# ETSI TS 101 376-5-3 V2.2.1 (2005-03)

*Technical Specification*

**GEO-Mobile Radio Interface Specifications (Release 2)**
**General Packet Radio Service;**
**Part 5: Radio interface physical layer specifications;**
**Sub-part 3: Channel Coding;**
**GMPRS-1 05.003**

Reference

RTS/SES-00230-5-3

Keywords

coding, GMPRS, GMR, GPRS, GSM, GSO, interface, MES, mobile, MSS, radio, satellite, S-PCN

***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

# Contents

# Intellectual Property Rights

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

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

# Foreword

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

The contents of the present document are subject to continuing work within TC-SES and may change following formal TC-SES approval. Should TC-SES modify the contents of the present document it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version 2.m.n

where:

- the third digit (n) is incremented when editorial only changes have been incorporated in the specification;
- the second digit (m) is incremented for all other types of changes, i.e. technical enhancements, corrections, updates, etc.

The present document is part 5, sub-part 3 of a multi-part deliverable c covering the GEO-Mobile Radio Interface Specifications (Release 2) General Packet Radio Service, as identified below:

Part 1: "General specifications";

Part 2: "Service specifications";

Part 3: "Network specifications";

Part 4: "Radio interface protocol specifications";

**Part 5: "Radio interface physical layer specifications":**

Sub-part 1: "Physical Layer on the Radio Path: General Description";

Sub-part 2: "Multiplexing and Multiple Access; Stage 2 Service Description";

**Sub-part 3: "Channel Coding";**

Sub-part 4: "Modulation";

Sub-part 5: "Radio Transmission and Reception";

Sub-part 6: "Radio Subsystem Link Control";

Sub-part 7: "Radio Subsystem Synchronization";

Part 6: "Speech coding specifications";

Part 7: "Terminal adaptor specifications".

# Introduction

GMR stands for GEO (Geostationary Earth Orbit) Mobile Radio interface, which is used for mobile satellite services (MSS) utilizing geostationary satellite(s). GMR is derived from the terrestrial digital cellular standard GSM and supports access to GSM core networks.

The present document is part of the GMR Release 2 specifications. Release 2 specifications are identified in the title and can also be identified by the version number:

- Release 1 specifications have a GMR-1 prefix in the title and a version number starting with "1" (V1.x.x.).
- Release 2 specifications have a GMPRS-1 prefix in the title and a version number starting with "2" (V2.x.x.).

The GMR release 1 specifications introduce the GEO-Mobile Radio interface specifications for circuit mode mobile satellite services (MSS) utilizing geostationary satellite(s). GMR release 1 is derived from the terrestrial digital cellular standard GSM (phase 2) and it supports access to GSM core networks.

The GMR release 2 specifications add packet mode services to GMR release 1. The GMR release 2 specifications introduce the GEO-Mobile Packet Radio Service (GMPRS). GMPRS is derived from the terrestrial digital cellular standard GPRS (included in GSM Phase 2+) and it supports access to GSM/GPRS core networks.

Due to the differences between terrestrial and satellite channels, some modifications to the GSM standard are necessary. Some GSM specifications are directly applicable, whereas others are applicable with modifications. Similarly, some GSM specifications do not apply, while some GMR specifications have no corresponding GSM specification.

Since GMR is derived from GSM, the organization of the GMR specifications closely follows that of GSM. The GMR numbers have been designed to correspond to the GSM numbering system. All GMR specifications are allocated a unique GMR number. This GMR number has a different prefix for Release 2 specifications as follows:

- Release 1: GMR-n xx.zyy.
- Release 2: GMPRS-n xx.zyy.

where:

- xx.0yy (z = 0) is used for GMR specifications that have a corresponding GSM specification. In this case, the numbers xx and yy correspond to the GSM numbering scheme.
- xx.2yy (z = 2) is used for GMR specifications that do not correspond to a GSM specification. In this case, only the number xx corresponds to the GSM numbering scheme and the number yy is allocated by GMR.
- n denotes the first (n = 1) or second (n = 2) family of GMR specifications.

A GMR system is defined by the combination of a family of GMR specifications and GSM specifications as follows:

- If a GMR specification exists it takes precedence over the corresponding GSM specification (if any). This precedence rule applies to any references in the corresponding GSM specifications.

NOTE: Any references to GSM specifications within the GMR specifications are not subject to this precedence rule. For example, a GMR specification may contain specific references to the corresponding GSM specification.

- If a GMR specification does not exist, the corresponding GSM specification may or may not apply. The applicability of the GSM specifications is defined in GMPRS-1 01.201 [2].

# 1 Scope

The present document specifies the data blocks given to the encryption unit and the mapping onto the free bits of a burst. It includes the specifications for encoding, reordering, interleaving, and detailed mapping onto the burst. It does not specify the channel decoding method. The definition is given for each kind of logical channel, starting with the data provided to the channel encoder by the speech coder, the data terminal equipment, or the controller of the Mobile Earth Station (MES).

# 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] GMPRS-1 01.004 (ETSI TS 101 376-1-1): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service (GMPRS); Part 1: General specifications; Sub-part 1: Abbreviations and acronyms".

