} \sum_{s=0}^{N_s-1} \sum_{k=K_{\min}}^{K_{\max}} c_{r,s,k} \psi_{r,s,k}(t) \right\}$$

where:

$$\psi_{r,s,k}(t) = \begin{cases} e^{j2\pi \frac{k}{T_u}(t - Tg - sT_s - N_s r T_s)} & (s + N_s r)T_s \leq t \leq (s + N_s r + 1)T_s \\ 0 & \text{otherwise} \end{cases}$$

and:

- N_s number of OFDM symbols per transmission frame;
- k denotes the carrier number ($= K_{\min}, \dots, K_{\max}$);
- s denotes the OFDM symbol number ($= 0 \dots N_s - 1$);
- r denotes the transmission frame number ($= 0 \dots \text{infinity}$);
- K is the number of transmitted carriers ($\leq K_{\max} - K_{\min}$);
- T_s is the symbol duration;
- T_u is the duration of the useful part of a symbol;
- T_g is the duration of the guard interval;
- f_R is the reference frequency of the RF signal;
- $c_{r,s,k}$ complex cell value for carrier k in symbol s of frame number r .

The $c_{r,s,k}$ values depend on the type of cell, as defined below.

For data/control cells (MSC, SDC, FAC), $c_{r,s,k} = z$ where z is the constellation point for each cell as given by the mappings defined in clause 7.

For each reference cell, a defined phase and amplitude is transmitted, $c_{r,s,k} = a_{s,k} U_{s,k}$, where:

$a_{s,k}$ is the amplitude, which always takes one of the values $\{1, \sqrt{2}, 2\}$, and

$U_{s,k} = e^{j2\pi \vartheta_{s,k}}$ is a unit-amplitude term of phase $\vartheta_{s,k}$.

$a_{s,k}$ and $\vartheta_{s,k}$ are defined for each type of reference cell in clause 8.4.

8.2 Propagation-related OFDM parameters

OFDM parameters must be chosen to match propagation conditions and the coverage area that the operator wants to serve.

Various sets of OFDM parameters are therefore defined for different conditions of propagation and their parameter values are listed in table 75.

Table 75: Numerical values of the OFDM parameters

Robustness mode	Duration T_u	Carrier spacing $1/T_u$	Duration of guard interval T_g	Duration of symbol $T_s = T_u + T_g$	T_g/T_u	Number of symbols per frame N_s
A	24 ms	$41^{2/3}$ Hz	2,66 ms	26,66 ms	1/9	15
B	21,33 ms	$46^{7/8}$ Hz	5,33 ms	26,66 ms	1/4	15
C	14,66 ms	$68^{2/11}$ Hz	5,33 ms	20 ms	4/11	20
D	9,33 ms	$107^{1/7}$ Hz	7,33 ms	16,66 ms	11/14	24

8.3 Signal bandwidth related parameters

8.3.1 Parameter definition

The OFDM parameters depend upon the available frequency bandwidth, the number of carriers K , and their location with respect to the reference frequency (named DC, in relation with the traditional carrier used in AM analogue transmissions).

The Spectrum occupancy defines the nominal channel bandwidth. The group of carriers carrying the FAC is always to the right (higher in frequency) with respect to the reference frequency, f_R , which is an integer multiple of 1 kHz.

Table 76 relates the spectrum occupancy parameter, signalled in the FAC (see clause 6.3), to the nominal channel bandwidth, and figures 30 and 31 show the position of the carriers.

Table 76: Relationship between spectrum occupancy parameter and channel bandwidth

Channel bandwidth (kHz)	Spectrum occupancy					
	0	1	2	3	4	5
	4,5	5	9	10	18	20

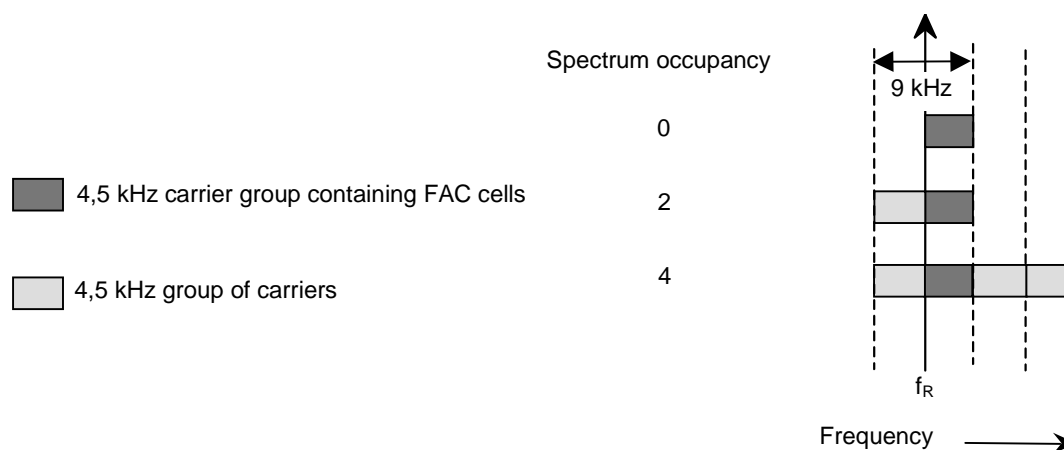


Figure 30: Spectrum occupancy for 9 kHz channels

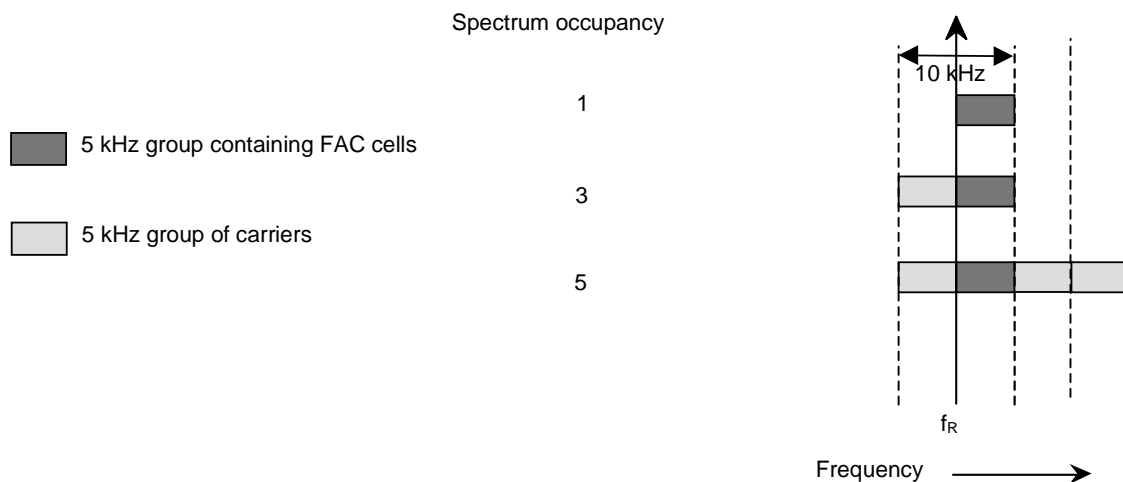


Figure 31: Spectrum occupancy for 10 kHz channels

The carriers are indexed by $k \in [K_{\min}, K_{\max}]$, $k = 0$ being the DC carrier and determined by the following values depending on the choice made and related to the occupied bandwidth.

Carriers with $k < 0$ are said to be to the left of DC, and $k > 0$, to the right of DC.

Table 77 presents the lower and upper carrier numbers for each robustness mode and nominal bandwidth.

Table 77: Carrier numbers for each mode

Robustness mode	Carrier	Spectrum occupancy					
		0	1	2	3	4	5
A	K_{\min}	2	2	-102	-114	-98	-110
	K_{\max}	102	114	102	114	314	350
B	K_{\min}	1	1	-91	-103	-87	-99
	K_{\max}	91	103	91	103	279	311
C	K_{\min}	-	-	-	-69	-	-67
	K_{\max}	-	-	-	69	-	213
D	K_{\min}	-	-	-	-44	-	-43
	K_{\max}	-	-	-	44	-	135

Depending on the mode, the DC carrier and certain carriers around DC are not used, as detailed in table 78.

Table 78: Unused carriers according to robustness mode

Robustness mode	Unused carrier(s)
A	$k \in \{-1, 0, 1\}$
B	$k \in \{0\}$
C	$k \in \{0\}$
D	$k \in \{0\}$

8.3.2 Simulcast transmission

The DRM signal is designed to work in the same broadcast bands as AM signals. Simulcast transmission of services using DRM and AM can be performed by the juxtaposition of the analogue AM signal (DSB or VSB or SSB) and a DRM digital signal. Many arrangements are possible and some are illustrated in annex K.

The spectrum occupancy number relates to the DRM signal. A broadcaster may choose to signal the presence of the AM simulcast by use of the Alternative frequency signalling: Other services data entity in the SDC (see clause 6.4.3.12).

8.4 Pilot cells

8.4.1 Functions and derivation

Some cells within the OFDM transmission frame are modulated with known fixed phases and amplitudes.

These cells are pilot cells for channel estimation and synchronization. The positions, amplitudes and phases of these cells are carefully chosen to optimize the performance, especially the initial synchronization duration and reliability.

The phases are defined, directly or indirectly, in $1/1024^{\text{th}}$ of a cycle, i.e. $U_{s,k} = e^{j2\pi\vartheta_{s,k}} = e^{\frac{j2\pi\vartheta_{1024}[s,k]}{1024}}$, where $\vartheta_{1024}[s,k]$ takes integer values and is either explicitly tabulated or derived using integer arithmetic, as defined in the following clauses.

8.4.2 Frequency references

These cells are used by the receiver to detect the presence of the received signal and to estimate its frequency offset. They may also be used for channel estimation and various tracking processes.

8.4.2.1 Cell positions

Frequency references are located at frequencies which are common to all the system variants.

There are three frequency references, which are 750 Hz, 2 250 Hz and 3 000 Hz as referenced to the DC carrier, as defined in table 79.

Table 79: Carrier numbers for frequency references

Robustness mode	Carrier numbers
A	18, 54, 72
B	16, 48, 64
C	11, 33, 44
D	7, 21, 28

They shall be present in all symbols of each transmission frame.

8.4.2.2 Cell gains and phases

All frequency reference cells shall have a power gain of 2, i.e. $a_{s,k} = \sqrt{2}$, in order to optimize performances at low signal to noise ratio and be compatible when the same cell functions as both a frequency reference and a time reference.

The phases are defined as follows. For the first symbol in the frame (i.e. $s = 0$), the phases $\vartheta_{1024}[s, k]$ are given in table 80.

Table 80: Cell phases for frequency references

Robustness mode	Carrier index, k	Phase index, $\vartheta_{1024}[0, k]$
A	18	205
	54	836
	72	215
B	16	331
	48	651
	64	555
C	11	214
	33	392
	44	242
D	7	788
	21	1 014
	28	332

For subsequent symbols, the phases are chosen in order to ensure the tones are continuous, which is achieved by applying the following rules.

For robustness modes A, B and C, and carrier 28 only of mode D:

$$\vartheta_{1024}[s, k] = \vartheta_{1024}[0, k]$$

For robustness mode D, carriers 7 and 21:

$$\vartheta_{1024}[s, k] = \vartheta_{1024}[0, k], \text{ for even values of } s, \text{ and}$$

$$\vartheta_{1024}[s, k] = (\vartheta_{1024}[0, k] + 512) \bmod 1024, \text{ for odd values of } s.$$

NOTE: This is equivalent to the complex value $U_{s,k}$ multiplied by -1 for odd values of s.

8.4.3 Time references

These cells are located in the first OFDM symbol of each transmission frame, i.e. $s = 0$.

The time reference cells are mainly used for performing ambiguity resolution since guard time correlation provides a fast and frequency insensitive estimation of time of arrival with a periodicity of one symbol. They are used for determining the first symbol of a transmission frame. They can also be used for frequency-offset estimation.

8.4.3.1 Cell positions and phases

Tables 81 to 84 define the phases of the time reference cells, and the phases of the frequency reference cells for the first symbol of the transmission frame.

$\vartheta_{1024}[0, k]$ is the phase index in 1 024^{ths} of a cycle.

Table 81: Phase of time reference cells for Mode A

Carrier index, k	Phase index, $\vartheta_{1024}[0, k]$
17	973
18 *	205
19	717
21	264
28	357
29	357
32	952
33	440
39	856
40	88
41	88
53	68
54 *	836
55	836
56	836
60	1 008
61	1 008
63	752
71	215
72 *	215
73	727
NOTE: Carrier numbers marked with an asterisk "*" also serve as frequency references (see clause 8.4.2.1); the definitions of phase index are consistent.	

Table 82: Phase of time reference cells for Mode B

Carrier index k	Phase index, $\vartheta_{1024}[0, k]$
14	304
16 *	331
18	108
20	620
24	192
26	704
32	44
36	432
42	588
44	844
48 *	651
49	651
50	651
54	460
56	460
62	944
64 *	555
66	940
68	428
NOTE: Carrier numbers marked with an asterisk "*" also serve as frequency references (see clause 8.4.2.1); the definitions of phase index are consistent.	

Table 83: Phase of time reference cells for Mode C

Carrier index k	Phase index, $\vartheta_{1024}[0,k]$
8	722
10	466
11 *	214
12	214
14	479
16	516
18	260
22	577
24	662
28	3
30	771
32	392
33 *	392
36	37
38	37
42	474
44 *	242
45	242
46	754

NOTE: Carrier numbers marked with an asterisk "*" also serve as frequency references (see clause 8.4.2.1); the definitions of phase index are consistent.

Table 84: Phase of time reference cells for Mode D

Carrier index k	Phase index, $\vartheta_{1024}[0,k]$
5	636
6	124
7 *	788
8	788
9	200
11	688
12	152
14	920
15	920
17	644
18	388
20	652
21 *	1 014
23	176
24	176
26	752
27	496
28 *	332
29	432
30	964
32	452

NOTE: Carrier numbers marked with an asterisk "*" also serve as frequency references (see clause 8.4.2.1); the definitions of phase index are consistent.

8.4.3.2 Cell gains

All time reference cells have a power gain of 2,0 in order to optimize performance at low signal to noise ratio, i.e. $a_{s,k} = \sqrt{2}$.

8.4.4 Gain references

The gain reference cells are mainly used for coherent demodulation. These cells are scattered throughout the overall time frequency pattern and are used by the receiver to estimate the channel response.

8.4.4.1 Cell positions

In a transmission frame, for the symbol of index s (ranging from 0 to $N_s - 1$), carriers for which index k belongs to the subsets as defined in table 85 are gain references.

Table 85: Carrier indices k for gain reference cells

Robustness mode	Subset	Condition	Periodicity of the gain reference pattern
A	$k = 2 + 4 \times (s \bmod 5) + 20 \times p$	p integer $k_{\min} \leq k \leq k_{\max}$	5 symbols
B	$k = 1 + 2 \times (s \bmod 3) + 6 \times p$	p integer $k_{\min} \leq k \leq k_{\max}$	3 symbols
C	$k = 1 + 2 \times (s \bmod 2) + 4 \times p$	p integer $k_{\min} \leq k \leq k_{\max}$	2 symbols
D	$k = 1 + (s \bmod 3) + 3 \times p$	p integer $k_{\min} \leq k \leq k_{\max}$	3 symbols

NOTE: The gain reference cell patterns have been chosen such that the edge carriers are included as gain reference cell positions.

Annex L gives some example figures illustrating the position of the gain reference cells.

8.4.4.2 Cell gains

Gain reference cells mostly have a power gain of 2 (i.e. $a_{s,k} = \sqrt{2}$), in order to optimize performances at low signal to noise ratio. However, gain reference cells close to the band lower and upper edges are over-boosted by a further power gain of 2 (i.e. overall power gain of 4, so that the amplitude $a_{s,k} = 2$) as defined in table 86.

Table 86: Carrier numbers given a power boost of 4, i. e. $a_{s,k} = 2$

Robustness mode	Spectrum occupancy					
	0	1	2	3	4	5
A	2, 6, 98, 102	2, 6, 110, 114	-102, -98, 98, 102	-114, -110, 110, 114	-98, -94, 310, 314	-110, -106, 346, 350
B	1, 3, 89, 91	1, 3, 101, 103	-91, -89, 89, 91	-103, -101, 101, 103	-87, -85, 277, 279	-99, -97, 309, 311
C	-	-	-	-69, -67, 67, 69	-	-67, -65, 211, 213
D	-	-	-	-44, -43, 43, 44	-	-43, -42, 134, 135

8.4.4.3 Cell phases

In some cases gain references fall in locations which coincide with those already defined for either frequency or time references. In these cases, the phase definitions given in clauses 8.4.2 and 8.4.3 take precedence.

In all other locations, the phases of the gain reference cells are obtained by integer arithmetic applied to small tables of values, as defined in the following procedure.

8.4.4.3.1 Procedure for calculation of cell phases

The procedure is:

First, compute values of m , n and p for each cell, where the carrier number is k and the symbol number is s :

$$\begin{aligned} n &= s \bmod y, \\ m &= \lfloor s / y \rfloor \\ p &= \frac{k - k_0 - nx}{xy} \end{aligned}$$

x , y , and k_0 are constants which are defined for each robustness mode in table 87.

Table 87: Definition of x , y , k_0

Robustness mode	x	y	k_0
A	4	5	2
B	2	3	1
C	2	2	1
D	1	3	1

NOTE 1: The value of p obtained by this procedure is an integer, as a consequence of the definition of reference cell locations in clause 8.4.4.1; while the values of n and m are integer by definition of the mathematical operations producing them.

Secondly, calculate the integer phase index by the following formula:

$$\vartheta_{1024}[s, k] = (4Z_{256}[n, m] + pW_{1024}[n, m] + p^2(1 + s)Q_{1024}) \bmod 1024$$

Q_{1024} and the small tables $Z_{256}[n, m]$ and $W_{1024}[n, m]$ are defined for each robustness mode in the following clauses.

NOTE 2: The values in the tables $Z_{256}[n, m]$ and $W_{1024}[n, m]$ may be represented precisely as 8-bit and 10-bit unsigned integers respectively; similarly Q_{1024} may be represented precisely as a 10-bit unsigned integer.