[2] GMPRS-1 01.201 (ETSI TS 101 376-1-2): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service (GMPRS); Part 1: General specifications; Sub-part 2 : Introduction to the GMR-1 family".

[3] GMR-1 05.003: (ETSI TS 101 376-5-3) (V1.2.1): "GEO-Mobile Radio Interface Specifications; Part 5: Radio interface physical layer specifications; Sub-part 3: Channel Coding".

NOTE: This is a reference to a GMR-1 Release 1 specification. See the introduction for more details.

[4] GMPRS-1 04.008 (ETSI TS 101 376-4-8): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service; Part 4: Radio interface protocol specifications; Sub-part 8: Mobile Radio Interface Layer 3 Specifications".

[5] GMPRS-1 04.060 (ETSI TS 101 376-4-12): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service; Part 4: Radio interface protocol specifications; Sub-part 12: Mobile Earth Station (MES) - Base Station System (BSS) interface; Radio Link Control/ Medium Access Control (RLC/MAC) protocol".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in GMPRS-1 01.201 [2] apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in GMPRS-1 01.004 [1] apply.

# 4 General

## 4.1 General organization

Same as clause 4.1 in GMR-1 05.003 [3].

## 4.2 Naming convention

Same as clause 4.2 in GMR-1 05.003 [3].

**Table 4.1: Void**

## 4.3 Parity checking

Same as clause 4.3 in GMR-1 05.003 [3], except table 4.2.

Table 4.2 indicates the CRC polynomials used in GMR-1 channels.

**Table 4.2: CRC polynomials used in GMR-1**

| Channel | $g_8(D)$ | $g_{12}(D)$ | $g_{16}(D)$ |
|---|---|---|---|
| BCCH | | | X |
| PCH | | | X |
| AGCH | | | X |
| RACH | X | X | |
| CBCH | | | X |
| SDCCH | | | X |
| SACCH | | | X |
| FACCH3 | | | X |
| FACCH6 | | | X |
| FACCH9 | | | X |
| TACCH | | | X |
| GBCH | | | X |
| PDCH | | | X |
| PRACH | X | | |

## 4.4 Convolutional coding

### 4.4.1 Convolutional encoding (all channels except TCH3)

Same as clause 4.4.1 in GMR-1 05.003 [3].

### 4.4.2 Convolutional encoding for TCH3

Same as clause 4.4.2 in GMR-1 05.003 [3].

### 4.4.3 Viterbi decoder for TCH3

Same as clause 4.4.3 in GMR-1 05.003 [3].

# 4.5 Puncturing and repetition

The number of available free bits on a burst may not equal the number of coded bits output by the convolutional encoder. In this case, selected coded bits are either punctured (not processed for transmission) or repeated (transmitted twice) as needed to match the coded output to the available payload. The coded bits to be punctured and/or repeated are specified by channel-dependent puncturing and repetition masks. These masks take the form of an $n \times L$ integer array, in which the $i^{th}$ row applies to the coded bits produced by the $g_i$(D) generator polynomial, $i = 0, \ldots, n-1$, and each entry specifies the number of times that the corresponding coded bit is to be transmitted. The parameter $L$ denotes the period of the pattern. If the period is less than the total number of encoder input (information plus tail bits), the mask is reapplied on a periodic basis. If the number of encoder input is not divisible by $L$, the mask applies on all the encoder input and stops at the end of the encoder input. In some instances, prefix and suffix masks are applied at the beginning and end of the burst, respectively, to facilitate the rate matching.

The puncturing and repetition masks used in GMR-1 for the Rate 1/2, Rate 1/3, and Rate 1/5 convolutional code (K = 5) are listed in tables 4.3, 4.4 and 4.5 respectively. The puncturing masks used for Rate 1/2 convolutional code with constraint length K = 7 are listed in table 4.6. The puncturing masks used for Rate 1/2 convolutional code with constraint length K = 9 are listed in table 4.7. The identifier P($r$;$L$) denotes the preferred puncturing mask used for circuit-switched services in which $r$ coded bits are punctured every $L$ input bits to the convolutional encoder. The time-reversed version of this mask is denoted by P*($r$;$L$).

The puncturing/repetition masks used for convolutionally-encoded packet-switched services are formed as concatenations of the five basic masks denoted A, B, C, D, and E in table 4.3. This allows complex puncturing and repetition patterns to be specified by short mathematical descriptions. For example, the composite mask P = A(BC)$^{10}$D denotes the puncture pattern in which mask A is applied at the beginning of the burst; the combination of mask B followed by mask C is applied 10 times during the middle of the burst; and, finally, mask D is applied at the end of the burst.