8.4.4.3.2 Robustness Mode A

The $W_{1024}[n, m]$ matrix is defined as:

$W_{1024}[n, m] = \{$	{228,	341,	455},
	{455,	569,	683},
	{683,	796,	910},
	{910,	0,	114},
	{114,	228,	341}}

The $Z_{256}[n, m]$ matrix is defined as:

$Z_{256}[n, m] = \{$	{0,	81,	248},
	{18,	106,	106},
	{122,	116,	31},
	{129,	129,	39},
	{33,	32,	111}}

$$Q_{1024} = 36.$$

8.4.4.3.3 Robustness Mode B

The $W_{1024}[n, m]$ matrix is defined as:

$W_{1024}[n, m] = \{$	{512,	0,	512,	0,	512},
	{0,	512,	0,	512,	0},
	{512,	0,	512,	0,	512}}

The $Z_{256}[n, m]$ matrix is defined as:

$Z_{256}[n, m] = \{$	{0,	57,	164,	64,	12},
	{168,	255,	161,	106,	118},
	{25,	232,	132,	233,	38}}

$$Q_{1024} = 12.$$

8.4.4.3.4 Robustness Mode C

The $W_{1024}[n, m]$ matrix is defined as:

$W_{1024}[n, m] = \{$	{465,	372,	279,	186,	93,	0,	931,	838,	745,	652},
	{931,	838,	745,	652,	559,	465,	372,	279,	186,	93}}

The $Z_{256}[n, m]$ matrix is defined as:

$Z_{256}[n, m] = \{$	{0,	76,	29,	76,	9,	190,	161,	248,	33,	108},
	{179,	178,	83,	253,	127,	105,	101,	198,	250,	145}}

$$Q_{1024} = 12.$$

8.4.4.3.5 Robustness Mode D

The $W_{1024}[n, m]$ matrix is defined as:

$W_{1024}[n, m] = \{$	{366,	439,	512,	585,	658,	731,	805,	878},
	{731,	805,	878,	951,	0,	73,	146,	219},
	{73,	146,	219,	293,	366,	439,	512,	585}}

The $Z_{256}[n, m]$ matrix is defined as:

$Z_{256}[n, m] = \{$	{0,	240,	17,	60,	220,	38,	151,	101},
	{110,	7,	78,	82,	175,	150,	106,	25},
	{165,	7,	252,	124,	253,	177,	197,	142}}

$$Q_{1024} = 14.$$

8.5 Control cells

8.5.1 General

The control cells consist of two parts:

- the Fast Access Channel (FAC), integrated in every transmission frame. It is used to quickly obtain the necessary information for the receiver to be able to demodulate the DRM signal;
- the Service Description Channel (SDC), repeated every transmission super-frame. It contains all the additional information that describes the services currently available. The SDC is also used for Alternative Frequency Switching (AFS).

Figure 32 describes the time-frequency location of these signals.

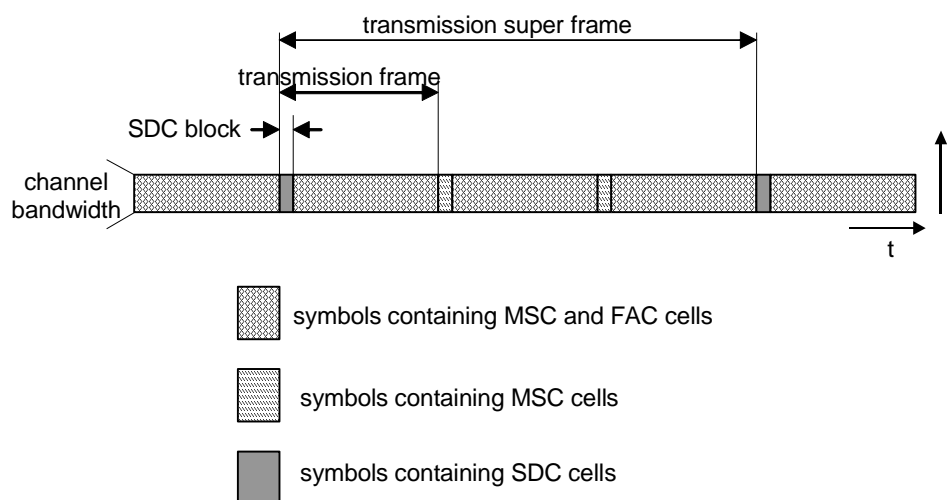


Figure 32: Time-frequency location of FAC and SDC signals

8.5.2 FAC cells

8.5.2.1 Cell positions

The cells used for FAC are cells that are neither frequency references, nor time references, nor gain references, nor data cells in the symbols that do not contain the SDC.

FAC cells convey highly protected QAM symbols that allow fast detection by the receiver of the type of signal it is currently receiving.

There are always 65 FAC cells. Tables 88 to 91 give the position of the FAC cells for each robustness mode.

Table 88: Position of the FAC cells in robustness Mode A

Symbol	Carrier number
0	
1	
2	26, 46, 66, 86
3	10, 30, 50, 70, 90
4	14, 22, 34, 62, 74, 94
5	26, 38, 58, 66, 78
6	22, 30, 42, 62, 70, 82
7	26, 34, 46, 66, 74, 86
8	10, 30, 38, 50, 58, 70, 78, 90
9	14, 22, 34, 42, 62, 74, 82, 94
10	26, 38, 46, 66, 86
11	10, 30, 50, 70, 90
12	14, 34, 74, 94
13	38, 58, 78
14	

Table 89: Position of the FAC cells in robustness Mode B

Symbol	Carrier number
0	
1	
2	13, 25, 43, 55, 67
3	15, 27, 45, 57, 69
4	17, 29, 47, 59, 71
5	19, 31, 49, 61, 73
6	9, 21, 33, 51, 63, 75
7	11, 23, 35, 53, 65, 77
8	13, 25, 37, 55, 67, 79
9	15, 27, 39, 57, 69, 81
10	17, 29, 41, 59, 71, 83
11	19, 31, 43, 61, 73
12	21, 33, 45, 63, 75
13	23, 35, 47, 65, 77
14	

Table 90: Position of the FAC cells in robustness Mode C

Symbol	Carrier number
0	
1	
2	
3	9, 21, 45, 57
4	23, 35, 47
5	13, 25, 37, 49
6	15, 27, 39, 51
7	5, 17, 29, 41, 53
8	7, 19, 31, 43, 55
9	9, 21, 45, 57
10	23, 35, 47
11	13, 25, 37, 49
12	15, 27, 39, 51
13	5, 17, 29, 41, 53
14	7, 19, 31, 43, 55
15	9, 21, 45, 57
16	23, 35, 47
17	13, 25, 37, 49
18	15, 27, 39, 51
19	

Table 91: Position of the FAC cells in robustness Mode D

Symbol	Carrier number
0	
1	
2	
3	9, 18, 27
4	10, 19
5	11, 20, 29
6	12, 30
7	13, 22, 31
8	5, 14, 23, 32
9	6, 15, 24, 33
10	16, 25, 34
11	8, 17, 26, 35
12	9, 18, 27, 36
13	10, 19, 37
14	11, 20, 29
15	12, 30
16	13, 22, 31
17	5, 14, 23, 32
18	6, 15, 24, 33
19	16, 25, 34
20	8, 17, 26, 35
21	9, 18, 27, 36
22	10, 19, 37
23	

8.5.2.2 Cell gains and phases

The $c_{r,s,k}$ values are normalized modulation values of the constellation point z according to the modulation alphabet used for the FAC (4-QAM) (see figure 29).

Successive constellation points are assigned to the FAC cells of a transmission frame in order of increasing carrier index k , starting from the most negative k ; then in time order starting from the first FAC bearing symbol of the frame.

8.5.3 SDC cells

8.5.3.1 Cell positions

The cells used for SDC are all the cells in the SDC symbols which are neither frequency references, nor time references, nor gain references for which $k_{\min} \leq k \leq k_{\max}$ and k does not belong to the subset of unused carriers defined above.

For robustness modes A and B, the SDC symbols are symbols 0 and 1 of each transmission super frame. For robustness modes C and D, the SDC symbols are symbols 0, 1 and 2 of each transmission super frame.

8.5.3.2 Cell gains and phases

The $c_{r,s,k}$ values are normalized modulation values of the constellation point z according to the modulation alphabet used for the SDC (16- or 4-QAM, see figures 28 and 29).

Successive constellation points are assigned to the SDC cells of a transmission super frame in order of increasing carrier index k , starting from the most negative k ; then in time order starting from the first SDC bearing symbol of the super frame.

8.6 Data cells

8.6.1 Cell positions

Data cells are all cells which are neither pilot cells, nor control cells; for which $k_{\min} \leq k \leq k_{\max}$ and k does not belong to the subset of unused carriers defined above.

8.6.2 Cell gains and phases

The $c_{r,s,k}$ values are the normalized modulation values of the constellation point z according to the modulation alphabet used for the MSC (64- or 16-QAM, see figures 25 to 28) taken from the vector S_m (see clause 7.7).

Successive elements $s_{m,i}$ are assigned to the cells of a transmission super frame in order of increasing carrier index k , starting from the most negative k ; then in time order starting from the first non-SDC symbol of the super frame.

Annex A (informative): Simulated system performance

Table A.1 gives simulated system performance anticipating perfect channel estimation and the absence of phase noise and quantization effects. Channel decoding is assumed to be done with a multistage decoder with two iterations.

The results are given for five of the channels of annex B, whereby the associated OFDM modes are the Mode A for the channels 1 and 2, the Mode B mode for the channels 3 to 5. The associated code rate is $R = 0,6$ and the modulation is 64-QAM. The signal power includes pilots and the guard interval.

Table A.1: Required S/N for a transmission to achieve a BER = 1×10^{-4} after the channel decoder for the MSC

Channel model	Code rate R = 0,6
Channel 1	14,9 dB
Channel 2	16,5 dB
Channel 3	23,2 dB
Channel 4	22,3 dB
Channel 5	20,4 dB

Further results for other combinations of DRM transmission and service parameters (including real channel estimation behaviour of the receiver) can be found in ITU-R Recommendation BS.1615 [9].

Annex B (informative): Definition of channel profiles

The channels to be considered are the LF, MF and HF broadcast radio transmission channels. In principle all three are multipath channels because the surface of the earth and the ionosphere are involved in the mechanism of electromagnetic wave propagation.

The approach is to use stochastic time-varying models with a stationary statistics and define models for good, moderate and bad conditions by taking appropriate parameter values of the general model. One of those models with adaptable parameters is the Wide Sense Stationary Uncorrelated Scattering model (WSSUS model). The justification for the stationary approach with different parameter sets is, that results on real channels lead to BER curves between best and worst cases found in the simulation.

The channel models have been generated from the following equations where $e(t)$ and $s(t)$ are the complex envelopes of the input and output signals respectively:

$$s(t) = \sum_{k=1}^n \rho_k c_k(t) e(t - \Delta_k) \quad (\text{B.1})$$

This is a tapped delay-line where:

- ρ_k is the attenuation of the path number k - listed in table B.1.
- Δ_k is the relative delay of the path number k - listed in table B.1
- the time-variant tap weights $\{c_k(t)\}$ are zero mean complex-valued stationary gaussian random processes. The magnitudes $|c_k(t)|$ are Rayleigh-distributed and the phases $\Phi(t)$ are uniformly distributed.

For each weight $\{c_k(t)\}$ there is one stochastic process, characterized by its variance and its power density spectrum (PDS). The variance is a measure for the average signal power which is received via this path and is defined by the relative attenuation ρ_k - listed in table B.1 - and the PDS determines the average speed of variation in time. The width of the PDS is quantified by a number and is referred to as the Doppler spread D_{sp} of that path - listed in table B.1.

There might be also a non-zero centre frequency of the PDS, which can be interpreted as an average frequency shift or Doppler shift D_{sh} - listed in table B.1.

The PDS is modelled by filtering of white noise (i.e. with constant PDS) and is equal to:

$$\phi_{n,n_t}(f) = N_0 \cdot |H(f)|^2 \quad (\text{B.2})$$

$H(f)$ is the transfer function of the filter. The stochastic processes belonging to every individual path then become Rayleigh processes. For the ionospheric path, a Gaussian shape has proven to be a good approach with respect to real observations.

The Doppler profile on each path k is then defined as:

$$|H(f)|^2 = \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{(f - D_{sh})^2}{2\sigma_d^2}} \quad (\text{B.3})$$

The Doppler spread is specified as 2-sided and contains 68 % of the power:

$$D_{sp} = 2\sigma_d \quad (\text{B.4})$$

Table B.1: Set of channels

Channel no 1: AWGN	good typical/moderate bad			LF, MF, HF LF, var.SNR
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0			
Path gain, rms (ρ_k)	1			
Doppler shift (D_{sh})	0			
Doppler spread (D_{sp})	0			

Channel no 2: Rice with delay	good typical/moderate bad			MF, HF
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0	1 ms		
Path gain, rms (ρ_k)	1	0,5		
Doppler shift (D_{sh})	0	0		
Doppler spread (D_{sp})	0	0,1 Hz		

Channel no 3: US Consortium	good typical/moderate bad			HF MF
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0	0,7 ms	1,5 ms	2,2 ms
Path gain, rms (ρ_k)	1	0,7	0,5	0,25
Doppler shift (D_{sh})	0,1 Hz	0,2 Hz	0,5 Hz	1,0 Hz
Doppler spread (D_{sp})	0,1 Hz	0,5 Hz	1,0 Hz	2,0 Hz

Channel no 4: CCIR Poor	good typical/moderate bad			HF
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0	2 ms		
Path gain, rms (ρ_k)	1	1		
Doppler shift (D_{sh})	0	0		
Doppler spread (D_{sp})	1 Hz	1 Hz		

Channel no 5	good typical/moderate bad			HF
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0	4 ms		
Path gain, rms (ρ_k)	1	1		
Doppler shift (D_{sh})	0	0		
Doppler spread (D_{sp})	2 Hz	2 Hz		

Channel no 6	good typical/moderate bad			HF
	path 1	path 2	path 3	path 4
Delay (Δ_k)	0	2 ms	4 ms	6 ms
Path gain, rms (ρ_k)	0,5	1	0,25	0,0625
Doppler shift (D_{sh})	0	1,2 Hz	2,4 Hz	3,6 Hz
Doppler spread (D_{sp})	0,1 Hz	2,4 Hz	4,8 Hz	7,2 Hz

Annex C (informative): Example of mapping of logical frames to multiplex frames

There are many service and stream combinations possible within the DRM system. One example is illustrated in this annex.

This example DRM signal contains two services: an audio service (service A) and a data service (service D). The audio service also carries a data application.

UEP is applied to the audio service. The data application carried with the audio service uses the lower protection. The data service uses the higher protection. The code rates chosen are 0,5 and 0,6 corresponding to protection level 0 and 1 respectively.

Service A consists of two streams: stream 0 carries the audio, stream 1 carries the data application.

Service D consists of one stream: stream 2.

Stream 0 is carried in logical frames L0, stream 1 is carried in logical frames L1 and stream 2 is carried in logical frames L2.

L0 has 266 bytes in the higher protected part (part A) with protection level 0, and 798 bytes in the lower protected part (part B) with protection level 1.

L1 has 59 bytes in the lower protected part (part B) with protection level 1.

L2 has 19 bytes in the higher protected part (part A) with protection level 0.

The resulting multiplex frame is illustrated in figure C.1.

Protection level 0		Protection level 1	
Stream 0	Stream 2	Stream 0	Stream 1
266 bytes	19 Bytes	798 Bytes	59 Bytes

Figure C.1

The multiplex description data entity is coded as follows:

Field name	Field size	Field value
length	7	9
version number	1	0
type	4	0
protection level for part A	2	0
protection level for part B	2	1
data length of part A (stream 0)	12	266
data length of part B (stream 0)	12	798
data length of part A (stream 1)	12	0
data length of part B (stream 1)	12	59
data length of part A (stream 2)	12	19
data length of part B (stream 2)	12	0

Annex D (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 :

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

with $n \geq 1$:

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

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 D.1). The stages are denoted by $b_0 \dots b_{n-1}$, 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".

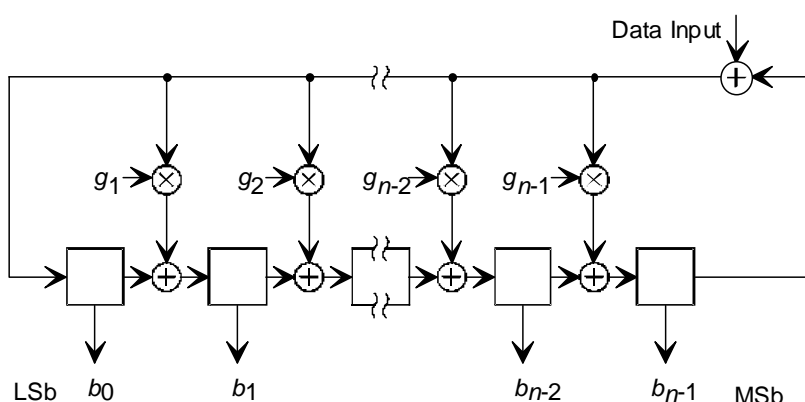


Figure D.1: General CRC block diagram

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 first. The CRC shall be inverted (1's complement) prior to transmission.

The CRC codes used in the DRM system are based on the following polynomials:

- $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$
- $G_8(x) = x^8 + x^4 + x^3 + x^2 + 1$
- $G_6(x) = x^6 + x^5 + x^3 + x^2 + x + 1$
- $G_5(x) = x^5 + x^4 + x^2 + x + 1$
- $G_3(x) = x^3 + x + 1$
- $G_2(x) = x^2 + x + 1$

- $G_1(x) = x + 1$

The assignment of the polynomials to the respective applications is given in each clause.

Annex E (informative): RF protection ratios

Protection ratios are required for:

- AM interfered with by DRM digital signals;
- DRM digital signals interfered with by AM;
- DRM digital signals interfered with by DRM digital signals.

The combinations of spectrum occupancy types and robustness modes of DRM signals lead to several transmitter RF spectra, which cause different interference and therefore require different RF protection ratios. The applied calculation method is described in detail in ITU-R Recommendation BS.1615 [9]. The differences in protection ratios for the different DRM robustness modes are quite small. Therefore, the RF protection ratios presented in the following tables are restricted to the robustness mode B. More calculation results can be found in ITU-R Recommendation BS.1615 [9].

Table E.1 shows calculation results for AM interfered with by DRM and table E.2, DRM interfered with by AM. These values are calculated for AM signals with high compression. The RF protection ratios for DRM interfered with by DRM are given in table E.3. Correction values for DRM reception using different modulation schemes and protection levels are given in table E.4.