**Table 4.3: GMR-1 puncturing and repetition masks for the rate 1/2 convolutional code (K = 5)**

| Identifier | Mask | Remark |
|---|---|---|
| P(2;3) | $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 3/4. |
| P(2;5) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 5/8. |
| P*(2;5) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$ | Time-reversal of the puncturing mask P(2;5). |
| P(3;11) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 11/19. |
| P(4;12) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 12/20. |
| P*(4;12) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ | Time-reversal of the puncturing mask P(4;12). |
| P(1;2) | $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 2/3. |
| A | $\begin{bmatrix} 1111 \\ 1111 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 1/2 (no puncturing). |
| B | $\begin{bmatrix} 1111 \\ 0111 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 4/7. |
| C | $\begin{bmatrix} 1111 \\ 0101 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 2/3. |
| D | $\begin{bmatrix} 0101 \\ 1011 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 4/5. |

| Identifier | Mask | Remark |
|---|---|---|
| E | $\begin{bmatrix} 1111 \\ 2111 \end{bmatrix}$ | Repetition mask that, if applied repetitively, produces effective code rate = 4/9. |

**Table 4.4: GMR-1 puncturing masks for the rate 1/3 convolutional code (K = 5)**

| Identifier | Mask | Remark |
|---|---|---|
| P(1;6) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively for NT6 burst punctures 24 bits giving an effective code rate = 0,3523. |
| P(2;5) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 5/13. |
| P(1;5) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 5/14. |
| P*(1;5) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ | Time-reversal of the puncturing mask P(1;5). |

**Table 4.5: GMR-1 puncturing masks for the rate 1/5 convolutional code (K = 5)**

| Identifier | Mask | Remark |
|---|---|---|
| P(2;3) | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 3/13. |
| P(5;3) | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 3/10. |
| P*(5;3) | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ | Time-reversal of the puncturing mask P(5;3). |

**Table 4.6: GMR-1 puncturing masks for the rate 1/2 convolutional code (K = 7)**

| Identifier | Mask | Remark |
|---|---|---|
| P(2;3) | $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 3/4. |
| P(4;10) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 5/8. |
| P(5;12) | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 12/19. |

| Identifier | Mask | Remark |
|---|---|---|
| P(1;16) | $\begin{bmatrix} 1 & 1 & ... & ... & 1 \\ 0 & 1 & ... & ... & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 16/31. |
| P(1;48) | $\begin{bmatrix} 1 & 1 & ... & ... & 1 \\ 0 & 1 & ... & ... & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 48/95. |
| P(1;84) | $\begin{bmatrix} 1 & 1 & ... & ... & 1 \\ 0 & 1 & ... & ... & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 84/167. |
| P(1;152) | $\begin{bmatrix} 1 & 1 & ... & ... & 1 \\ 0 & 1 & ... & ... & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 152/303. |

**Table 4.7: GMR-1 puncturing masks for the rate 1/2 convolutional code (K = 9)**

| Identifier | Mask | Remark |
|---|---|---|
| P(1;3) | $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 3/5. |
| P(4;7) | $\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 7/10. |
| P(3;4) | $\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ | Puncturing mask that, if applied repetitively, produces effective code rate = 4/5. |

# 4.6 Golay encoding

Same as clause 4.6 in GMR-1 05.003 [3].

# 4.7 Reed-Solomon encoding

Same as clause 4.7 in GMR-1 05.003 [3].

## 4.7.1 Encoder

Same as clause 4.7.1 in GMR-1 05.003 [3].

## 4.7.2 Galois field arithmetics

Same as clause 4.7.2 in GMR-1 05.003 [3].

## 4.7.3 Encoder feedback register operation

Same as clause 4.7.3 in GMR-1 05.003 [3].

# 4.8 Interleaving

Intraburst and interburst interleaving schemes are based on block interleaving methods with pseudorandom permutations and are channel dependent.

### 4.8.1 Intraburst interleaving

Intraburst interleaving is performed by mapping the block of the coded bits {c(0), c(1), …, c(M-1)} into a N × 8 matrix by rows, interchanging the columns using the pseudorandom permutation factor of 5, and reading out blocks of data by columns. Matrix dimension N is channel dependent and $N = \lceil M/8 \rceil$.

When columns are interchanged the index of the matrix element (i, j) changes to $(i, j_p)$, where $j_p = (j \times 5) \bmod 8$.

When the data is read out by columns, note there may be only N-1 elements in certain columns.

### 4.8.2 Interburst interleaving

Same as clause 4.8.2 in GMR-1 05.003 [3].

## 4.9 Scrambling

Same as clause 4.9 in GMR-1 05.003 [3].

# 5 Traffic channels

## 5.1 Traffic channel-3 (TCH3)

Same as clause 5.1 in GMR-1 05.003 [3].

### 5.1.1 Channel coding

Same as clause 5.1.1 in GMR-1 05.003 [3].

### 5.1.2 Interleaving

Same as clause 5.1.2 in GMR-1 05.003 [3].

### 5.1.3 Scrambling, multiplexing, and encryption

Same as clause 5.1.3 in GMR-1 05.003 [3].

## 5.2 Traffic channel-6 (TCH6)

### 5.2.1 Channel coding

#### 5.2.1.1 Coding for 2,4 kbps fax

Same as clause 5.2.1.1 in GMR-1 05.003 [3].

#### 5.2.1.2 Coding for 2,4 kbps data

Same as clause 5.2.1.2 in GMR-1 05.003 [3].

#### 5.2.1.3 Coding for 4,8 kbps fax/data

Same as clause 5.2.1.2 in GMR-1 05.003 [3].

### 5.2.2 Interleaving

Same as clause 5.2.2 in GMR-1 05.003 [3].

### 5.2.3 Scrambling, multiplexing, and encryption

Same as clause 5.2.3 in GMR-1 05.003 [3].

## 5.3 Traffic channel-9 (TCH9)

### 5.3.1 Channel coding

#### 5.3.1.1 Coding for 2,4 kbps fax

Same as clause 5.3.1.1 in GMR-1 05.003 [3].

#### 5.3.1.2 Coding for 4,8 kbps fax

Same as clause 5.3.1.2 in GMR-1 05.003 [3].

#### 5.3.1.3 Coding for 9,6 kbps fax/data

Same as clause 5.3.1.3 in GMR-1 05.003 [3].

### 5.3.2 Interleaving

Same as clause 5.3.2 in GMR-1 05.003 [3].

### 5.3.3 Scrambling, multiplexing, and encryption

Same as clause 5.3.3 in GMR-1 05.003 [3].

# 6 Control channels

## 6.1 Broadcast Control CHannel (BCCH)

Same as clause 6.1 in GMR-1 05.003 [3].

### 6.1.1 Channel coding

Same as clause 6.1.1 in GMR-1 05.003 [3].

### 6.1.2 Interleaving

Same as clause 6.1.2 in GMR-1 05.003 [3].

### 6.1.3 Scrambling and multiplexing

Same as clause 6.1.3 in GMR-1 05.003 [3].

## 6.2 Paging CHannel (PCH)

Same as clause 6.2 in GMR-1 05.003 [3].

### 6.2.1 Channel coding

Same as clause 6.2.1 in GMR-1 05.003 [3].

### 6.2.2 Interleaving

Same as clause 6.2.2 in GMR-1 05.003 [3].

### 6.2.3 Scrambling and multiplexing

Same as clause 6.2.3 in GMR-1 05.003 [3].

## 6.3 Access Grant CHannel (AGCH)

Same as clause 6.3 in GMR-1 05.003 [3].

### 6.3.1 Channel coding

Same as clause 6.3.1 in GMR-1 05.003 [3].

### 6.3.2 Interleaving

Same as clause 6.3.2 in GMR-1 05.003 [3].

## 6.4 Broadcast Alerting CHannel (BACH)

Same as clause 6.4 in GMR-1 05.003 [3].

### 6.4.1 Channel coding

Same as clause 6.4.1 in GMR-1 05.003 [3].

## 6.5 Random Access CHannel (RACH)

Same as clause 6.5 in GMR-1 05.003 [3].

### 6.5.1 Channel coding

Same as clause 6.5.1 in GMR-1 05.003 [3].

### 6.5.2 Interleaving

Same as clause 6.5.2 in GMR-1 05.003 [3].

### 6.5.3 Scrambling and multiplexing

Same as clause 6.5.3 in GMR-1 05.003 [3].

## 6.6 Cell Broadcast CHannel (CBCH)

Same as clause 6.6 in GMR-1 05.003 [3].

### 6.6.1 Channel coding

Same as clause 6.6.1 in GMR-1 05.003 [3].

### 6.6.2 Interleaving

Same as clause 6.6.2 in GMR-1 05.003 [3].

## 6.7 Standalone Dedicated Control CHannel (SDCCH)

Same as clause 6.7 in GMR-1 05.003 [3].

### 6.7.1 Channel coding

Same as clause 6.7.1 in GMR-1 05.003 [3].

### 6.7.2 Interleaving

Same as clause 6.7.2 in GMR-1 05.003 [3].

### 6.7.3 Scrambling, multiplexing, and encryption

Same as clause 6.7.3 in GMR-1 05.003 [3].

## 6.8 Slow Associated Control CHannel (SACCH)

Same as clause 6.8 in GMR-1 05.003 [3].

### 6.8.1 Channel coding

Same as clause 6.8.1 in GMR-1 05.003 [3].

### 6.8.2 Interleaving

Same as clause 6.8.2 in GMR-1 05.003 [3].

## 6.9 Fast Associated Control CHannel-3 (FACCH3)

Same as clause 6.9 in GMR-1 05.003 [3].

### 6.9.1 Channel coding

Same as clause 6.9.1 in GMR-1 05.003 [3].

### 6.9.2 Interleaving

Same as clause 6.9.2 in GMR-1 05.003 [3].

### 6.9.3 Scrambling, multiplexing, and encryption

Same as clause 6.9.3 in GMR-1 05.003 [3].

## 6.10 Fast Associated Control CHannel-6 (FACCH6)

Same as clause 6.10 in GMR-1 05.003 [3].

### 6.10.1 Channel coding

Same as clause 6.10.1 in GMR-1 05.003 [3].

### 6.10.2 Interleaving

Same as clause 6.10.2 in GMR-1 05.003 [3].

### 6.10.3 Scrambling, multiplexing, and encryption

Same as clause 6.10.3 in GMR-1 05.003 [3].

## 6.11 Fast Associated Control CHannel-9 (FACCH9)

Same as clause 6.11 in GMR-1 05.003 [3].

### 6.11.1 Channel coding

Same as clause 6.11.1 in GMR-1 05.003 [3].

### 6.11.2 Interleaving

Same as clause 6.11.2 in GMR-1 05.003 [3].