The values in tables E.1 to E.4 represent relative RF protection ratios ($A_{RF_relative}$). For the pure AM case, the relative protection ratio is the difference in dB between the protection ratio when the carriers of the wanted and unwanted transmitters have a frequency difference of Δf Hz and the protection ratio when the carriers of these transmitters have the same frequency (ITU-R Recommendation BS.560), i.e. the co-channel RF protection ratio, A_{RF} , which corresponds to the Audio Frequency (AF) protection ratio (A_{AF}). In the case of a digital signal its nominal frequency instead of the carrier frequency is the relevant value for the determination of the frequency difference. For spectrum occupancy types 2 and 3 the nominal frequency corresponds to the centre frequency of the OFDM block, for the types 0 and 1 the centre frequency is shifted about 2,2 kHz and 2,4 kHz, respectively, above the nominal frequency. Due to the fact that the spectrum of the interference signal is different from the AF spectrum of analogue AM, the values for relative RF protection ratio in the case of co-channel interference are not equal to zero.

To adjust table E.1 to a given AM planning scenario, the relevant AF protection ratio has to be added to the values in the table to get the required RF protection ratio. Relevant values may be determined taking into account:

- for HF, the AF protection ratio of 17 dB, which was adopted for HFBC planning by WARC HFBC-87 for AM interfered with by AM;
- for LF/MF, the AF protection of 30 dB, which was adopted by the Regional Administrative LF/MF Broadcasting Conference for Regions 1 and 3 (Geneva, 1975) for AM interfered with by AM.

With DRM as the wanted signal, the AF protection ratio as a parameter for the quality of service has to be replaced by the S/I ratio required to achieve a certain BER. A BER threshold of 10^{-4} is supposed for the calculations. The protection ratio values in tables E.2 and E.3 are based on 64-QAM modulation and protection level No. 1. For other combinations the correction values in table E.4 have to be added to the S/I values given in the tables.

Table E.1: Relative RF protection ratios (dB) for AM interfered with by DRM

Wanted signal	Unwanted signal	Frequency separation $f_{unwanted} - f_{wanted}$ (kHz)													Parameters	
		-20	-18	-15	-10	-9	-5	0	5	9	10	15	18	20	B_{DRM} (kHz)	A_{AF} (dB) (see notes 3 and 4)
AM	DRM_B0 (see note 1)	-50,4	-50,4	-49	-35,5	-28,4	6,4	6,6	-30,9	-46,7	-48,2	-50,4	-50,4	-50,4	4,5	-
AM	DRM_B1 (see note 2)	-51	-50,5	-47,6	-32	-23,8	6	6	-31,1	45,7	47,4	-51	-51	-51	5	-
AM	DRM_B2	-48,8	-46,9	-43,5	-34,4	-29,7	3,4	6,5	3,4	-29,7	-34,4	-43,5	-46,9	-48,8	9	-
AM	DRM_B3	-47,2	-45,3	-41,9	-32	-25,9	3	6	3	-25,9	-32	-41,9	-45,3	-47,2	10	-

B_{DRM} : nominal bandwidth of DRM signal.
 DRM_B0: DRM signal, robustness mode B, spectrum occupancy type 0.
 NOTE 1: The centre frequency of DRM_B0 transmission is shifted about 2,2 kHz above the nominal frequency.
 NOTE 2: The centre frequency of DRM_B1 transmission is shifted about 2,4 kHz above the nominal frequency.
 NOTE 3: The RF protection ratio for AM interfered with by DRM can be calculated by adding a suitable value for the AF protection ratio according to a given planning scenario to the values in the table.
 NOTE 4: The values presented in this table refer to the specific case of high AM compression. For consistency with table E.2, the same modulation depth, namely that associated with high compression, has been assumed for the AM signal. In order to offer adequate protection to AM signals with normal levels of compression (see ITU-R Recommendation BS.1615 [9]), each value in the table should be increased to accommodate the difference between normal and high compression.

Table E.2: Relative RF protection ratios (dB) for DRM (64-QAM, protection level No. 1) interfered with by AM

Wanted signal	Unwanted signal	Frequency separation $f_{unwanted} - f_{wanted}$ (kHz)													Parameters	
		-20	-18	-15	-10	-9	-5	0	5	9	10	15	18	20	B_{DRM} (kHz)	S/I (dB)
DRM_B0 (see note 1)	AM	-57,7	-55,5	-52,2	-46,1	-45	-36,2	0	-3,5	-30,9	-41,1	-46,9	-50,6	-53	4,5	4,6
DRM_B1 (see note 2)	AM	-57,4	-55,2	-51,9	-45,9	-44,7	-36	0	-0,2	-22	-37,6	-46	-49,6	-52	5	4,6
DRM_B2	AM	-54,6	-52,4	-48,8	-42,8	-33,7	-6,4	0	-6,4	-33,7	-42,8	-48,8	-52,4	-54,6	9	7,3
DRM_B3	AM	-53,9	-51,5	-48	-39,9	-25	-3,1	0	-3,1	-25	-39,9	-48	-51,5	-53,9	10	7,3

B_{DRM} : nominal bandwidth of DRM signal.
 DRM_B0: DRM signal, robustness mode B, spectrum occupancy type 0.
 S/I: signal-to-interference ratio for a BER of 1×10^{-4} .
 NOTE 1: The centre frequency of DRM_B0 transmission is shifted about 2,2 kHz above the nominal frequency.
 NOTE 2: The centre frequency of DRM_B1 transmission is shifted about 2,4 kHz above the nominal frequency.

Table E.3: Relative RF protection ratios (dB) for DRM (64-QAM, protection level No. 1) interfered with by DRM

Wanted signal	Unwanted signal	Frequency separation $f_{unwanted} - f_{wanted}$ (kHz)													Parameters	
		-20	-18	-15	-10	-9	-5	0	5	9	10	15	18	20	B_{DRM} (kHz)	S/I (dB)
DRM_B0	DRM_B0	-60	-59,9	-60	-55,2	-53,2	-40,8	0	-40,8	-53,2	-55,2	-60	-59,9	-60	4,5	16,2
DRM_B0	DRM_B1	-60,1	-60	-59,5	-52,5	-50,4	-37,4	0	-40	-51,6	-53,6	-59,8	-60	-60,1	5	15,7
DRM_B0	DRM_B2	-57,4	-55,7	-52,9	-46,7	-45,1	-36,6	0	-0,8	-35,6	-38,4	-47,7	-51,5	-53,6	9	13,2
DRM_B0	DRM_B3	-55,2	-53,6	-50,7	-44,5	-42,9	-33,1	0	-0,1	-13,6	-36,2	-45,5	-49,3	-51,4	10	12,6
DRM_B1	DRM_B0	-59,4	-59,5	-59,5	-55	-53	-40,8	0	-37,9	-51,7	-53,9	-59,4	-59,5	-59,4	4,5	16,2
DRM_B1	DRM_B1	-60	-60	-59,5	-52,8	-50,8	-37,8	0	-37,8	-50,8	-52,8	-59,5	-60	-60	5	16,2
DRM_B1	DRM_B2	-57,1	-55,4	-52,6	-46,4	-44,9	-36,4	0	-0,1	-13,7	-36,8	-46,6	-50,5	-52,7	9	13,2
DRM_B1	DRM_B3	-55,5	-53,8	-51	-44,8	-43,3	-33,5	0	-0,1	-8,1	-35,2	-45	-48,9	-51,1	10	13,2
DRM_B2	DRM_B0	-57	-56,8	-54,8	-43,4	-39,1	-0,7	0	-40,6	-52,2	-53,9	-57	-57	-57	4,5	15,9
DRM_B2	DRM_B1	-56,9	-56,1	-52,7	-40,2	-14,1	-0,1	0	-39,7	-50,8	-52,5	-56,9	-57	-57	5	15,4
DRM_B2	DRM_B2	-55,1	-53,1	-49,5	-40,7	-38,1	-3,7	0	-3,7	-38,1	-40,7	-49,5	-53,1	-55,1	9	15,9
DRM_B2	DRM_B3	-52,9	-51	-47,4	-38,6	-16,6	-3,2	0	-3,2	-16,6	-38,6	-47,4	-51	-52,9	10	15,4
DRM_B3	DRM_B0	-56,4	-56,2	-53,8	-41,1	-14,1	-0,1	0	-37,7	-50,9	-52,8	-56,4	-56,4	-56,4	4,5	15,9
DRM_B3	DRM_B1	-56,8	-55,7	-52,1	-38,2	-8,2	-0,1	0	-37,6	-50,1	-51,9	-56,7	-57	-57	5	15,9
DRM_B3	DRM_B2	-54,3	-52,3	-48,6	-39,3	-16,7	-3,1	0	-3,1	-16,7	-39,3	-48,6	-52,3	-54,3	9	15,9
DRM_B3	DRM_B3	-52,7	-50,7	-47	-37,7	-11,1	-3,1	0	-3,1	-11,1	-37,7	-47	-50,7	-52,7	10	15,9

B_{DRM} : nominal bandwidth of DRM signal.
 DRM_B0: DRM signal, robustness mode B, spectrum occupancy type 0.
 S/I: signal-to-interference ratio for a BER of 1×10^{-4} .

Table E.4: *S/I* correction values in tables E.2 and E.3 to be used for other combinations of modulation scheme and protection level No.

Modulation scheme	Protection level No.	Average code rate	Correction values (dB) for DRM robustness mode/spectrum occupancy type	
			B/0 (4,5 kHz), B/1 (5 kHz)	B/2 (9 kHz), B/3 (10 kHz)
16-QAM	0	0,5	-6,7	-6,6
	1	0,62	-4,7	-4,6
64-QAM	0	0,5	-1,3	-1,2
	1	0,6	0	0
	2	0,71	1,7	1,8
	3	0,78	3,3	3,4

Annex F (informative): Alternative Frequency and announcement signalling

The DRM system can signal alternative frequencies for the whole DRM multiplex or some DRM services of the tuned DRM multiplex to allow the receiver to counter reception problems by automatically and quickly switching to an alternative frequency providing better reception conditions.

In addition, the DRM system can signal announcements (e.g. traffic or news announcements), so that the receiver can automatically switch to another DRM service or even to another broadcast system for the duration of an active announcement.

These two features are signalled using a combination of SDC data entities. This annex describes the general capabilities of the alternative frequency signalling and the announcement feature. It also explains how the different SDC entities work together to enable this kind of signalling.

F.1 Possibilities of the Alternative Frequency Signalling feature

Using the Alternative Frequency feature, the broadcaster can signal to the receiver alternative frequencies for the following items:

- the whole DRM multiplex, being broadcast identically and synchronously on other frequencies; the receiver can check whether it receives the identical DRM multiplex on the indicated frequencies and also their reception quality; if required the receiver can instantly switch to another frequency without service interruption;
- the whole DRM multiplex (all services with the same service identifiers), but with different channel parameters and/or multiplex timing (non-synchronous); frequency switching causes a service interruption;
- single services of the tuned DRM multiplex; frequency switching causes a service interruption; single services can be available:
 - in other DRM multiplexes using the same service identifiers;
 - in other broadcast systems (e.g. AM, FM, FM-RDS, DAB) or other DRM multiplexes using a different service identifier;
- the frequency of the enhancement layer from the base layer, or vice versa.

The validity of alternative frequency lists can be restricted to certain times (schedule definition feature) and/or to certain geographic areas (region definition feature). The schedule definition feature is based on a weekly schedule. The region definition feature allows the definition of geographic areas by longitude/latitude plus extent as well as by internationally standardized CIRAF zones.

If the list of alternative frequencies links to a service carried using another broadcast system, or to a service with a different service identifier, it can be indicated whether the other service carries the identical audio programme or a similar one. The receiver will try to switch to the "same service" before trying to switch to an alternative service.

DRM receivers should store the complete alternative frequency information (all four data entity types 3, 4, 7 and 11) when assigning a DRM service to a station button (see clause G.2). Therefore it is sensible for the broadcaster to signal his complete frequency schedule to the receiver. The frequency schedule should provide the full week's changes rather than only provide a subset, for example only indicating daytime frequencies during daytime transmissions and nighttimes frequencies during night time transmissions. This permits faster start-up of a service if the frequency of the DRM multiplex is different from the last time the service was selected.

It is essential that all frequencies used in a synchronous multi-frequency network are signalled because the bitstream from all transmitters in the network must be identical. This is also highly recommended even if the network is not synchronous because receivers can then store all the possible frequencies when defining a station button including the principle tuned frequency of the multiplex. This frequency information is necessary if the receiver has to tune away and then wants to get back again. It is also required to help a receiver to identify its current region (see clause F.3).

It is recommended that broadcasters specify all alternative sources of each service and include all frequencies. Receivers will sequentially test all given frequencies. If no frequency information is provided, then receivers have to scan for the service identifier if they are to find the service, and that may take so long as to provide an unacceptable user experience.

Broadcasters should note that links to completely analogue broadcast systems have unpredictable results for the receiver, because receivers are unable to check that the correct service is received on the given frequency.

If the broadcaster uses scheduled frequency information he shall provide SDC data entity "Time and date information data entity - type 8" as the time reference for the schedules in "Alternative frequency signalling: Schedule definition data entity - type 4".

F.2 Possibilities of the announcement feature

The announcement feature can be used to interrupt the currently presented audio programme by another providing short clips of information.

Using the announcement feature, the broadcaster can signal to a receiver:

- which types of announcements are provided;
- which type of announcement is currently active;
- whether the announcement content is carried by a DRM service within the current DRM multiplex or by a service on another frequency or from a different broadcast system like FM or DAB.

This information can be specified for each DRM service or for any combination of DRM services within the tuned DRM multiplex.

If the announcement signalling directs the receiver away from the tuned multiplex to another service carrying the announcement content (e.g. on a different type of broadcast system) then the other service shall provide the mechanism to indicate the end of the announcement such that the original listening can be restored. The service identifiers for the other services carrying the announcement content (and optionally their frequencies) are signalled in the SDC by the "Alternative frequency signalling: Other services data entity - type 11" with the "Short Id/Announcement Id flag" set to 1.

EXAMPLE 1: Within the tuned DRM multiplex the broadcaster provides four DRM services named A, B, C and D. A and B provide English programmes while C and D carry German versions of the programmes.

EXAMPLE 2: The broadcaster may use the announcement feature to signal to the receiver that in case of a traffic announcement service A should switch to service B, while service C should switch to service D. Alarm information may only be available in English, so services A, C and D should switch to service B in case of available warning information. The news channel is not broadcast in the tuned DRM multiplex, so service A shall switch to a DAB service (specifying the other service identifier along with an optional list of frequencies) while service C shall switch to an FM-RDS service.

F.3 SDC data entities overview for Alternative Frequency and announcement signalling

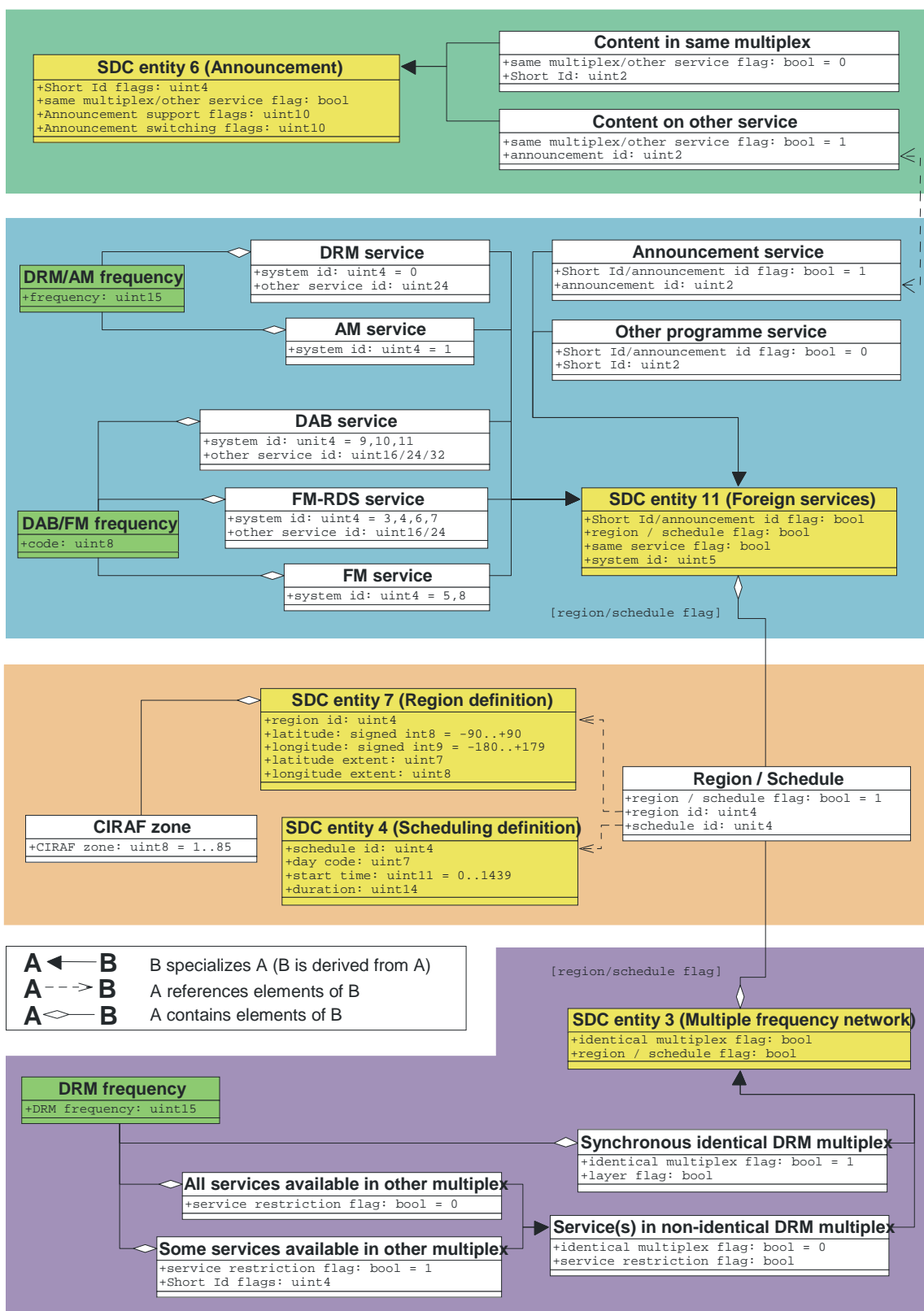


Figure F.1: Alternative Frequency and Announcement Signalling - Involved SDC data entities

F.4 SDC data entities and setup for alternative frequency signalling

The following SDC entities are used to carry the alternative frequency and announcement signalling:

- Alternative frequency signalling: Schedule definition data entity - type 4.
- Alternative frequency signalling: Region definition data entity - type 7.
- Alternative frequency signalling: Multiple frequency network information data entity - type 3.
- Alternative frequency signalling: Other services data entity - type 11.

To set up the alternative frequency signalling feature, the broadcaster will typically perform the following steps:

- 1) If some alternative frequencies are only valid at certain times, provide up to 15 schedules (SDC data entities type 4), each identified by its unique Schedule Id; to explicitly indicate that some frequencies are available all the time, it is recommended that a "24 hours, 7 days a week" schedule is signalled.
- 2) If some alternative frequencies are only valid in certain geographic regions, provide up to 15 region definitions (SDC data entities type 7), each identified by its unique Region Id.
- 3) If alternative frequencies are available for the current DRM multiplex or at least some services, provide SDC data entities type 3 for all lists of frequencies; each list may point to alternative frequencies carrying the identical DRM multiplex in a synchronous way (seamless alternative frequency checking and switching may be performed by the receiver) or carrying some or all services of the current DRM multiplex with different channel parameters and/or not synchronized (checking and switching will interrupt the service presentation); each list of frequencies may be restricted to a geographic area and/or a schedule by referencing one SDC data entity 4 and/or 7.
- 4) If alternative frequencies are available for individual services of the current DRM multiplex using different DRM service identifiers or being carried on a different broadcast system (e.g. DAB or FM-RDS), provide SDC data entities type 11 for all lists of frequencies; each list can indicate one other service identifier, the broadcast system type and the "same service" flag along with a list of frequency values; if only a service identifier is specified (without any frequencies), the receiver has to scan for an available frequency; each list of frequencies may be restricted to a geographic area and/or a schedule by referencing one SDC data entity 4 and/or 7.

A broadcaster can choose to group the frequencies by region by using a common region id per group even if no "Alternative frequency signalling: Region definition data entity - type 7" is provided for a particular Region Id. This allows receivers to check alternative frequencies within the same group first (defined by the Region Id) before checking other groups - the current position of the receiver does not need to be known.

EXAMPLE: Broadcaster X broadcasts a service on frequencies 6 200 kHz and 9 500 kHz in Europe and on frequencies 11 600 kHz and 13 800 kHz in Africa. These four frequencies should be sent as two groups of two using different Region Ids for Europe and Africa, even if no Region definition (data entity type 7) information is given.

F.5 SDC data entities and setup for announcement

The following SDC entities are used to carry the Alternative Frequency and Announcement signalling:

- Alternative frequency signalling: Schedule definition data entity - type 4.
- Alternative frequency signalling: Region definition data entity - type 7.
- Alternative frequency signalling: Other services data entity - type 11.
- Announcement support and switching data entity - type 6.

To set up the announcement signalling, the broadcaster will perform the following steps:

- 1) Set up all required SDC data entities type 6, indicating which types of announcements are provided and which are currently active, and which internal DRM services are linked to which services carrying the announcement content (services in the same DRM multiplex or other services).
- 2) If some alternative frequencies for announcements are only valid at certain times, provide up to 15 schedules (SDC data entities type 4), each identified by its unique Schedule Id; to explicitly indicate that some frequencies are available all the time, it is recommended that a "24 hours, 7 days a week" schedule is signalled.
- 3) If some alternative frequencies for announcements are only valid in certain geographic regions, provide up to 15 region definitions (SDC data entities type 7), each identified by its unique Region Id.
- 4) For every "Announcement Id" value provided by a SDC data entities type 6 (thereby linking to another service), there should be at least one SDC data entity type 11 (with the "Short Id/Announcement Id flag" being set to 1 and using the same "Announcement Id"), providing the broadcast system type, service identifier and frequencies of the other service.

F.6 Alternative frequency and announcement signalling - coding example

Situation

Broadcaster A transmits to the UK and to North America excluding the US.

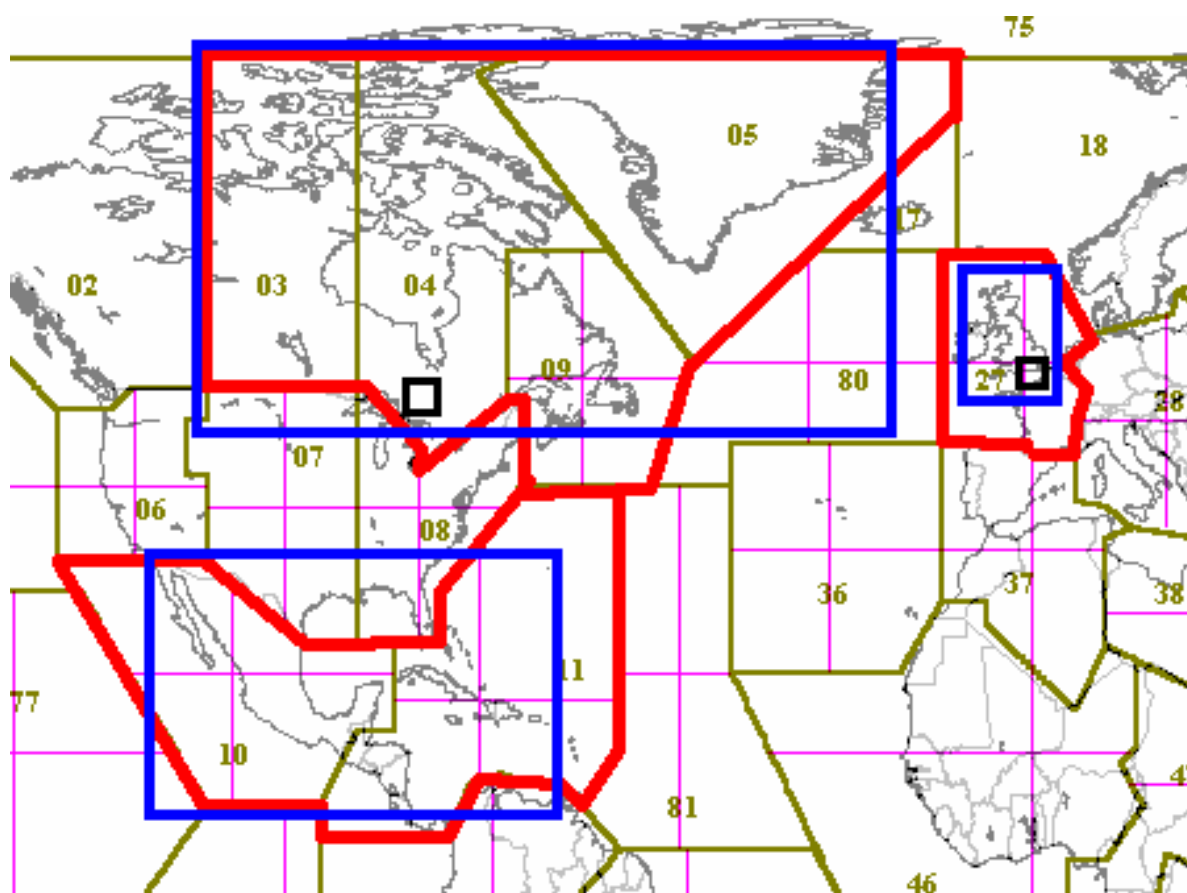


Figure F.2: Coding Example for Alternative Frequency and Announcement Signalling - regions and CIRAF zones

The service is available:

- 1) On SW freq. 1, DRM daily from 16:00 to 02:00 UTC using Service Identifier 1 (all marked regions).
- 2) On SW freq. 2, DRM daily from 12:00 to 18:00 UTC using Service Identifier 1 only Canada, Greenland, UK. This DRM multiplex is synchronous to the first one.
- 3) On SW freq. 3, AM daily from 16:00 to 02:00 UTC (no Service Identifier, all marked region).
- 4) On MW DRM, UK, daily, 24 hours.
- 5) On FM in several parts of UK, using many different frequencies, normally daily, 24 hours, but sometimes there is a different programme on this network (e.g. sports coverage) but a different PI code is then being used.
- 6) On FM in London on 2 different frequencies, daily, 24 hours.
- 7) On FM in a City in North America, weekdays from 18:00 to 22:00 UTC, weekend from 12:00 to 16:00 UTC.
- 8) On DAB, daily, 24 hours, in the UK only.
- 9) In the UK, traffic messages can be obtained from a different service on DAB, daily, 24 hours a day.
- 10) In North America the receiver should tune to the DRM transmission of Broadcaster B when no transmission is available.

Encoded SDC Entities

Schedules (SDC entities type 4)

- Schedule Id 1 | Day Code = 1234567 | Start = 00:00 UTC | Duration = 24 h
alternative coding for the same information:
Schedule Id 1 | Day Code = 1 | Start = 00:00 UTC | Duration = 7×24 h
(means: 24 h, daily).
- Schedule Id 2 | Day Code = 1234567 | Start = 16:00 UTC | Duration = 10 h.
- Schedule Id 3 | Day Code = 1234567 | Start = 12:00 UTC | Duration = 6 h.
- Schedule Id 4 | Day Code = 12345 | Start = 18:00 UTC | Duration = 4 h.
- Schedule Id 4 | Day Code = 67 | Start = 12:00 UTC | Duration = 4 h
(means 18:00 to 22:00 UTC on weekdays and 12:00 to 16:00 UTC on Saturday and Sunday).

Regions (SDC entities type 7)

- Region Id 1 | upper blue rectangle North America | CIRAF 3, 4, 5, 9, 27 (red)
Region Id 1 | lower blue rectangle North America | CIRAF 10, 11 (red)
(means: all marked regions).
- Region Id 2 | upper blue rectangle North America | CIRAF 3, 4, 5, 9, 27 (red).
- Region Id 3 | blue rectangle UK.
- Region Id 4 | black rectangle London.
- Region Id 5 | black rectangle North American City.

DRM Services / Frequencies - same service identifiers (SDC entities type 3)

- Synchronous Multiplex flag = 1 | Layer flag = 0 | Region Id 1 | Schedule Id 2 | SW-Freq. 1
(see number 1 above).
- Synchronous Multiplex flag = 1 | Layer flag = 0 | Region Id 2 | Schedule Id 3 | SW-Freq. 2
(see number 2 above).

Other Services / Frequencies - different service identifiers (SDC entities type 11)

- Same Service flag = 1 | System Id = 00010 (AM without Id) | Region Id 1 | Schedule Id 2 | SW Freq 3 (see number 3 above).
- Same Service flag = 1 | System Id = 00010 (AM without Id) | Region Id 3 | Schedule Id 1 | MW Freq (see number 4 above).
- Same Service flag = 1 | System Id = 00011 (FM-RDS with ECC, Europe and North America) | Region Id 3 | Schedule Id 0 | ECC + PI 1 (see number 5 above).
- Same Service flag = 1 | System Id = 00100 (FM-RDS without ECC, Europe and North America) | Region Id 4 | Schedule Id 1 | PI 2 | FM Freq. 1, FM Freq 2 (see number 6 above).
- Same Service flag = 1 | System Id = 00101 (FM without RDS, Europe and North America) | Region Id 5 | Schedule Id 1 | FM Freq. 3 (see number 7 above).
- Same Service flag = 1 | System Id = 01001 (DAB with ECC + SId) | Region Id 3 | Schedule Id 1 | ECC + SId 1 | DAB Freq 1 (see number 8 above).
- Same Service flag = 0 | System Id = 00000 (DRM) | Region Id 1 | Schedule Id 0 | DRM Service Identifier of Broadcaster B | DRM Freqs 1-n (see number 10 above).

Announcements (SDC entities type 6)

- Short Id 1 | Other Service flag = 1 | Announcement Id = 1 | announcement support flags | announcement switching flags.

Other Service / Frequencies - for Announcement (SDC entities type 11)

- Announcement Id flag = 1 | Announcement Id 1 | System Id = 01001 (DAB with ECC + SId) | Region Id 3 | Schedule Id 1 | ECC + SId 2 | DAB Freq 1 (see number 9 above).

Annex G (informative): Guidelines for receiver implementation

This annex provides some guidelines for receiver behaviour. It does not imply that all types of receivers will include all the described features.

G.1 Alternative Frequency checking and Switching (AFS)

If the receiver notices that the currently selected DRM service has reception problems, it should check for alternative frequencies in the following way:

- 1) First the receiver tries to find an alternative frequency that provides the identical DRM multiplex as the currently tuned frequency. The receiver checks for DRM multiplexes that are identical to the currently tuned multiplex and synchronous to it. This information is available in SDC data entities "Alternative frequency signalling: Multiple frequency network information data entity - type 3"; all suitable DRM multiplexes have the "Synchronous Multiplex flag" set to 1.
For identical and synchronous DRM multiplexes the receiver can check for the availability of the identical DRM multiplex on another frequency (and switch) without service interruption, see clause G.3.
- 2) If no identical and synchronous DRM multiplex is available the receiver checks other DRM frequencies that carry at least the currently selected DRM service(s). On a non-identical DRM multiplex the channel parameter and the service structure can be different (e.g. more or less services; with or without multimedia data) and the currently selected service might be carried with different parameters (e.g. different bitrate or audio mode). The SDC data entities "Alternative frequency signalling: Multiple frequency network information data entity - type 3" ("Synchronous Multiplex flag" has the value 0) lists all alternative DRM multiplexes that carry one or more DRM services of the current DRM multiplex.

If the receiver can present multiple services at the same time (i.e. audio and multimedia) the receiver should first try the alternative frequencies that carry all currently selected services.

Attention: checking for the availability of a DRM service on another frequency causes a service interruption (regardless whether the DRM service is available on another frequency or not).

- 3) If "Alternative frequency signalling: Other services - data entity type 11" (marked as "same service") are available that signal at least one alternative frequency, the receiver shall check these frequencies. If none of the given frequencies is valid then the receiver checks all given frequencies for other services NOT marked as "same service".
- 4) If "Alternative frequency signalling: Other services - data entity type 11" signals a DRM service identifier or another service identifier but no frequency, then the receiver has to scan for the selected DRM service identifier or for the other service identifier. It starts scanning for "same services" before trying NOT "same services".

Scanning will usually not be sensible if the selected DRM service signals that the alternative source is currently not broadcast (described by scheduled frequency lists) or that it is not receivable in the receiver's region (indicated by region definitions).

If no alternative frequencies are signalled at all or no valid alternative frequency could be found then the receiver could scan for the current DRM service and any known alternative other services. Such a scanning can take a very long time!.

NOTE 1: In step 4 the broadcaster explicitly asks the receiver to scan for a given other service identifier (by not providing any frequency). In step 5 the receiver assumes that the alternative frequency information provided by the broadcaster is not complete.

- 5) If also step 5 fails then there is no alternative source for the current DRM service available.

When the receiver checks for alternative frequencies (steps 1 to 4) it will also take into account the SDC data entities "Alternative frequency signalling: Region definition data entity - type 7" and "Alternative frequency signalling: Schedule definition data entity - type 4" to determine when and where a certain alternative frequency is valid. Note that SDC data entity "Time and date information data entity - type 8" provides the time reference for the schedules in "Alternative frequency signalling: Schedule definition data entity - type 4".

If a receiver does not know its current position or if the "Alternative frequency signalling: Region definition data entity - type 7" describing the region has not yet been received (or is not broadcast), it might still evaluate the Region Id. The receiver can determine the Region Id of the currently tuned frequency and thus first check the alternative frequencies belonging to the same region.

NOTE 2: These proposed steps do not tell when a receiver should switch to an alternative frequency nor do they forbid that the receiver selects DAB services before DRM ones. But the receiver shall never try NOT-"same services" unless all other sources of the selected DRM service have failed. NOT-"same services" are the last resort!.

NOTE 3: Probably a receiver will try seamless AFS ("Seamless Alternative Frequency Checking and Switching"; see clause G.3) to identical and synchronous DRM multiplexes even while the currently tuned frequency has no reception problems. In case the reception of the currently tuned frequency fails the receiver can then find an alternative frequency faster.

G.2 Station buttons for DRM services

If the user assigns a DRM service to a station button it is recommended for the DRM receiver to store all alternative frequency information (all four data entity types 3, 4, 7 and 11) as well as the currently used frequency and the DRM service identifier. This permits the receiver to find the right frequency even if the service is selected (by the user pressing the station button) while the service is broadcast on another frequency compared to the time the station button was defined.

If such a station button is selected the expected behaviour of the receiver would be as follows:

- a) Check on what frequency the service was received the last time. If this frequency was marked "same service" in the alternative frequency information the receiver should try tuning to this frequency. If tuning succeeds (the desired DRM service identifier is available on the frequency) tuning is successfully finished. If the receiver did not find the expected service identifier it has to start with step 1 of the above list (see clause G.1).
- b) If the frequency we successfully tuned to the last time is NOT marked as "same service" then the receiver directly starts with step 1 of the above list (see clause G.1).
- c) If the desired DRM service cannot be found the receiver might then offer the user to scan for this DRM service, i.e. try to find the DRM service identifier on another frequency. Please note that scanning might be sensible if the receiver did not switch to a certain DRM service for a long time (i.e. if the frequencies might have changed in the mean time). Scanning will usually not be sensible if the user tries to tune to a DRM service that is broadcast just part of the day (described by scheduled frequency lists) and that provides no frequencies for the current day of week/time.

The "same service" flag permits faster start of service presentation and by setting this flag to "0" the broadcaster can order the receiver to first try "same services" before NOT-"same services".

Please note that for scheduled frequencies it is useful to have a battery powered real time clock that provides at least day of week and time (UTC) when the receiver is switched on.

G.3 Seamless Alternative Frequency checking and Switching (AFS)

Seamless Alternative Frequency Switching (AFS) provides the functionality of seamlessly checking for the availability of the same programme material on a differing frequency and then switching to it if it is valid. Alternative frequencies can be signalled by use of SDC data entity type 3. The various steps of this process are indicated in figure G.1.

AFS specific mathematical symbols are defined as follows:

- T_d : time delay at point of reception between the current and the possible alternative frequency.
- T_{tune} : time needed by the receiver to tune to the alternative frequency.
- T_{check} : time available to acquire the data required for the validation of the AF.