### 6.11.3 Scrambling, multiplexing, and encryption

Same as clause 6.11.3 in GMR-1 05.003 [3].

## 6.12 Terminal-to-terminal Associated Control CHannel (TACCH)

Same as clause 6.12 in GMR-1 05.003 [3].

### 6.12.1 TACCH channel coding

Same as clause 6.12.1 in GMR-1 05.003 [3].

### 6.12.2 Interleaving

Same as clause 6.12.2 in GMR-1 05.003 [3].

### 6.12.3 Scrambling and multiplexing

Same as clause 6.12.3 in GMR-1 05.003 [3].

### 6.12.4 PHYsical (PHY) header for TACCH

Same as clause 6.12.4 in GMR-1 05.003 [3].

## 6.13 GPS Broadcast CHannel (GBCH)

Same as clause 6.13 in GMR-1 05.003 [3].

### 6.13.1 Channel coding

Same as clause 6.13.1 in GMR-1 05.003 [3].

### 6.13.2 Interleaving

Same as clause 6.13.2 in GMR-1 05.003 [3].

### 6.13.3 Scrambling and multiplexing

Same as clause 6.13.3 in GMR-1 05.003 [3].

# 7 Logical channel multiplexing

## 7.1 SACCH multiplexing

Same as clause 7.1 in GMR-1 05.003 [3].

## 7.2 Status field

Same as clause 7.2 in GMR-1 05.003 [3].

### 7.2.1 Power control field

Same as clause 7.2.1 in GMR-1 05.003 [3].

### 7.2.2 Comfort noise field

Same as clause 7.2.2 in GMR-1 05.003 [3] except deleting the last sentence.

## 7.3 Status field with NTN bursts

### 7.3.1 Status field with NT6 and NT9 bursts

Same as clause 7.3.1 in GMR-1 05.003 [3].

### 7.3.2 Status field with NT3 bursts

#### 7.3.2.1 Status field with NT3 bursts for encoded speech

Same as clause 7.3.2.1 in GMR-1 05.003 [3].

#### 7.3.2.2 Status field with NT3 bursts for FACCH

Same as clause 7.3.2.2 in GMR-1 05.003 [3].

### 7.3.3 Status field with Keep-Alive Bursts (KAB)

Same as clause 7.3.3 in GMR-1 05.003 [3].

# 8 Encryption

Same as clause 8 in GMR-1 05.003 [3].

# 9 Packet Switched Channels

## 9.1 Packet Data Traffic Channels

Several types of bursts are used by the various logical channels in the downlink and uplink Packet Data CHannels (PDCHs). The Packet Normal Bursts, PNB(m,n), are characterized by their transmission rate in multiples, m, of the basic transmission rate of 23,4 ksps, and their duration in multiples, n, of the timeslot of 40/24 ms. For example, PNB(4,3) is a burst transmitted at m = 4 times the basic rate (93,6 ksps transmission rate occupying a nominal bandwidth of 125 kHz) with a nominal duration n = 3 timeslots, or 5 ms.

Bursts transmitted over the downlink and uplink PDCHs consist of two parts:

- PUblic Information (PUI);
- PRivate Information (PRI).

PRI is also termed payload. The PRI comprises N message bits $d_0$ to $d_{N-1,}$ as defined in clause 10.1 in GMPRS-1 04.060 [5]. PUI is also called burst header, or physical layer header (hereafter referred to as PUI). The coding for the PUI and the various burst formats are described in the following clauses.

### 9.1.1 PUblic Information (PUI)

For all packet data bursts, the PUI comprises 12 information bits $b_0$ to $b_{11}$, as defined in GMPRS-1 04.060 [5]. These 12 bits are encoded via the extended Golay (24,12) code as described in clause 4.6, where u(i) = $b_i$, i = 0,…, 11. The output from the Golay encoder is a block of 24 bits, {c(0),c(1), …, c(23)}. After Golay encoding, each encoded bit is repeated once to form the 48-bit coded PUI as {c(0), c(1), …, c(23), c(0), c(1), …, c(23)}. Note there are no interleaving, scrambling and encryption on PUI data.

### 9.1.2 Void

### 9.1.3 Packet Normal Burst PNB(4,3)

The information rate supported by the PNB(4,3) depends on the channel coding used. Channel coding rates of approximately 1/2, 5/8, and 3/4 are defined for the PNB(4,3). Independent of channel coding rate, the PNB(4,3) provides a common payload for 792 coded bits.

### 9.1.3.1 Rate 1/2 convolutional coding

A 16-bit CRC is applied to the 376 message bits $d_0$ to $d_{375}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 375. An 8-bit masking function is applied to the block of 392 CRC-protected bits {u(0),…,u(391)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(391)}, where u´(i) = u(i), i = 0,…, 383 and u´(i + 384) = u(i + 384) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 392 bits {u´(0),…, u´(391)} is delivered to the encoder.

The 392 bits {u´(0),…, u´(391)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 800 coded bits {b(0),…, b(799)}.

Puncturing is performed using the puncture mask P(1;48), as listed in table 4.6. After that, one bit in {b} (with index 769) needs to be depunctured. This action turns out that 8 bits in {b} with index 96k + 1 (k = 0,…, 7) are punctured out. The result is a block of 792 coded bits {c(0),…, c(791)}.