Procedure:

At the start of known SDC block on the tuned frequency, the receiver re-tunes to the alternative frequency. It acquires the data necessary to perform the AF-check and immediately tunes back to the original tuned frequency. This process has to be completed within the time interval T_{check} . Subsequently the validity of the alternative frequency can be computed before the next occurrence of the SDC. Subject to the validation of the alternative frequency, the receiver may choose to switch to the new frequency at this point without an interruption of service.

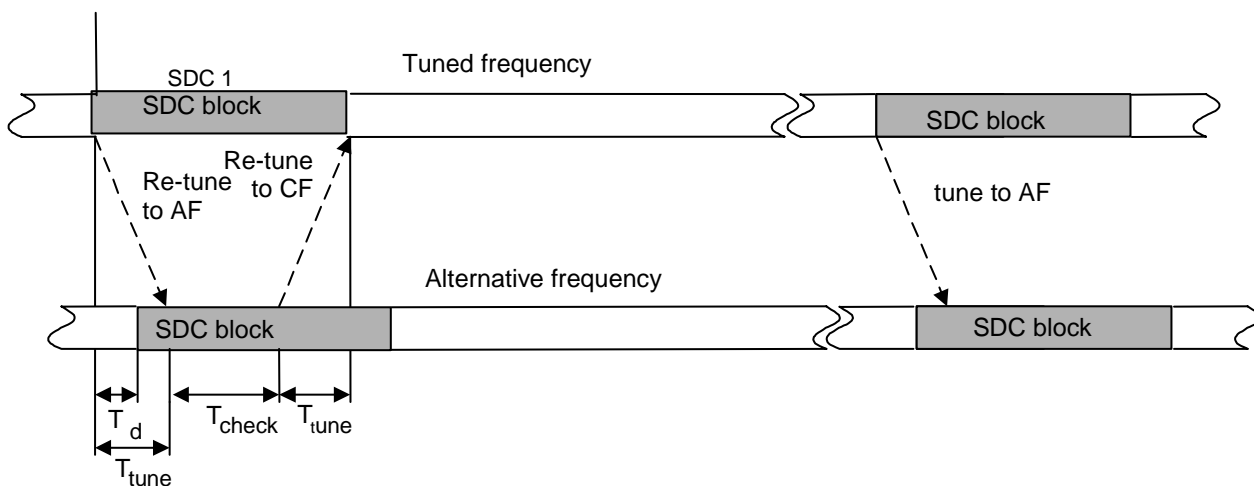


Figure G.1: Illustration of AFS function

The points at which the receiver may check the alternative frequencies are governed by the Identity field in the FAC in combination with the AFS index signalled in the SDC.

If the receiver detects a failure of the FAC CRC for the first transmission frame of the transmission super frame then it cannot perform an AFS check because the value of the Identity field is unknown.

For fully dynamic operation (see clause 6.4.5), no AFS is possible because the receiver has no knowledge of the data that will be sent in future SDC blocks.

For fully static operation, the AFS function may be performed every super transmission frame, provided that the receiver has stored all the different SDC blocks in the cycle. The number of SDC blocks in the cycle is given by the AFS index + 1.

For semi-dynamic operation, the AFS function may only be performed at certain transmission super frames. The following examples illustrate some of the many possibilities.

EXAMPLE 1: Changing the content of the SDC block (A to B) with AFS index = 0.

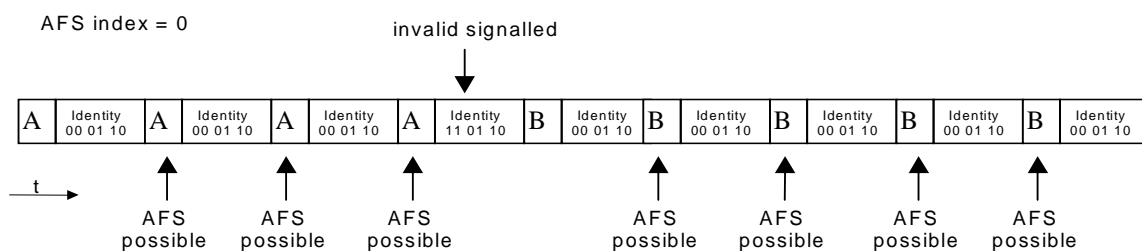


Figure G.2: Example 1

NOTE 1: Very fast AFS possibility after tuning;
 Very limited SDC data size when AFS feature should be used.

EXAMPLE 2: Changing the content of both SDC blocks (A to C; B to D) with AFS index = 1.

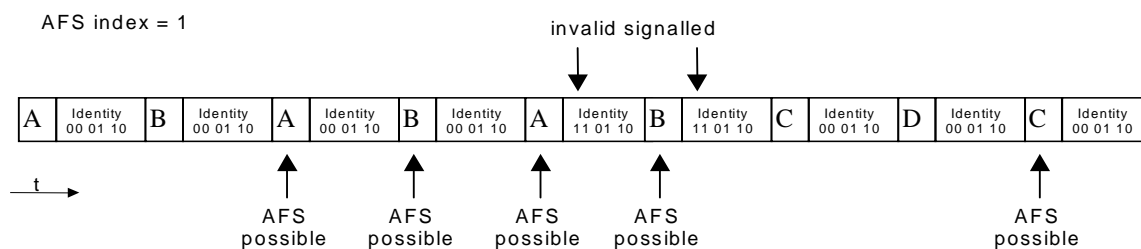


Figure G.3: Example 2

NOTE 2: While changing the SDC blocks in two consecutive SDC blocks is no AFS possible;
 With AFS = n the first AFS can occur after (n + 1) received SDC blocks.

EXAMPLE 3: Changing the content of one SDC block (A to C) with AFS index = 1.

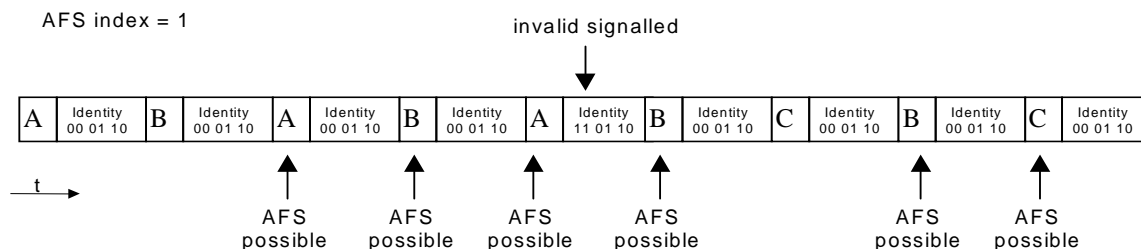


Figure G.4: Example 3

NOTE 3: Only one AFS possibility is missed.

EXAMPLE 4: Continuous changing of one SDC block (B to C to D...) with AFS index = 1.

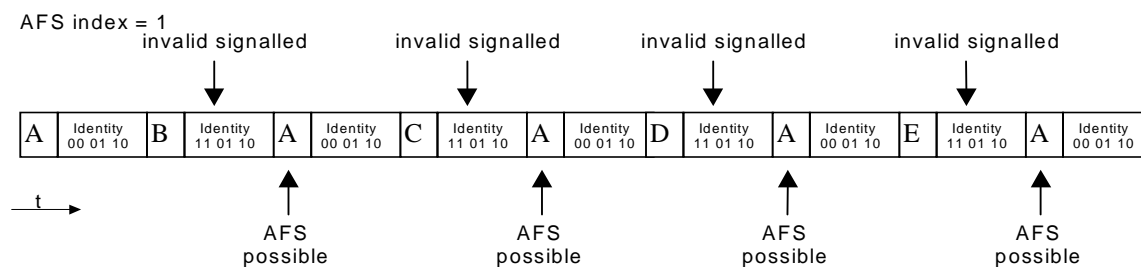


Figure G.5: Example 4

NOTE 4: Only every second frame AFS is possible;
 SDC data size is increased.

EXAMPLE 5: Change of repetition rate of SDC block (without reconfiguration) with AFS index = 1.

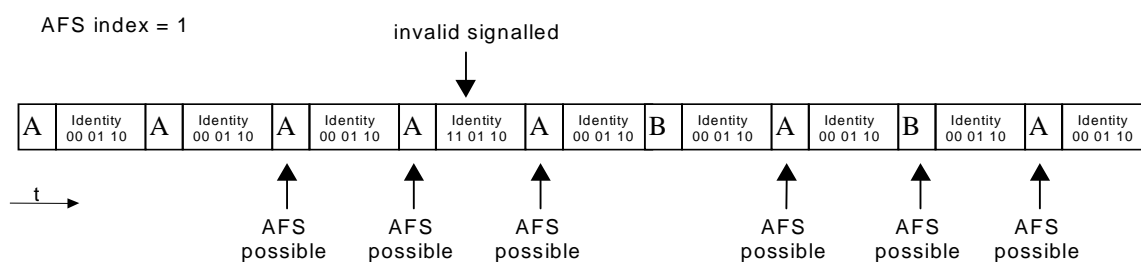


Figure G.6: Example 5

NOTE 5: First AFS possible after n frames after tuning;
Flexible SDC data size.

For other values of AFS index, similar schemes can be applied.

G.4 Character sets

DRM uses UTF-8 character coding, and so allows all known characters to be broadcast. However, receivers may be produced that do not support all characters. In this case the following behaviour is recommended:

- For the label (sent using SDC data entity type 1): if the receiver character set does not support most of the characters being broadcast (for example, the label is composed of Hindi characters, but the receiver only supports European characters) then the receiver could display the Service Identifier and/or frequency instead.
- For the text message (carried in the MSC): if the receiver character set does not support all of the characters being broadcast then the receiver should attempt to determine if any portion of the message can be displayed. This would be, for example, if at least 10 consecutive characters could be displayed.

Annex H (informative): Service capacity and bit rates

The following table gives the orders of magnitude of the available total bit rates, which depend upon the signal bandwidth, the protection mode, and the error correction code rates.

For a 64-QAM modulation, a coding rate of 0,6, for the MSC (EEP SM):

Robustness mode	Spectrum occupancy					
	0	1	2	3	4	5
A	11,3 kbit/s	12,8 kbit/s	23,6 kbit/s	26,6 kbit/s	49,1 kbit/s	55 kbit/s
B	8,7 kbit/s	10 kbit/s	18,4 kbit/s	21 kbit/s	38,2 kbit/s	43 kbit/s
C	-	-	-	16,6 kbit/s	-	34,8 kbit/s
D	-	-	-	11 kbit/s	-	23,4 kbit/s

For a 16-QAM modulation, a coding rate of 0,62, for the MSC (EEP SM):

Robustness mode	Spectrum occupancy					
	0	1	2	3	4	5
A	7,8 kbit/s	8,9 kbit/s	16,4 kbit/s	18,5 kbit/s	34,1 kbit/s	38,2 kbit/s
B	6 kbit/s	6,9 kbit/s	12,8 kbit/s	14,6 kbit/s	26,5 kbit/s	29,8 kbit/s
C	-	-	-	11,5 kbit/s	-	24,1 kbit/s
D	-	-	-	7,6 kbit/s	-	16,3 kbit/s

Minimum absolute (R = 0,50, 16-QAM, Mode B, 4,5 kHz) 4,8 kbit/s.

Maximum absolute (R = 0,78, 64-QAM, Mode A, 20 kHz) 72 kbit/s.

Annex I (normative): SBR tables

I.1 Predefined tunings for CELP + SBR

This clause defines all SBR parameters needed to decode *sbr_speech_data()* in the case that the SDC field *sbr_header_flag* = 0, i.e. *sbr_speech_header()* is absent in the bitstream. The SBR tuning is derived from the SDC signals *audio sampling rate* (8 kHz or 16 kHz), *CELP_CRC* and the bitrate for the SBR part of the combined CELP + SBR bitstream.

An SBR preset bitrate, *BR*, is calculated as:

$$BR = BR_{tot} - BR_{celp} - BR_{sbr_crc}$$

where:

BR_{tot} = total bitrate for the CELP + SBR service as derived from the SDC multiplex description data entity (type 0).

BR_{celp} = bitrate for the CELP, including CRC bits when enabled, as given by *CELP_index* (at the particular *audio sampling rate*) and *CELP_CRC*.

BR_{sbr_crc} = bitrate for the CRC of the SBR bitstream as given by *CELP_CRC* (i.e. 0 or 200).

The SBR data matrix composition for a given *BR* is defined in table I.1 for *audio sampling rate* = 8 kHz and table I.2 for *audio sampling rate* = 16 kHz.

In addition to the values given in tables I.1 and I.2, default values according to table 37 in clause 5.6.2.2.1 are used. Finally, *k0* = 15 and *k2* = 48 is used for 8 kHz CELP (3,75 kHz resp. 12 kHz) and *k0* = 30 and *k2* = 58 is used for 16 kHz CELP (7,5 kHz resp. 14,5 kHz).

Table I.1: SBR presets for 8 kHz CELP

Input BR (bit/s)	SBR Preset Parameters							SBR Payload Size (bit/frame)
	bs_ num_ level	bs_ num_ shape	num_ shape_ bands	bs_ num_ noise	num_ noise_ bands	bs_ num_ invf	bs_ num_ rfield	
700	2	1	5	1	1	0	0	28
780	2	1	5	1	2	0	0	31
1 200	2	2	5	2	1	1	0	48
1 260	2	2	5	2	1	1	1	50
1 300	2	2	5	2	2	0	0	52
1 360	2	2	5	2	2	1	0	54
1 400	2	2	6	2	1	1	1	56
1 460	2	2	6	2	2	0	0	58
1 500	2	2	7	2	1	1	0	60
1 600	2	2	7	2	2	0	0	64
1 700	4	2	7	2	1	0	0	68
1 860	4	2	7	2	2	0	0	74

NOTE: The optional CRC bits are not included in Size.

Table I.2: SBR presets for 16 kHz CELP

Input BR (bit/s)	SBR Preset Parameters							SBR Payload Size (bit/frame)
	bs_ num_ level	bs_ num_ shape	num_ shape_ bands	bs_ num_ noise	num_ noise_ bands	bs_ num_ invf	bs_ num_ rfield	
900	2	2	3	2	1	1	0	36
1 000	2	2	3	2	2	0	0	40
1 060	2	2	4	2	1	1	0	42
1 160	2	2	4	2	2	0	0	46
1 200	4	2	3	2	1	1	1	48
1 260	4	2	3	2	2	0	0	50
1 300	4	2	4	2	1	1	0	52
1 400	4	2	4	2	2	0	0	56
1 760	4	4	3	4	1	1	0	70
2 000	4	4	3	4	2	0	0	80
2 060	4	4	4	4	1	1	0	82
2 300	4	4	4	4	2	0	0	92

NOTE: The optional CRC bits are not included in Size.

1.2 Predefined tunings for HVXC + SBR

This clause defines all SBR parameters needed to decode *sbr_speech_data()* in the case that the SDC field *sbr_header_flag* = 0, i.e. *sbr_speech_header()* is absent in the bitstream. The SBR tuning is derived from the SDC signals *HVXC_CRC*, *sbr_hvxc_preset* and the nominal bitrate for the SBR part of the combined HVXC + SBR bitstream.

The nominal SBR bitrate, *BR_n*, is calculated as:

$$BR_n = BR_{tot} - BR_{hvxc}$$

where:

BR_{tot} = total bitrate for the HVXC + SBR service as derived from the SDC multiplex description data entity (type 0).

BR_{hvxc} = bitrate for the HVXC, including padding bits and CRC bits when enabled, as given by *HVXC_rate* and *HVXC_CRC*.

The effective SBR bitrate, *BR_e*, is equal to or less than *BR_n* and is given for reference in the tables below. The calculated value of *BR_n* is compared to the intervals or single values given in table I.3 for *HVXC_CRC* = 0 and table I.4 for *HVXC_CRC* = 1. For a given match such that $BR_n \in iBR$, the SBR data matrix composition is then further specified by *xmit_noise* and *freq_prio* as defined below.

The *sbr_hvxc_preset* field is subdivided according to:

- *xmit_noise* 1 bit.
- *freq_prio* 1 bit.
- *cutoff_freq* 2 bits.

The following definitions apply:

xmit_noise: this field indicates whether noise related parameters are transmitted or not:

0: noise floor and inverse filtering parameters absent.

1: noise floor and inverse filtering parameters present.

freq_prio: this field indicates tuning emphasis on time or frequency resolution:

0: low frequency resolution and high time resolution.

1: high frequency resolution and low time resolution.

cutoff_freq: this field indicates the upper corner frequency of the SBR range:

00: $k_2 = 32$ (8 kHz).

01: $k_2 = 40$ (10 kHz).

10: $k_2 = 48$ (12 kHz).

11: $k_2 = 56$ (14 kHz).

Thus, four compositions of the SBR data matrix are available at every nominal bitrate range. The frequency range to which the SBR bitstream data is mapped is signalled by *cutoff_freq*, offering four different applications of every data matrix composition, offering a total of 16 tunings per SBR bitrate range.

In addition to the values given in tables I.3 and I.4, default values according to table 38 in clause 5.6.2.2.1 are used. Finally, $k_0 = 15$ is used for all tunings below.

Table I.3: SBR bitstream data matrix for HVXC_CRC = 0

Input Parameters			SBR Preset Parameters							SBR Payload	
iBR (bit/s)	xmit_n oise	freq_ prio	bs_ num_ level	bs_ num_ shape	num_ shape_ bands	bs_ num_ noise	num_ noise_ bands	bs_ num_ invf	bs_ num_ rfield	Size (bit/ frame)	BRe (bit/s)
[780 to 800]	0	0	2	2	3	0	1	0	2	32	800
	0	1	2	1	6	0	1	0	2	32	800
	1	0	2	1	5	2	1	0	0	31	775
	1	1	2	1	5	1	2	0	0	31	775
1 200	0	0	4	2	4	0	1	0	2	48	1 200
	0	1	2	2	6	0	1	0	1	48	1 200
	1	0	2	2	5	2	1	1	0	48	1 200
	1	1	2	2	5	1	2	1	0	48	1 200
1 400	0	0	4	2	6	0	1	0	0	56	1 400
	0	1	2	2	7	0	1	0	2	56	1 400
	1	0	2	2	6	2	1	1	1	56	1 400
	1	1	2	2	6	1	2	1	1	56	1 400
1 600	0	0	4	2	7	0	1	0	1	64	1 600
	0	1	2	2	9	0	1	0	0	64	1 600
	1	0	2	2	7	2	1	1	2	64	1 600
	1	1	2	2	7	2	2	0	0	64	1 600
1 760	0	0	4	4	4	0	2	0	1	70	1 750
	0	1	4	2	8	0	2	0	1	70	1 750
	1	0	4	2	6	2	2	1	0	70	1 750
	1	1	2	2	7	2	2	1	2	70	1 750

Table I.4: SBR bitstream data matrix for HVXC_CRC = 1

Input Parameters			SBR Preset Parameters							SBR Payload	
iBR (bit/s)	xmit_noise	freq_prio	bs_num_level	bs_num_shape	num_shape_bands	bs_num_noise	num_noise_bands	bs_num_invf	bs_num_rfield	Size (bit/frme)	BRe (bit/s)
[760 to 780]	0	0	2	1	4	0	1	0	0	30	750
	0	1	1	1	5	0	1	0	1	30	750
	1	0	2	1	3	1	1	0	0	30	750
	1	1	1	1	4	1	2	0	0	31	775
[1 060 to 1 080]	0	0	2	2	4	0	1	0	0	42	1 050
	0	1	2	1	7	0	1	0	2	43	1 075
	1	0	2	2	3	2	1	0	0	42	1 050
	1	1	2	1	5	1	3	0	0	42	1 050
[1 160, 1 200]	0	0	2	2	4	0	1	0	2	46	1 150
	0	1	2	2	5	0	1	0	0	48	1 200
	1	0	2	2	4	2	1	0	0	48	1 200
	1	1	2	2	4	1	2	0	0	48	1 200
1 260	0	0	4	2	3	0	1	0	2	50	1 250
	0	1	2	2	5	0	1	0	1	50	1 250
	1	0	2	2	4	2	1	1	0	50	1 250
	1	1	2	2	4	1	2	1	0	50	1 250
1 400	0	0	4	2	4	0	1	0	2	56	1 400
	0	1	2	2	6	0	1	0	1	56	1 400
	1	0	2	2	5	2	1	1	0	56	1 400
	1	1	2	2	5	1	2	1	0	56	1 400
1 600	0	0	4	2	6	0	1	0	0	64	1 600
	0	1	2	2	7	0	1	0	2	64	1 600
	1	0	2	2	6	2	1	1	1	64	1 600
	1	1	2	2	6	1	2	1	1	64	1 600
1 800	0	0	4	2	7	0	2	0	1	72	1 800
	0	1	4	2	7	0	2	0	1	72	1 800
	1	0	2	2	6	2	2	1	2	72	1 800
	1	1	2	2	7	2	2	0	0	72	1 800
1 860	0	0	4	2	7	0	2	0	2	74	1 850
	0	1	2	2	9	0	2	0	1	74	1 850
	1	0	2	2	7	2	2	0	1	74	1 850
	1	1	2	2	7	2	2	1	0	74	1 850
1 900	0	0	4	4	4	0	2	0	0	76	1 900
	0	1	4	2	8	0	2	0	0	76	1 900
	1	0	4	2	6	2	2	0	0	76	1 900
	1	1	2	2	7	2	2	1	1	76	1 900

NOTE: The CRC bits are included in Size and Bre.

1.3 Parametric stereo Huffman tables

The function *huff_dec()* is used as:

$$data = huff_dec(t_huff, codeword),$$

where *t_huff* is the selected Huffman table and *codeword* is the word read from the bitstream. The returned value *data* is the index in the Huffman table with an offset of the corresponding Largest Absolute Value (LAV) of the table.

Table I.5: PS Huffman tables overview

table name	bs_sa_dtdf_flag	bs_pan_dtdf_flag	LAV	Notes
f_huffman_sa	0	dc	7	
t_huffman_sa	1	dc	7	
f_huffman_sa	0	dc	7	
t_huffman_sa	1	dc	7	
f_huffman_pan	dc	0	14	
t_huffman_pan	dc	1	14	
f_huffman_pan	dc	0	14	
t_huffman_pan	dc	1	14	

Table I.6: f_huffman_sa

index	Length (hexadecimal)	codeword (hexadecimal)	index	length (hexadecimal)	codeword (hexadecimal)
0	0x00000007	0x0000007E	8	0x00000004	0x0000000A
1	0x00000007	0x0000007F	9	0x00000005	0x0000001A
2	0x00000006	0x0000003D	10	0x00000005	0x0000001C
3	0x00000006	0x0000003C	11	0x00000006	0x0000003A
4	0x00000005	0x0000001B	12	0x00000006	0x0000003B
5	0x00000004	0x0000000C	13	0x00000006	0x0000003E
6	0x00000004	0x0000000B	14	0x00000003	0x00000004
7	0x00000001	0x00000000			

Table I.7: t_huffman_sa

index	Length (hexadecimal)	codeword (hexadecimal)	index	length (hexadecimal)	codeword (hexadecimal)
0	0x00000008	0x000000FE	8	0x00000003	0x00000005
1	0x00000007	0x0000007E	9	0x00000004	0x0000000D
2	0x00000007	0x0000007C	10	0x00000005	0x0000001D
3	0x00000006	0x0000003C	11	0x00000006	0x0000003D
4	0x00000005	0x0000001C	12	0x00000007	0x0000007D
5	0x00000004	0x0000000C	13	0x00000009	0x000001FE
6	0x00000003	0x00000004	14	0x00000009	0x000001FF
7	0x00000001	0x00000000			

Table I.8: f_huffman_pan

index	Length (hexadecimal)	Codeword (hexadecimal)	index	length (hexadecimal)	codeword (hexadecimal)
0	0x00000010	0x0000FFFC	15	0x00000003	0x00000006
1	0x0000000F	0x00007FFA	16	0x00000005	0x0000001D
2	0x00000010	0x0000FFFD	17	0x00000006	0x0000003D
3	0x0000000F	0x00007FFB	18	0x00000007	0x0000007D
4	0x0000000D	0x00001FFC	19	0x00000008	0x000000FD
5	0x0000000E	0x00003FFA	20	0x00000009	0x000001FD
6	0x0000000C	0x00000FFC	21	0x0000000A	0x000003FE
7	0x00000009	0x000001FE	22	0x0000000C	0x00000FFD
8	0x00000009	0x000001FC	23	0x0000000E	0x00003FFB
9	0x00000008	0x000000FC	24	0x0000000E	0x00003FFC
10	0x00000007	0x0000007C	25	0x0000000F	0x00007FFC
11	0x00000006	0x0000003C	26	0x00000010	0x0000FFFE
12	0x00000005	0x0000001C	27	0x0000000F	0x00007FFD
13	0x00000002	0x00000002	28	0x00000010	0x0000FFFF
14	0x00000001	0x00000000			

Table I.9: t_huffman_pan

index	Length (hexadecimal)	codeword (hexadecimal)	index	length (hexadecimal)	codeword (hexadecimal)
0	0x00000012	0x0003FFF8	15	0x00000003	0x00000006
1	0x00000012	0x0003FFF9	16	0x00000005	0x0000001E
2	0x00000012	0x0003FFFA	17	0x00000007	0x0000007E
3	0x00000012	0x0003FFFB	18	0x00000009	0x000001FE
4	0x0000000F	0x00007FFC	19	0x0000000B	0x000007FD
5	0x00000010	0x0000FFFC	20	0x0000000C	0x00000FFD
6	0x0000000E	0x00003FFC	21	0x0000000D	0x00001FFD
7	0x0000000D	0x00001FFC	22	0x0000000E	0x00003FFD
8	0x0000000C	0x00000FFC	23	0x00000010	0x0000FFFD
9	0x0000000B	0x000007FC	24	0x0000000F	0x00007FFD
10	0x00000008	0x000000FE	25	0x00000012	0x0003FFFC
11	0x00000006	0x0000003E	26	0x00000012	0x0003FFFD
12	0x00000004	0x0000000E	27	0x00000012	0x0003FFFE
13	0x00000002	0x00000002	28	0x00000012	0x0003FFFF
14	0x00000001	0x00000000			

I.4 Miscellaneous parametric stereo tables

Table I.10: SA quantization class vector $\mathbf{c}_{SaQClass}(k)$

k	$\mathbf{c}_{SaQClass}(k)$	k	$\mathbf{c}_{SaQClass}(k)$
0	0	4	2
1	1	5	2
2	2	6	2
3	2	7	2

Table I.11: Pan quantization class vector $\mathbf{c}_{PanQClass}(k)$

k	$\mathbf{c}_{PanQClass}(k)$	k	$\mathbf{c}_{PanQClass}(k)$
0	0	10	2
1	1	11	2
2	1	12	2
3	1	13	2
4	1	14	3
5	1	15	3
6	1	16	3
7	1	17	4
8	1	18	4
9	1	19	4

Table I.12: SA quantization matrix $C_{SaQMatrix}(c, k)$

k	$C_{SaQMatrix}(0, k)$	$C_{SaQMatrix}(1, k)$	$C_{SaQMatrix}(2, k)$
0	0	0	0
1	0,0501	0,1778	0,1778
2	0,0706	0,2818	0,2818
3	0,0995	0,4467	0,4467
4	0,1399	0,5623	0,5623
5	0,1957	0,7079	0,7079
6	0,2713	0,8913	0,8913
7	0,3699	1	1

Table I.13: Pan quantization matrix $C_{PanQMatrix}(c, k)$

k	$C_{PanQMatrix}(0, k)$	$C_{PanQMatrix}(1, k)$	$C_{PanQMatrix}(2, k)$	$C_{PanQMatrix}(3, k)$	$C_{PanQMatrix}(4, k)$
0	0	0	0	0	0
1	0,1661	0,1661	0,3322	0,3322	0,3322
2	0,3322	0,3322	0,6644	0,8305	0,8305
3	0,4983	0,6644	0,9966	1,4949	1,6610
4	0,6644	0,9966	1,4949	2,1593	2,4914
5	0,8305	1,3288	2,1593	2,9897	3,4880
6	0,9966	1,8271	2,8236	3,8202	4,6507
7	1,3288	2,3253	3,4880	4,6507	5,8134

Table I.14: SA frequency scale vectors $f_{SaTable24kHz}(k)$ and $f_{SaTable48kHz}(k)$

k	$f_{SaTable24kHz}(k)$	$f_{SaTable48kHz}(k)$
0	0	0
1	1	1
2	2	2
3	3	3
4	5	5
5	7	7
6	10	10
7	13	13
8	46	23

Table I.15: Pan frequency scale vector $f_{PanTable}(k)$

k	$f_{PanTable}(k)$	k	$f_{PanTable}(k)$
0	0	11	11
1	1	12	12
2	2	13	13
3	3	14	14
4	4	15	15
5	5	16	18
6	6	17	22
7	7	18	26
8	8	19	32
9	9	20	64
10	10		

Table I.16: Filter coefficient vector $\mathbf{a}(k)$

k	$\mathbf{a}(k)$
0	0,65143905753106
1	0,56471812200776
2	0,48954165955695

Table I.17: Delay length vectors $\mathbf{d}_{24kHz}(k)$ and $\mathbf{d}_{48kHz}(k)$

k	$\mathbf{d}_{24kHz}(k)$	$\mathbf{d}_{48kHz}(k)$
0	1	3
1	2	4
2	3	5

Table I.18: Fractional delay length vector $\mathbf{q}(k)$

k	$\mathbf{q}(k)$
0	0,43
1	0,75
2	0,347

Table I.19: Peak Decay Factors $\alpha_{Decay24kHz}$ and $\alpha_{Decay48kHz}$

$\alpha_{Decay24kHz}$	$\alpha_{Decay48kHz}$
0,58664621951003	0,76592833836465

Annex J (informative): Numbers of input bits

Table J.1: Number of input bits L per multiplex frame for EEP SM Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,5$	3 757	4 248	7 878	8 857	16 374	18 336
64 QAM, $R_{all} = 0,6$	4 509	5 096	9 450	10 628	19 646	21 998
64 QAM, $R_{all} = 0,71$	5 322	6 018	11 157	12 547	23 193	25 976
64 QAM, $R_{all} = 0,78$	5 898	6 664	12 364	13 908	25 704	28 788
16 QAM, $R_{all} = 0,5$	2 505	2 832	5 250	5 904	10 914	12 222
16 QAM, $R_{all} = 0,62$	3 131	3 540	6 565	7 381	13 645	15 280

Table J.2: Number of input bits L per multiplex frame for EEP SM Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,5$	2 880	3 312	6 133	6 991	12 727	14 304
64 QAM, $R_{all} = 0,6$	3 456	3 972	7 361	8 390	15 272	17 162
64 QAM, $R_{all} = 0,71$	4 080	4 692	8 688	9 900	18 026	20 264
64 QAM, $R_{all} = 0,78$	4 520	5 196	9 630	10 980	19 980	22 456
16 QAM, $R_{all} = 0,5$	1 920	2 208	4 089	4 662	8 484	9 534
16 QAM, $R_{all} = 0,62$	2 400	2 760	5 111	5 826	10 606	11 920

Table J.3: Number of input bits L per multiplex frame for EEP SM Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,5$	Not used			5 514	Not used	11 581
64 QAM, $R_{all} = 0,6$				6 615		13 898
64 QAM, $R_{all} = 0,71$				7 808		16 406
64 QAM, $R_{all} = 0,78$				8 654		18 188
16 QAM, $R_{all} = 0,5$				3 675		7 722
16 QAM, $R_{all} = 0,62$				4 595		9 651

Table J.4: Number of input bits L per multiplex frame for EEP SM Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,5$	Not used			3 660	Not used	7 800
64 QAM, $R_{all} = 0,6$				4 391		9 359
64 QAM, $R_{all} = 0,71$				5 185		11 050
64 QAM, $R_{all} = 0,78$				5 746		12 242
16 QAM, $R_{all} = 0,5$				2 439		5 199
16 QAM, $R_{all} = 0,62$				3 050		6 500

Table J.5: Number of input bits L per hierarchical frame for EEP HMsym VSPP Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		2 626	2 953	5 458	6 112
64 QAM, $R_0 = 0,57$			3 000	3 372	6 236	6 984
64 QAM, $R_0 = 0,6$			3 150	3 543	6 549	7 332
64 QAM, $R_0 = 0,66$			3 500	3 936	7 276	8 148

Table J.6: Number of input bits L per hierarchical frame for EEP HMsym VSPP Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		2 045	2 331	4 243	4 768
64 QAM, $R_0 = 0,57$			2 336	2 664	4 848	5 448
64 QAM, $R_0 = 0,6$			2 454	2 796	5 091	5 721
64 QAM, $R_0 = 0,66$			2 726	3 108	5 656	6 356

Table J.7: Number of input bits L per hierarchical frame for EEP HMsym VSPP Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used			1 838	Not used	3 861
64 QAM, $R_0 = 0,57$			2 100	4 412		
64 QAM, $R_0 = 0,6$			2 205	4 632		
64 QAM, $R_0 = 0,66$			2 450	5 148		

Table J.8: Number of input bits L per hierarchical frame for EEP HMsym VSPP Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used			1 220	Not used	2 600
64 QAM, $R_0 = 0,57$			1 392	2 968		
64 QAM, $R_0 = 0,6$			1 464	3 120		
64 QAM, $R_0 = 0,66$			1 626	3 466		

Table J.9: Number of input bits L per multiplex frame for EEP HMsym SPP Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,45$	Not used		4 725	5 313	9 822	10 998
64 QAM, $R_{all} = 0,55$			5 724	6 432	11 904	13 332
64 QAM, $R_{all} = 0,72$			7 592	8 538	15 784	17 680
64 QAM, $R_{all} = 0,78$			8 164	9 184	16 972	19 012

Table J.10: Number of input bits L per multiplex frame for EEP HMsym SPP Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,45$	Not used		3 681	4 194	7 635	8 580
64 QAM, $R_{all} = 0,55$			4 452	5 076	9 252	10 392
64 QAM, $R_{all} = 0,72$			5 913	6 738	12 268	13 792
64 QAM, $R_{all} = 0,78$			6 358	7 252	13 192	14 828

Table J.11: Number of input bits L per multiplex frame for EEP HMsym SPP Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,45$	Not used			3 306	Not used	6 948
64 QAM, $R_{all} = 0,55$				4 008		8 424
64 QAM, $R_{all} = 0,72$				5 313		11 167
64 QAM, $R_{all} = 0,78$				5 714		12 012

Table J.12: Number of input bits L per multiplex frame for EEP HMsym SPP Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,45$	Not used			2 196	Not used	4 680
64 QAM, $R_{all} = 0,55$				2 652		5 664
64 QAM, $R_{all} = 0,72$				3 527		7 518
64 QAM, $R_{all} = 0,78$				3 794		8 082

Table J.13: Number of input bits L per hierarchical frame for EEP HMmix VSP Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		1 310	1 473	2 726	3 053
64 QAM, $R_0 = 0,57$			1 496	1 684	3 112	3 488
64 QAM, $R_0 = 0,6$			1 572	1 767	3 270	3 663
64 QAM, $R_0 = 0,66$			1 746	1 964	3 634	4 070

Table J.14: Number of input bits L per hierarchical frame for EEP HMmix VSP Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		1 019	1 162	2 118	2 381
64 QAM, $R_0 = 0,57$			1 164	1 328	2 420	2 720
64 QAM, $R_0 = 0,6$			1 221	1 395	2 541	2 856
64 QAM, $R_0 = 0,66$			1 358	1 550	2 824	3 174

Table J.15: Number of input bits L per hierarchical frame for EEP HMmix VSP Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		916		Not used	1 927
64 QAM, $R_0 = 0,57$			1 044			2 200
64 QAM, $R_0 = 0,6$			1 098			2 313
64 QAM, $R_0 = 0,66$			1 220			2 570

Table J.16: Number of input bits L per hierarchical frame for EEP HMmix VSP Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_0 = 0,5$	Not used		607		Not used	1 297
64 QAM, $R_0 = 0,57$			692			1 480
64 QAM, $R_0 = 0,6$			726			1 554
64 QAM, $R_0 = 0,66$			808			1 728

Table J.17: Number of input bits L per multiplex frame for EEP HMmix SPP Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,48$	Not used		6 288	7 066	13 083	14 650
64 QAM, $R_{all} = 0,58$			7 571	8 506	15 751	17 649
64 QAM, $R_{all} = 0,71$			9 349	10 517	19 461	21 801
64 QAM, $R_{all} = 0,78$			10 244	11 516	21 308	23 872

Table J.18: Number of input bits L per multiplex frame for EEP HMmix SPP Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,48$	Not used		4 885	5 577	10 164	11 425
64 QAM, $R_{all} = 0,58$			5 885	6 717	12 244	13 753
64 QAM, $R_{all} = 0,71$			7 266	8 293	15 121	17 001
64 QAM, $R_{all} = 0,78$			7 960	9 088	16 556	18 620

Table J.19: Number of input bits L per multiplex frame for EEP HMmix SPP Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,48$	Not used			4 395	Not used	9 247
64 QAM, $R_{all} = 0,58$				5 286		11 139
64 QAM, $R_{all} = 0,71$				6 540		13 750
64 QAM, $R_{all} = 0,78$				7 152		15 072

Table J.20: Number of input bits L per multiplex frame for EEP HMmix SPP Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
64 QAM, $R_{all} = 0,48$	Not used			2 908	Not used	6 220
64 QAM, $R_{all} = 0,58$				3 500		7 484
64 QAM, $R_{all} = 0,71$				4 322		9 257
64 QAM, $R_{all} = 0,78$				4 728		10 136

Table J.21: Number of input bits L per SDC block for Mode A

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
16 QAM, $R_{all} = 0,5$	321	366	705	798	1 494	1 680
4 QAM, $R_{all} = 0,5$	161	184	353	399	748	840

Table J.22: Number of input bits L per SDC block for Mode B

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
16 QAM, $R_{all} = 0,5$	246	288	552	630	1 164	1 311
4 QAM, $R_{all} = 0,5$	124	144	276	316	582	656

Table J.23: Number of input bits L per SDC block for Mode C

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
16 QAM, $R_{all} = 0,5$	Not used			564	Not used	1 200
4 QAM, $R_{all} = 0,5$				282		601

Table J.24: Number of input bits L per SDC block for Mode D

Parameters	Spectrum occupancy					
	0	1	2	3	4	5
16 QAM, $R_{all} = 0,5$	Not used			291	Not used	651
4 QAM, $R_{all} = 0,5$				146		326

Annex K (informative): Simulcast transmission, alternate sources, and enhancement signalling

The DRM signal is designed to work in the same broadcast bands as AM signals. Simulcast transmission of services using DRM and AM can be performed by the juxtaposition of the analogue AM signal (DSB or VSB or SSB) and a DRM digital signal.