### 9.1.3.2 Rate 5/8 convolutional coding

A 16-bit CRC is applied to the 472 message bits $d_0$ to $d_{471}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 471. An 8-bit masking function is applied to the block of 488 CRC-protected bits {u(0),…,u(487)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(487)}, where u´(i) = u(i), i = 0, …, 479 and u´(i + 480) = u(i + 480) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 488 bits {u´(0),…, u´(487)} is delivered to the encoder.

The 488 bits {u´(0),…, u´(487)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 992 coded bits {b(0),…, b(991)}.

Puncturing is performed using the puncture mask P(4;10) as listed in table 4.6. Besides, one more bit in {b} (with index 991) needs to be punctured. This action turns out that 200 bits in {b} with index 20k + 1, 20k + 5, 20k + 9 (k = 0,…, 49), 20k + 17 (k = 0,…, 48) and 991 are punctured out. The result is a block of 792 coded bits {c(0),…, c(791)}.

### 9.1.3.3 Rate 3/4 convolutional coding

A 16-bit CRC is applied to the 568 message bits $d_0$ to $d_{567}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 567. An 8-bit masking function is applied to the block of 584 CRC-protected bits {u(0),…, u(583)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(583)}, where u´(i) = u(i), i = 0,…, 575 and u´(i + 576) = u(i + 576) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 584 bits {u´(0),…, u´(583)} is delivered to the encoder.

The 584 bits {u´(0),…, u´(583)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 1 184 coded bits {b(0),…, b(1 183)}.

Puncturing is performed using the puncture mask P(2;3) as listed in table 4.6. After that, two bits in {b}with index 1 179 and 1 180 are depunctured. This action turns out that 392 bits in {b} with index 6k + 3 and 6k + 4 (k = 0,…, 195) are punctured out. The result is a block of 792 coded bits {c(0),…, c(791)}.

#### 9.1.3.4 Interleaving

Intraburst interleaving is performed as described in clause 4.8.1.

#### 9.1.3.5 Scrambling, multiplexing, and encryption

Scrambling is performed as described in clause 4.9. No encryption is performed.

### 9.1.4 Packet Normal Burst PNB(5,3)

The information rate supported by the PNB(5,3) depends on the channel coding used. Channel coding rates of approximately 1/2, 5/8, and 3/4 are defined for the PNB(5,3). Independent of channel coding rate, the PNB(5,3) provides a common payload for 1 002 coded bits.

#### 9.1.4.1 Rate 1/2 convolutional coding

A 16-bit CRC is applied to the 480 message bits $d_0$ to $d_{479}$ as specified in clause 4.3, where $d(i) = d_i$, i = 0, …, 479. An 8-bit masking function is applied to the block of 496 CRC-protected bits {u(0),…, u(495)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(495)}, where u´(i) = u(i), i = 0, …, 487 and u´(i + 488) = u(i + 488) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 496 bits {u´(0),…, u´(495)} is delivered to the encoder.

The 496 bits {u´(0),…, u´(495)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 1 008 coded bits {b(0),…, b(1 007)}.

Puncturing is performed using the puncture mask P(1;84) as listed in table 4.6. This action turns out that 6 bits in {b} with index 168k + 1 (k = 0,…, 5) are punctured out. The result is a block of 1 002 coded bits {c(0),…, c(1 001)}.

#### 9.1.4.2 Rate 5/8 convolutional coding

A 16-bit CRC is applied to the 608 message bits $d_0$ to $d_{607}$ as specified in clause 4.3, where $d(i) = d_i$, i = 0, …, 607. An 8-bit masking function is applied to the block of 624 CRC-protected bits {u(0),…,u(623)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(623)}, where u´(i) = u(i), i = 0,…, 615 and u´(i + 616) = u(i + 616) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 624 bits {u´(0),…, u´(623)} is delivered to the encoder.

The 624 bits {u´(0),…, u´(623)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 1 264 coded bits {b(0),…, b(1 263)}.

Puncturing is performed using the puncture mask P(5;12) as listed in table 4.6. After that, one bit in {b} with index 1 263 needs to be depunctured. This action turns out that 262 bits in {b} with index 24k + 3, 24k + 7 (k = 0,…, 52) and 24k + 15, 24k + 19, 24k + 23 (k = 0,…, 51) are punctured out. The result is a block of 1 002 coded bits {c(0),…, c(1 001)}.

#### 9.1.4.3 Rate 3/4 convolutional coding

A 16-bit CRC is applied to the 728 message bits $d_0$ to $d_{727}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 727. An 8-bit masking function is applied to the block of 744 CRC-protected bits {u(0),…, u(743)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(743)}, where u´(i) = u(i), i = 0,…, 735 and u´(i + 736) = u(i + 736) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 744 bits {u´(0),…, u´(743)} is delivered to the encoder.

The 744 bits {u´(0),…, u´(743)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 1 504 coded bits {b(0),…, b(1 503)}.