The DRM system can cross-refer to the same or related services carried in another DRM signal, or in signals using AM, FM or DAB systems. Future quality enhancement is also permitted by the signalling to allow a second DRM multiplex to provide additional data capacity for services. See clause 6 for the detailed explanation of how this information is provided. This facility can be used in many ways, with some examples given in figure K.1.

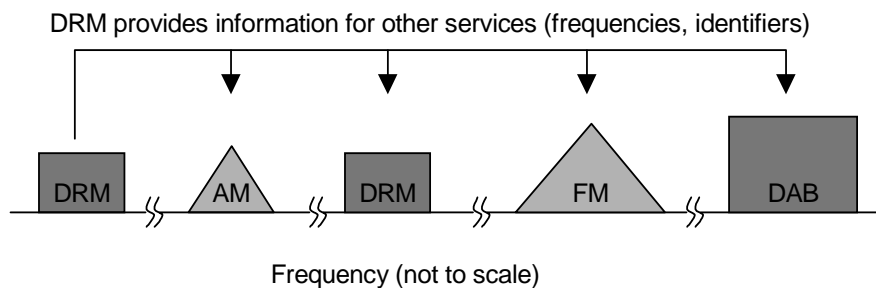


Figure K.1: Example of linking to related services

Figures K.2 and K.3 illustrate some solutions for transmitting the AM and DRM signals from a single transmitter. They can equally be produced by two separate transmitters.

Figure K.2 gives some possibilities for the case where the DRM reference frequency, f_R , is one channel or two channels (i.e. ± 9 kHz, ± 10 kHz, -18 kHz or -20 kHz) from the AM carrier frequency, f_C , and figure K.3 gives some possibilities for the case where the DRM reference frequency, f_R , is nominally half a channel from the AM carrier frequency, f_C . Due to the requirement to position the DRM reference frequency on an integer multiple of 1 kHz, the DRM reference frequency and the AM carrier frequency shall be either 4 kHz or 5 kHz apart.

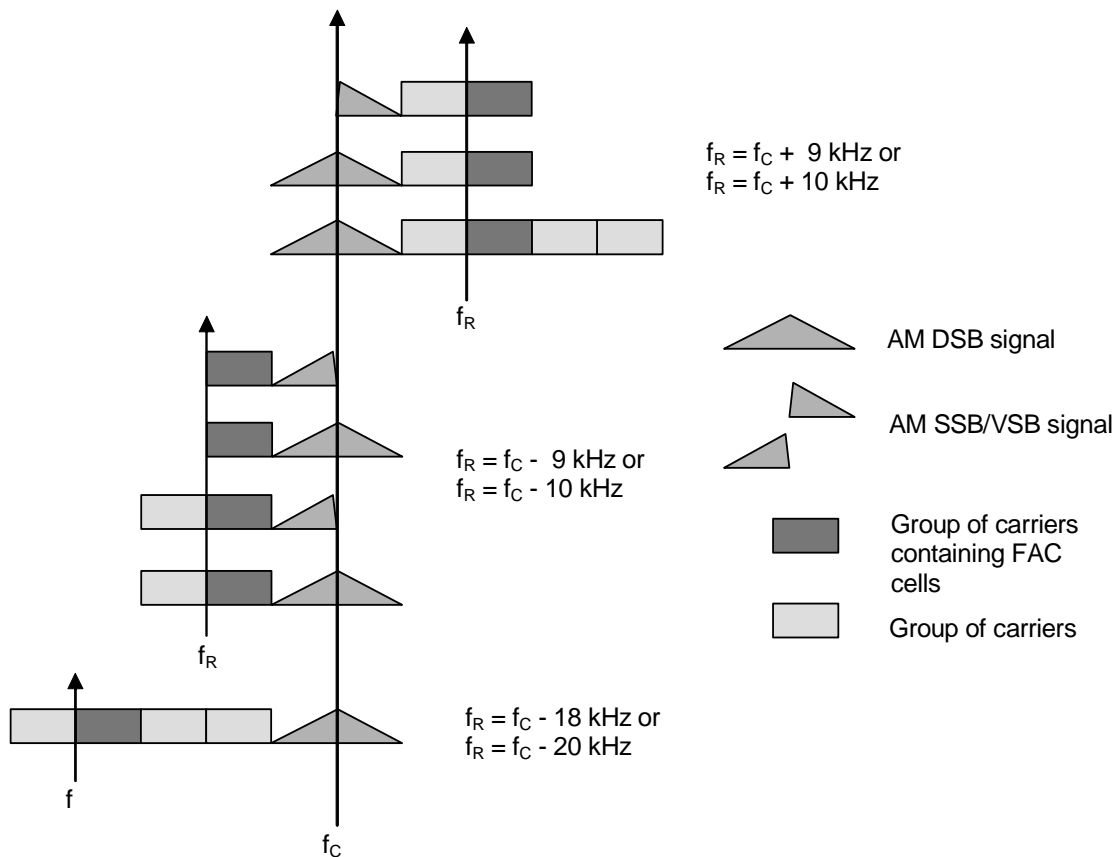


Figure K.2: Example simulcast modes for whole channel offsets

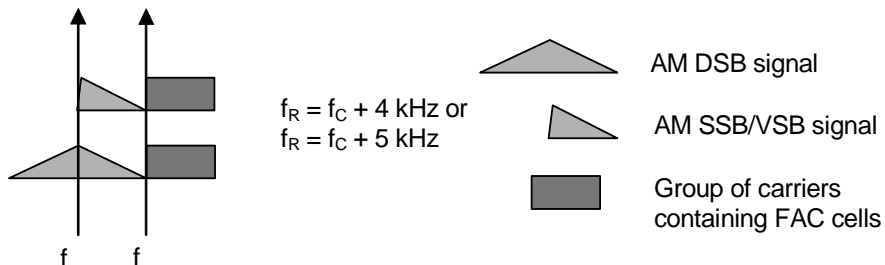


Figure K.3: Example simulcast modes for half channel offsets

Figures K.4 to K.6 illustrate some examples of the use of the base/enhancement signalling of DRM to provide solutions for transmitting higher quality DRM or DRM and AM signals from a single transmitter.

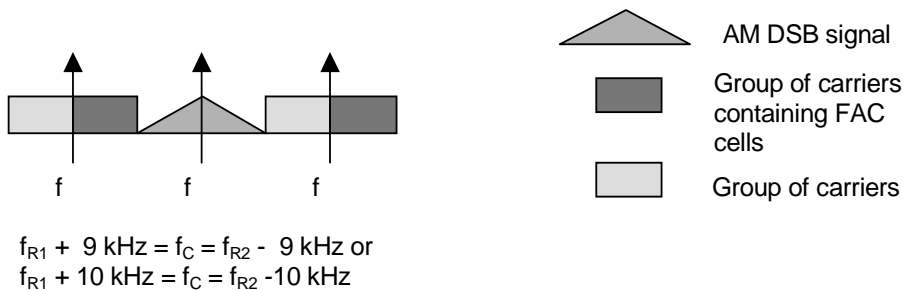


Figure K.4: Example simulcast modes with two DRM signals

In figure K.4, the two DRM signals may be alternate frequencies for the same multiplex, providing spectral diversity. In this case, the AFS data entities (see clause 6) indicate the two frequencies that the DRM multiplex is available on. Alternatively, the two DRM signals may be the base layer and enhancement layer for a multiplex. In this case the AFS data entities signal the frequency of the other layer. All receivers will be able to decode the base layer, whilst some will also be able to decode the enhancement layer and provide higher quality.

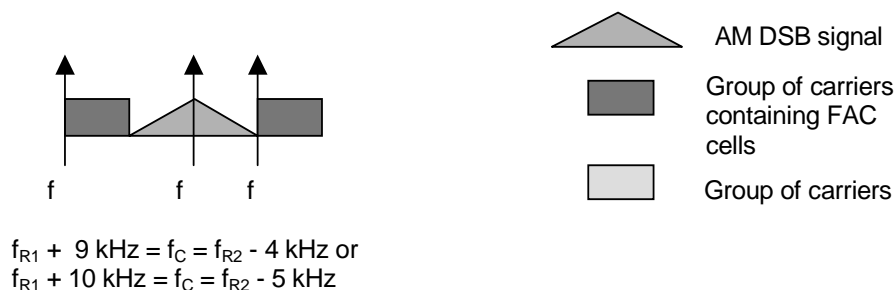


Figure K.5: Example simulcast modes with two DRM signals

Figure K.5 illustrates the possible use of two half channel DRM signals.



Figure K.6: Example of base/enhancement DRM signals

Figure K.6 illustrates the possible use of two DRM signals as a combination of a base layer and enhancement layer for a multiplex. In this case the AFS data entities signal the frequency of the other layer. All receivers will be able to decode the base layer, whilst some will also be able to decode the enhancement layer and provide higher quality.

Annex L (informative): Pilot reference illustrations

The figures below show the position of the gain reference cells (character "O") for nominal channel bandwidths of up to 10 kHz (spectrum occupancy parameter = 0, 1, 2 or 3). The patterns continue to the right for the 18 kHz and 20 kHz nominal channel bandwidth options. (Spectrum occupancy parameter = 4 or 5).

Mode A:

```
+--> carriers (k)
|
v
symbols (s)                DC
                            :
negative frequencies        :        positive frequencies
1..1                        :                        1..1
1..0..3333333322222222111111111000000000:00000000011111111122222222233333333..0..1
4..2..8765432109876543210987654321:12345678901234567890123456789012345678..2..4
:
0 . . . O.....O.....:O.....O..... O .
1 O . . . .O.....O.....:O.....O..... . .
2 . . . . .O.....O.....:O.....O..... . .
3 . . . . .O.....O.....:O.....O..... . .
4 . . O . . . .O.....O.....:O.....O..... O O .
5 . . . O.....O.....:O.....O..... . .
6 O . . . .O.....O.....:O.....O..... . .
7 . . . . .O.....O.....:O.....O..... . .
8 . . . . .O.....O.....:O.....O..... O O .
9 . O . . . .O.....O.....:O.....O..... O O .
10 . . O . . . .O.....O.....:O.....O..... . .
11 O . . . .O.....O.....:O.....O..... . .
12 . . . . .O.....O.....:O.....O..... . .
13 . . . . .O.....O.....:O.....O..... O O .
14 . O . . . .O.....O.....:O.....O..... O O .
```

Mode B:

```
+--> carriers (k)
|
v
symbols (s)                DC
                            :
negative frequencies        :        positive frequencies
1                            :                            1
0..9..3333333322222222111111111000000000:00000000011111111122222222233333333..9..0
3..1..8765432109876543210987654321:12345678901234567890123456789012345678..1..3
:
0 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... O O O
1 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
2 O O . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
3 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... O O O
4 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
5 O O . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
6 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... O O O
7 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
8 O O . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
9 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... O O O
10 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
11 O O . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
12 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... O O O
13 . . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
14 O O . . . .O.....O.....O.....O.....O.....O.....:O.....O.....O.....O.....O.....O..... . .
```



```

DC (not used)
:
:00000000011111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000
:12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
:
0 :::o.....f...o.....o.....f.....o.....f.....o.....o.....
1 :::.....f.....o.....o.....o.....f.....o.....f.....o.....o.....
2 :::.....o.....f.....x...o.....o.....x...o.....f.....x...o...f.....x...o.....
3 :::.....x...o...f.....x...o.....x...f.....x...f...o.....x...o.....
4 :::.....x...f...x.....x...o.....f...o...x...f...x...o.....x...o.....
5 :::o.....f...o...x.....x...o.....f...x...o...x...f...x...o.....o.....
6 :::.....f...x...o...x.....x...o.....f...x...o...x...f...x...o.....
7 :::.....o.....f.....x...o...x.....x...o...f.....x...o...f...x...x...o.....
8 :::.....x...o...f.....x...o...x.....x...o...f...x...o...x...f...o...x...o.....
9 :::.....x...f...x.....x...o...x.....f...o...x.....f...x...o...x...o.....
10 :::o.....f...o...x.....x...o...x.....f...o...x...f...o...x...o.....o.....
11 :::.....x...o...f.....o...x.....o...x...f.....o...x...f.....o...x...o.....
12 :::.....o...x...f.....o...x.....o...f.....o...f...x.....o...x.....
13 :::.....o...f.....o...x.....f...x.....f...o...x.....o.....o.....
14 :::.....f.....o.....o.....f...o.....f...o.....f...o.....o.....
    
```

Mode B: positions for pilot cells

```

+--> carriers (k)
|
v
symbols (s)
    
```

```

DC (not used)
:
:0000000001111111112222222222333333333344444444445555555555666666666677777777778888888888999
:1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
:
0 :::o.....o...f...o.....o.....o.....o.....fo.....o.....o...f...o.....o.....o.....o.....o.....
1 :::o.....o...of.....o.....o.....o.....o.....o.....f...o.....o.....of.....o.....o.....o.....o.....
2 :::.....o...o...x...fo.....o...x...o.....o...o...x...of.....o...x...o.....fo...x...o.....o.....o.....
3 :::o.....o...o...xf...o.....o...x...o.....o...o...x...fo.....o...x...o...f...o...x...o.....o.....o.....
4 :::o.....o...o...ofx...o.....o...x...o.....o...o...o...xf...o.....o...x...of.....o...x...o.....o.....
5 :::.....o...o...fo...x...o.....o...x...o.....o...o...ofx...o.....o...o...x...fo.....o...x...o.....
6 :::o.....o...x...o...f...o...x...o.....o...x...o.....fo...x...o.....o...xf...o.....o...x...o.....
7 :::.....o...x...o...of.....o...x...o.....o...x...o.....o...f...o...x...o...ofx...o.....o...x...o.....
8 :::.....o...o...x...fo.....o...x...o.....o...x...o.....of.....o...x...o...fo...x...o.....o...x...o.....
9 :::o.....o...o...ofx...o.....o...x...o.....o...o...o...fo.....o...x...o...f...o...x...o.....o...x...o.....
10 :::.....o...o...ofx...o.....o...x...o.....o...x...o...f...o.....o...x...of.....o...x...o.....o...x...o.....
11 :::.....o...o...fo...x...o.....o...x...o.....o...x...of.....o.....o...x...fo.....o...x...o.....o.....
12 :::o.....o...o...f...o...x...o.....o...x...o...fo.....o.....o...xf...o.....o...x...o.....o.....o.....
13 :::.....o...o...of...o...x...o.....o...x...o...o...o...of...o.....o...ofx...o.....o...x...o.....o.....
14 :::.....o...o...fo.....o.....o.....o.....o...of.....o.....o...fo.....o.....o.....o.....o.....
    
```

Mode C: positions for pilot cells

```

+--> carriers (k)
|
v
symbols (s)
    
```

```

DC (not used)
:
:00000000011111111122222222223333333333444444444455555555556666666666
:123456789012345678901234567890123456789012345678901234567890123456789
:
0 :::o.....o...f...o.....o.....o...f...o.....fo.....o.....o.....o.....o.....o.....
1 :::.....o...f...o.....o.....o...f...o.....of.....o.....o.....o.....o.....o.....
2 :::o.....o...f...o.....o.....o...f...o.....fo.....o.....o.....o.....o.....o.....
3 :::.....o...x...f...o.....o...x...o...f...o.....ofx...o.....o...x...o.....o.....
4 :::.....o...o...f...o.....o...x...o...f...x...o...fo...x...o.....o.....o.....
5 :::.....o...f...x...o...o...x...o...f...o...x...of...o...x...o.....o.....o.....
6 :::o.....o...f...o...x...o...o...x...o...f...o...x...fo...o...x...o.....o.....
7 :::.....o...x...o...f...o...x...o...o...x...o...f...o...x...of...o...x...o.....o.....
8 :::.....o...x...o...f...o...o...x...o...o...x...f...o...o...xfo...o...x...o.....o.....
9 :::.....o...x...f...o...o...x...o...o...f...o...o...ofx...o...o...x...o.....o.....
10 :::o.....o...f...o.....o...x...o...f...x...o...fox...o.....o.....o.....o.....
11 :::.....o...f...x...o...o...x...o...f...o...x...of...o...x...o.....o.....o.....
12 :::.....o...o...f...o...x...o...o...x...o...f...o...x...fo...o...x...o.....o.....
13 :::.....o...x...o...f...o...x...o...o...x...o...f...o...x...of...o...x...o.....o.....
14 :::o.....o...x...o...f...o...o...x...o...o...x...f...o...o...xfo...o...x...o.....o.....
15 :::.....o...x...f...o...o...x...o...o...f...o...o...ofx...o...o...x...o.....o.....
16 :::.....o...o...f...o...o...x...o...o...f...x...o...fo...x...o.....o.....o.....
17 :::.....o...o...f...x...o...o...x...o...f...o...x...of...o...x...o.....o.....o.....
18 :::o.....o...o...f...o...x...o...o...x...o...f...o...x...fo...o...x...o.....o.....
19 :::.....o...o...f...o.....o.....o...f...o.....of.....o.....o.....o.....o.....
    
```