Puncturing is performed using the puncture mask P(2;3) as listed in table 4.6. Besides, one more bit in {b} with index 1 495 is punctured. This action turns out that 502 bits in {b} with index 6k + 3 (k = 0,…, 250), 6k + 4 (k = 0,…, 249) and 1 495 are punctured out. The result is a block of 1 002 coded bits {c(0),…, c(1 001)}.

#### 9.1.4.4 Interleaving

Intraburst interleaving is performed as described in clause 4.8.1.

#### 9.1.4.5 Scrambling, multiplexing, and encryption

Scrambling is performed as described in clause 4.9. No encryption is performed.

## 9.1.5 Void

## 9.1.6 Packet Normal Burst PNB(1,6)

The information rate supported by the PNB(1,6) depends on the channel coding used. Channel coding rates of approximately 6/10, 7/10, and 4/5 are defined for the PNB(1,6). Independent of channel coding rate, the PNB(1,6) provides a common payload for 366 coded bits.

#### 9.1.6.1 Rate 3/5 convolutional coding

A 16-bit CRC is applied to the 192 message bits $d_0$ to $d_{191}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 191. An 8-bit masking function is applied to the block of 208 CRC-protected bits {u(0),…,u(207)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(207)}, where u´(i) = u(i), i = 0,…, 199 and u´(i + 200) = u(i + 200) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 208 bits {u´(0),…, u´(207)} is delivered to the encoder.

The 208 bits {u´(0),…, u´(207)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 432 coded bits {b(0),…, b(431)}.

Puncturing is performed using the puncture mask P(1;3), as listed in table 4.7. This action turns out that 72 bits in {b} with index 6k + 1 (k = 0,…, 71) are punctured out. After applying P(1:3), these bits will be depunctured: 37, 109, 181, 253, 325, 397. The result is a block of 366 coded bits {c(0),…,c(365)}.

### 9.1.6.2 Rate 7/10 convolutional coding

A 16-bit CRC is applied to the 232 message bits $d_0$ to $d_{231}$as specified in clause 4.3, where $d(i) = d_i$, i = 0, …, 231. An 8-bit masking function is applied to the block of 248 CRC-protected bits {u(0),…, u(247)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(247)}, where u´(i) = u(i), i = 0, …, 239 and u´(i + 239) = u(i + 239) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 248 bits {u´(0),…, u´(247)} is delivered to the encoder.

The 248 bits {u´(0),…, u´(247)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 512 coded bits {b(0),…, b(511)}.

Puncturing is performed using the puncture mask P(4;7) as listed in table 4.7. This action turns out that 146 bits in {b} with index 14k , 14k + 5, (k = 0,…,36) and 14k + 9, 14k + 13 (k = 0,…, 35) are punctured out. The result is a block of 366 coded bits {c(0),…,c(365)}.

### 9.1.6.3 Rate 4/5 convolutional coding

A 16-bit CRC is applied to the 272 message bits $d_0$ to $d_{271}$ as specified in clause 4.3, where $d(i) = d_i$, i = 0, …, 271. An 8-bit masking function is applied to the block of 288 CRC-protected bits {u(0),…, u(287)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(287)}, where u´(i) = u(i), i = 0,…, 279 and u´(i + 280) = u(i + 270) ⊕ m(i), i = 0,…,7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 288 bits {u´(0),…, u´(287)} is delivered to the encoder.

The 288 bits {u´(0),…, u´(287)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 592 coded bits {b(0),…, b(591)}.

Puncturing is performed using the puncture mask P(3;4) as listed in table 4.7. This action turns out that 216 bits in {b} with index 8k + 3, 8k + 4, 8k + 7 (k = 0,…, 73) are punctured out. After applying P(3:4), these bits also will be punctured: 0 , 8, 576, 584. The result is a block of 366 coded bits {c(0),…, c(365)}.

### 9.1.6.4 Interleaving

Intraburst interleaving is performed as described in clause 4.8.1.

### 9.1.6.5 Scrambling, multiplexing, and encryption

Scrambling is performed as described in clause 4.9. No encryption is performed.

## 9.1.7 Packet Normal Burst PNB(2,6)

The information rate supported by the PNB(2,6) depends on the channel coding used. Channel coding rates of approximately 3/5, 7/10, and 4/5 are defined for the PNB(2,6). Independent of channel coding rate, the PNB(2,6) provides a common payload for 810 coded bits.

### 9.1.7.1 Rate 3/5 convolutional coding

A 16-bit CRC is applied to the 456 message bits $d_0$ to $d_{455}$ as specified in clause 4.3, where $d(i) = d_i$, i = 0, …, 455. An 8-bit masking function is applied to the block of 472 CRC-protected bits {u(0),…, u(471)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(471)}, where u´(i) = u(i), i = 0,…, 463 and u´(i + 464) = u(i + 464) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 472 bits {u´(0),…, u´(471)} is delivered to the encoder.

The 472 bits {u´(0),…, u´(471)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 960 coded bits {b(0),…, b(959)}.

Puncturing is performed using the puncture mask P(1;3), as listed in table 4.7. This action turns out that 160 bits in {b} with index 6k + 1 (k = 0,…,159) are punctured out. After applying P(1:3) these bits will be depunctured: 73, 163, 253, 343, 433, 523 , 613, 703, 793, 883. The result is a block of 810 coded bits {c(0),…, c(809)}.