Mode D: positions for pilot cells

+--> carriers (k)

|

v

symbols (s)

DC (not used)

```

:
:0000000001111111112222222222333333333344444
:12345678901234567890123456789012345678901234
:
0 :o..o..f..o..o..o..o..fo..o..f..o..o..o..o..o..
1 :o..o..fo..o..o..o..of..o..o..fo..o..o..o..o..o..
2 :..o..of..o..o..o..o..f..o..of..o..o..o..o..o..o..
3 :o..o..f..xo..o..o..xo..fo..o..xf..o..o..o..o..o..o..
4 :o..o..fo..xo..o..o..xof..o..o..fo..o..o..o..o..o..o..
5 :..o..of..o..xo..o..o..xf..o..ofxo..o..o..o..o..o..
6 :o..o..f..o..xo..o..o..fo..o..f..xo..o..o..o..o..o..
7 :o..o..fo..o..xo..o..o..ofxo..o..fo..xo..o..o..o..o..
8 :..o..xof..o..o..xo..o..f..xo..of..o..xo..o..o..o..o..
9 :o..o..xf..o..o..xo..o..fo..xo..f..o..xo..o..o..o..o..
10 :o..o..fo..o..o..o..xo..of..o..xo..fo..o..xo..o..o..o..o..
11 :..o..ofxo..o..o..o..xo..f..o..xof..o..o..xo..o..o..o..
12 :o..o..f..xo..o..o..o..xo..fo..o..xf..o..o..o..xo..o..o..
13 :o..o..fo..xo..o..o..xof..o..o..fo..o..o..o..xo..o..o..
14 :..o..of..o..xo..o..o..xf..o..ofxo..o..o..o..o..o..o..
15 :o..o..f..o..xo..o..o..fo..o..f..xo..o..o..o..o..o..o..
16 :o..o..fo..o..xo..o..o..ofxo..o..fo..xo..o..o..o..o..o..
17 :..o..xof..o..o..xo..o..f..xo..of..o..xo..o..o..o..o..
18 :o..o..xf..o..o..o..xo..o..fo..xo..f..o..xo..o..o..o..o..
19 :o..o..fo..o..o..o..xo..of..o..xo..fo..o..xo..o..o..o..o..
20 :..o..ofxo..o..o..o..xo..f..o..xof..o..o..xo..o..o..o..
21 :o..o..f..xo..o..o..o..xo..fo..o..xf..o..o..o..xo..o..o..o..
22 :o..o..fo..xo..o..o..xof..o..o..fo..o..o..o..xo..o..o..o..
23 :..o..of..o..o..o..o..o..f..o..of..o..o..o..o..o..o..o..

```

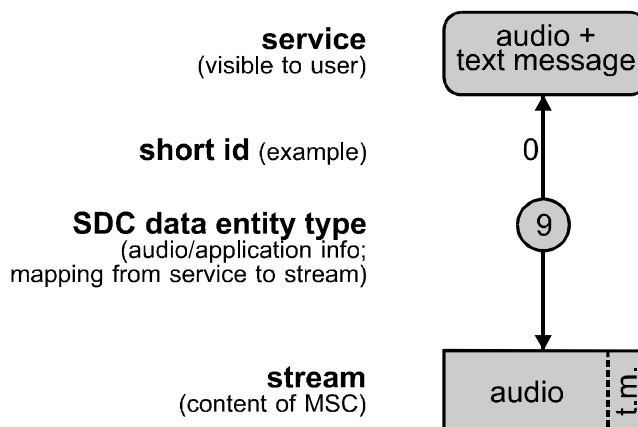
Annex M (informative): MSC configuration examples

The examples below demonstrate some possibilities for configuring the MSC. Especially the mapping from services to audio or data streams is covered together with some limitations which must be respected when assembling the DRM multiplex.

General Preface:

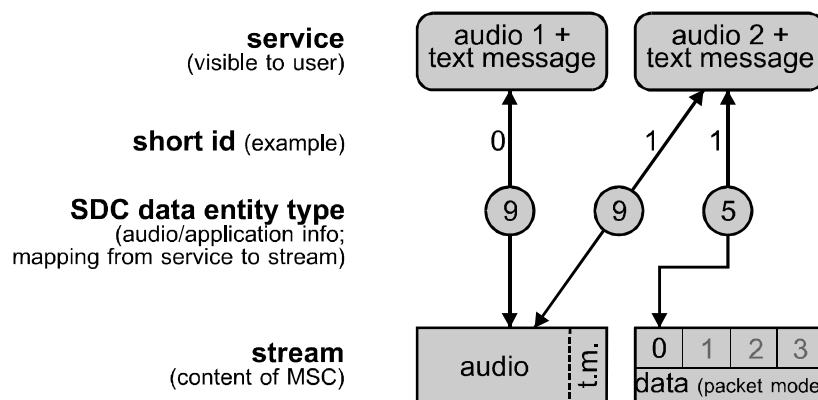
- The DRM multiplex may contain up to four streams in the MSC, each carrying audio or data information.
- An audio stream is described by SDC data entity type 9. A packet mode data stream consists of 1 to 4 "sub-streams" (distinguished by their packet id), each described by SDC data entity type 5.
- 1 to 4 services can be signalled to the user. A data service points to one data (sub-)stream. An audio service points to one audio stream plus optionally to text message information (contained in the audio stream) and/or one data (sub-)stream.
- Audio services are mapped to audio streams by SDC data entity type 9. Data (and audio) services are mapped to data streams by SDC data entity type 5.
- If several services point to the same stream, the stream configuration in SDC data entity type 5 or 9 must be identical.

EXAMPLE 1: A very simple DRM multiplex consists of just one single audio service pointing to the one and only audio stream. The audio stream may contain text messages.



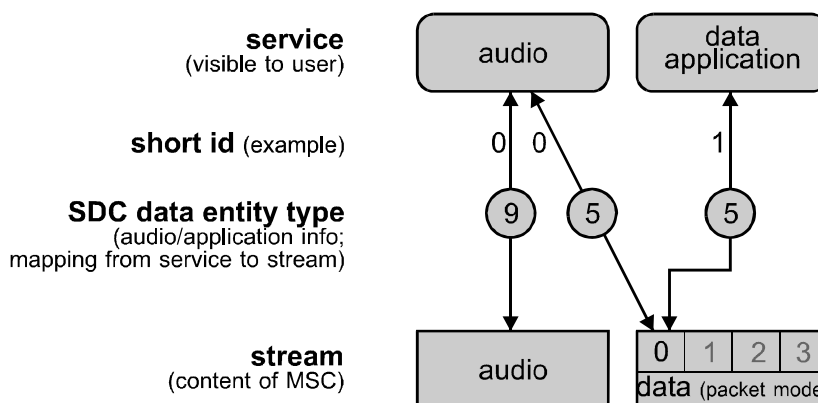
EXAMPLE 2: There are two audio services. Both point to the same audio stream. One of these services points to an additional packet mode data sub-stream carrying a multimedia data application. In total there are one audio stream and one data stream using packet mode.

Note that if multiple services point to the same stream the configuration for that stream (carried in SDC data entities type 9 for audio information or in SDC data entities type 5 for data application information) must be the same. So in this example both or none of the audio services can comprise the text message service.



EXAMPLE 3: There is one audio service and one data service signalled to the user. The DRM multiplex consists of one audio stream and one data stream in packet mode containing one sub-stream. The data service points to the data application carried in the one sub-stream of the data stream. The audio service points to the audio stream and additionally also to the one sub-stream of the data stream.

Note that in this case the data application description in the two SDC data entities type 5 must be identical for both services referencing the same packet mode sub-stream.



EXAMPLE 4: To make maximum use of the possibilities of a DRM multiplex, the following scenario is possible:

There are three audio services each pointing to its own audio stream. In addition to audio, each of these three audio streams carries text messages. Every audio service also points to its own data application carried as a sub-stream of a packet mode data stream (being the fourth stream in the DRM multiplex). Since a data stream can carry up to four sub-streams in packet mode, there can be an additional data service pointing to the fourth sub-stream of the data stream.

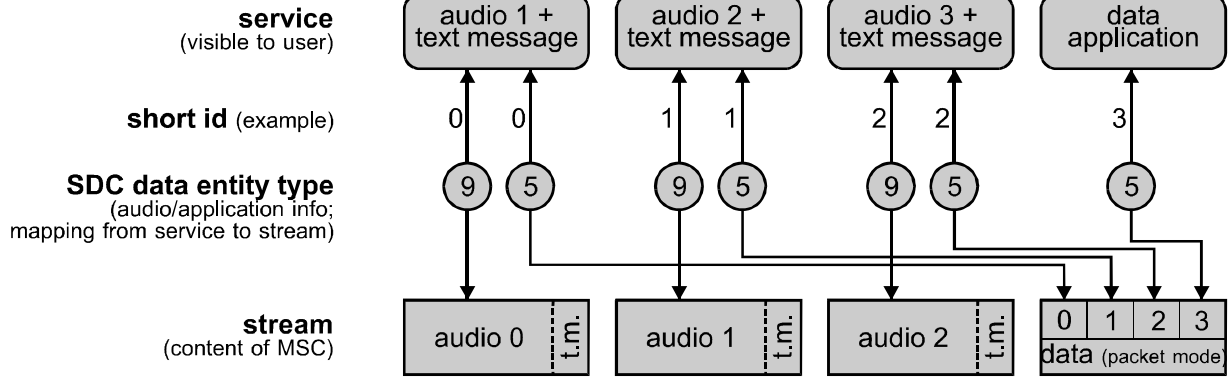
In total there are:

- three audio services;
- one data service.

These four services point to 10 different "logical channels":

- three different audio streams with their own text messages;
- one data stream in packet mode with four sub-streams.

NOTE: The packet mode configuration parameters (e.g. the packet length) of all four SDC data entities type 5 (describing the four sub-streams of the one packet mode data stream) must be identical!.



Annex N (informative): HVXC parameters

Table N.1: HVXC source coder parameters

Name	Meaning	Number of bits	Fixed Rate	
			2,0 kbit/s	4,0 kbit/s
LSP1	LSP index 1	5	X	X
LSP2	LSP index 2	7	X	X
LSP3	LSP index 3	5	X	X
LSP4	LSP index 4	1	X	X
VUV	voiced/unvoiced flag	2	X	X
Pitch	pitch parameter	7	X	X
SE_shape1	spectrum index 0	4	X	X
SE_shape2	spectrum index 1	4	X	X
SE_gain	spectrum gain index	5	X	X
VX_shape1[0]	stochastic codebook index 0	6	X	X
VX_shape1[1]	stochastic codebook index 1	6	X	X
VX_gain1[0]	gain codebook index 0	4	X	X
VX_gain1[1]	gain codebook index 1	4	X	X
LSP5	LSP index 5	8		X
SE_shape3	4k spectrum index 0	7		X
SE_shape4	4k spectrum index 1	10		X
SE_shape5	4k spectrum index 2	9		X
SE_shape6	4k spectrum index 3	6		X
VX_shape2[0]	4k stochastic codebook index 0	5		X
VX_shape2[1]	4k stochastic codebook index 1	5		X
VX_shape2[2]	4k stochastic codebook index 2	5		X
VX_shape2[3]	4k stochastic codebook index 3	5		X
VX_gain2[0]	4k gain codebook index 0	3		X
VX_gain2[1]	4k gain codebook index 1	3		X
VX_gain2[2]	4k gain codebook index 2	3		X
VX_gain2[3]	4k gain codebook index 3	3		X

NOTE: X indicates that the parameter is used in the corresponding mode.

Table N.2: Bit allocations of 2,0/4,0 kbit/s HVXC coder (fixed rate)

Parameter	2,0 kbit/s (fixed rate)		4,0 kbit/s (fixed rate)	
	Voiced	Unvoiced	Voiced	Unvoiced
LSP	18 bits/20 ms	18 bits/20 ms	18 bits/20 ms	18 bits/20 ms
LSP(enh)			8 bits/20 ms	8 bits/20 ms
V/UV	2 bits/20 ms	2 bits/20 ms	2 bits/20 ms	2 bits/20 ms
Pitch	7 bits/20 ms		7 bits/20 ms	
spectral shape	4 + 4 bits/20 ms		4 + 4 bits/20 ms	
spectral gain	5 bits/20 ms		5 bits/20 ms	
spectral shape(enh)			32 bits/20 ms	
VXC shape		6 bits/10 ms		6 bits/10 ms
VXC gain		4 bits/10 ms		4 bits/10 ms
VXC shape(enh)				5 bits/5 ms
VXC gain(enh)				3 bits/5 ms
Total - 2,0 kbit/s	40 bits/20 ms	40 bits/20 ms		
Total - 4,0 kbit/s			80 bits/20 ms	80 bits/20 ms

Annex O (normative): Interpretation of schedules for Alternative Frequency Signalling

The "Alternative frequency signalling: Schedule definition data entity - type 4" provides the functionality to restrict the availability of a list of alternative frequencies to certain time intervals based on a weekly schedule.

In every SDC data entity type 4 the following information can be signalled:

- With the Day Code field it can be indicated to which days of the week (Monday to Sunday) the following time range shall apply. Any day-combination from 1 to 7 days can be signalled.
- Using the Start Time and the Duration value, a time interval can be specified. This time interval applies to all specified days-of-the-week (using the Day Code).
The Start Time value indicates the minutes since midnight UTC (for every indicated day of the week), ranging from 00:00 to 23:59.
The Duration value specifies the number of minutes after (and including) the start time. It can potentially span more than one week. So for example it is possible to cover a full weekend using one single SDC data entity type 4.
- More than one time interval per day or different day-time-combinations can be specified by broadcasting multiple SDC data entities type 4 with the same Schedule Id (using the list mechanism for the version flag).

Every receiver has to evaluate these values in a consistent way. Therefore the following text defines how the receiver has to interpret the SDC "Alternative frequency signalling: Schedule description data entities - type 4". The function (presented in pseudo program code notation) checks whether the current time/date is within a scheduled time interval:

```
// input: time_in_week (minutes since last Monday 00:00 in UTC;
//           the value is in the range  $0 \leq \text{time\_in\_week} < 60 \times 24 \times 7$ );
//           schedule_id (id of the schedule to be checked;
//           the value is in the range  $0 \leq \text{schedule\_id} \leq 15$  )
// output: boolean value (true: time_in_week is inside schedule)

bool IsInsideSchedule (time_in_week, schedule_id)
{
  // the schedule_id value 0 is fixed to 'always valid':
  if (schedule_id == 0)
  {
    return true
  }

  for every SDC entity with the given schedule_id
  {
    extract (day_code, start_time, duration) from SDC entity

    for every day specified by day_code
    {
      // minutes_since_monday(day) returns the number of minutes
      // of the start (00:00) of the indicated day since Monday 00:00
      // (it is a multiple of  $24 \times 60$ )
      // e.g. Monday    ->  $0 \times 24 \times 60 = 0$ 
      //      Tuesday   ->  $1 \times 24 \times 60 = 1\ 440$ 
      //      Wednesday ->  $2 \times 24 \times 60 = 2\ 880$ , etc.

      schedule_start = minutes_since_monday (day) + start_time
      schedule_end   = schedule_start + duration

      // the normal check (are we inside start and end?):
      if (time_in_week >= schedule_start AND
          time_in_week <= schedule_end)
      {
        return true
      }

      // the wrap-around check:
      minutes_per_week =  $7 \times 24 \times 60$ 
      if (schedule_end > minutes_per_week)
    }
  }
}
```

```
    {
      // our duration wraps into next Monday (or even later)
      if (time_in_week < (schedule_end - minutes_per_week))
      {
        return true
      }
    }
  }
}
return false
}
```

The encoding for a certain time interval is not unique. A 48-hour interval starting on Wednesday 10:00 could be encoded as:

- *"Wednesday and Thursday; start time 10:00; duration 24 hours"*.
- *"Wednesday; start time 10:00; duration 48 hours"*.
- or using two SDC data entities with the same schedule id:
"Wednesday; start time 10:00; duration 24 hours" and *"Thursday; start time 10:00; duration 24 hours"*.
- or using two SDC data entities with the same schedule id:
"Wednesday; start time 10:00; duration 10 hours" and *"Wednesday; start time 20:00; duration 38 hours"*.

It is up to the encoding side to describe a certain schedule with as few SDC data entities as possible.

Annex P (informative): Bibliography

- DRM Output Requirements, November 1999.
- ITU-R Circular letter LCCE/39: "Submission of candidate systems for digital sound broadcasting at frequencies below 30 MHz, September 1999".
- ITU-R Recommendation BS.1348: "Service requirements for digital sound broadcasting at frequencies below 30 MHz".
- ITU-R Recommendation BS.1349: "Implementation of digital sound broadcasting to vehicular, portable and fixed receivers using terrestrial transmitters in the LF, MF and HF bands".
- CENELEC EN 62106: "Specification of the Radio Data System (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 to 108,0 MHz".
- ITU-R Recommendation BS.559-2: "Objective measurement of radio-frequency protection ratios in LF, MF and HF broadcasting".
- ITU-R Recommendation SM.328-10: "Spectra and bandwidth of emissions".
- ITU-R Recommendation BS.560: "Radio-frequency protection ratios in LF, MF and HF broadcasting".

History

Document history		
V1.1.1	September 2001	Publication as TS 101 980
V1.2.2	April 2003	Publication
V2.1.1	April 2004	Membership Approval Procedure MV 20040618: 2004-04-20 to 2004-06-18
V2.1.1	June 2004	Publication