### 9.1.7.2 Rate 7/10 convolutional coding

A 16-bit CRC is applied to the 544 message bits $d_0$ to $d_{543}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 543. An 8-bit masking function is applied to the block of 560 CRC-protected bits {u(0),…,u(559)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(559)}, where u´(i) = u(i), i = 0, …, 551 and u´(i + 552) = u(i + 552) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 560 bits {u´(0),…, u´(559)} is delivered to the encoder.

The 560 bits {u´(0),…, u´(559)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 1 136 coded bits {b(0),…, b(1 135)}.

Puncturing is performed using the puncture mask P(4;7) as listed in table 4.7. This action turns out that 325 bits in {b} with index 14k , (k = 0,…, 81) and 14k + 5, 14k + 9, 14k + 13 (k = 0,…, 80). After applying P(4;7), bit 2 will be punctured out. The result is a block of 810 coded bits {c(0),…, c(809)}.

### 9.1.7.3 Rate 4/5 convolutional coding

A 16-bit CRC is applied to the 624 message bits $d_0$ to $d_{623}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 623. An 8-bit masking function is applied to the block of 640 CRC-protected bits {u(0),…, u(639)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 16 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(639)}, where u´(i) = u(i), i = 0,…, 631 and u´(i + 632) = u(i + 632) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 640 bits {u´(0),…, u´(639)} is delivered to the encoder.

The 640 bits {u´(0),…, u´(639)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 9 is used. Eight tail bits are added. The encoding results in a block of 1 296 coded bits {b(0),…, b(1 295)}.

Puncturing is performed using the puncture mask P(3;4) as listed in table 4.7. This action turns out that 486 bits in {b} with index 8k + 3, 8k + 4, 8k + 7 (k = 0,…, 161) are punctured out. The result is a block of 810 coded bits {c(0),…, c(809)}.

### 9.1.7.4 Interleaving

Intraburst interleaving is performed as described in clause 4.8.1.

### 9.1.7.5 Scrambling, multiplexing, and encryption

Scrambling is performed as described in clause 4.9. No encryption is performed.

# 9.2 Packet Access Burst (PAB)

## 9.2.1 Channel coding

An 8-bit CRC is applied to 64 message bits $d_0$ to $d_{63}$ as specified in clause 4.3, where d(i) = $d_i$, i = 0, …, 63. An 8-bit masking function is applied to the block of 72 CRC-protected bits {u(0),…,u(71)} in the following manner. The mask, {m(0), m(1), …, m(7)}, is the string RACH_SB_Mask broadcast in system information as specified in GMPRS-1 04.008 [4]. Here m(0) is the MSB bit and m(7) is the LSB bit of the mask. The 8-bit mask is applied to 8 CRC bits of the message with an XOR operation to give us {u´(0),…, u´(71)}, where u´(i) = u(i), i = 0,…, 63 and u´(i + 64) = u(i + 64) ⊕ m(i), i = 0,…, 7, where ⊕ denotes modulo 2 addition (XOR). The resultant block of 72 bits {u´(0),…, u´(71)} is delivered to the encoder.

The 72 bits {u´(0),…, u´(71)} are encoded via the rate 1/2 convolutional code. The code is defined by the generator polynomial specified in clause 4.4.2, where the constraint length K = 7 is used. Eight tail bits are used in order to accommodate K = 9 in the future for the possible enhancement. The encoding results in a block of 160 coded bits {b(0),…, b(159)}.

Puncturing is performed using mask P(2;3) shown in table 4.6. Besides, one more bit in {b} (with index 0) needs to be punctured. This turns out that 54 bits in {b} with index 6k + 3 (k = 0,…, 26), 6k + 4 (k = 0,…, 25) and 0 are punctured. The result is a block of 106 coded bits {c(0),…, c(105)}. The coding rate after puncturing is approximately 3/4.

## 9.2.2 Interleaving

Intraburst interleaving is performed as described in clause 4.8.1.

## 9.2.3 Scrambling and multiplexing

No scrambling is performed.

# Annex A (informative): Bibliography

GMPRS-1 05.002 (ETSI TS 101 376-5-2): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service; Part 5: Radio interface physical layer specifications; Sub-part 2: Multiplexing and Multiple Access; Stage 2 Service Description; GMPRS-1 05.002".

GMPRS-1 05.004 (ETSI TS 101 376-5-4): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service Part 5: Radio interface physical layer specifications; Sub-part 4: Modulation; GMPRS-1 05.004".

GMPRS-1 05.008 (ETSI TS 101 376-5-6): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service Part 5: Radio interface physical layer specifications; Sub-part 6: Radio Subsystem Link Control; GMPRS-1 05.008".

GMPRS-1 05.010 (ETSI TS 101 376-5-7): "GEO-Mobile Radio Interface Specifications (Release 2); General Packet Radio Service Part 5: Radio interface physical layer specifications; Sub-part 7: Radio Subsystem Synchronization; GMPRS-1 05.010".

# History

| Document history | | |
|---|---|---|
| V2.1.1 | March 2003 | Publication |
| V2.2.1 | March 2005 | Publication |
| | | |
| | | |
| | | |