ETSI TS 125 222 V3.1.1 (2000-01)

Technical Specification

Universal Mobile Telecommunications System (UMTS); Multiplexing and channel coding (TDD) (3G TS 25.222 version 3.1.1 Release 1999)

Reference

DTS/TSGR-0125222U

Keywords

UMTS

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - 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

Internet

secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

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 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://www.etsi.org/ipr).

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 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 the ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables. The mapping of document identities is as follows:

For 3GPP documents:

3G TS | TR nn.nnn "<title>" (with or without the prefix 3G)

is equivalent to

ETSI TS | TR 1nn nnn "[Digital cellular telecommunications system (Phase 2+) (GSM);] Universal Mobile Telecommunications System; <title>

For GSM document identities of type "GSM xx.yy", e.g. GSM 01.04, the corresponding ETSI document identity may be found in the Cross Reference List on www.etsi.org/key

Contents

Foreword ... 5
1 Scope ... 6
2 References ... 6
3 Definitions, symbols and abbreviations ... 6
3.1 Definitions ... 6
3.2 Symbols ... 6
3.3 Abbreviations ... 7
4 Multiplexing, channel coding and interleaving ... 8
4.1 General ... 8
4.2 Transport channel coding/multiplexing ... 9
4.2.1 Error detection ... 11
4.2.1.1 CRC calculation ... 11
4.2.1.2 Relation between input and output of the Cyclic Redundancy Check ... 11
4.2.2 Transport block concatenation and code block segmentation ... 12
4.2.2.1 Concatenation of transport blocks ... 12
4.2.2.2 Code block segmentation ... 12
4.2.3 Channel coding ... 13
4.2.3.1 Convolutional Coding ... 13
4.2.3.2 Turbo coding ... 14
4.2.3.2.1 Turbo coder ... 14
4.2.3.2.2 Trellis termination in turbo code ... 15
4.2.3.2.3 Turbo code internal interleaver ... 15
4.2.4 Radio frame size equalisation ... 18
4.2.5 1st interleaving ... 18
4.2.6 Radio frame segmentation ... 19
4.2.7 Rate matching ... 20
4.2.7.1 Determination of rate matching parameters ... 21
4.2.7.1.1 Uncoded and convolutionally encoded TrCHs ... 21
4.2.7.1.2 Turbo encoded TrCHs ... 22
4.2.7.2 Bit separation and collection for rate matching ... 23
4.2.7.2.1 Bit separation ... 24
4.2.7.2.2 Bit collection ... 25
4.2.7.3 Rate matching pattern determination ... 25
4.2.8 TrCH multiplexing ... 26
4.2.9 Physical channel segmentation ... 27
4.2.10 2nd interleaving ... 27
4.2.10.1 Frame related 2nd interleaving ... 27
4.2.10.2 Timeslot related 2nd interleaving ... 28
4.2.11 Physical channel mapping ... 29
4.2.11.1 Mapping scheme after frame related 2nd interleaving ... 29
4.2.11.1.1 Mapping scheme after frame related 2nd interleaving in uplink ... 29
4.2.11.1.2 Mapping scheme after frame related 2nd interleaving in downlink ... 29
4.2.11.2 Mapping scheme after timeslot related 2nd interleaving ... 30
6.2.11.2.1 Mapping scheme after timeslot related 2nd interleaving in uplink ... 30
6.2.11.2.2 Mapping scheme after timeslot related 2nd interleaving in downlink ... 30
4.2.12 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels ... 30
4.2.12.1 Allowed CCTrCH combinations for one UE ... 31
4.2.12.1.1 Allowed CCTrCH combinations on the uplink ... 31
4.2.12.1.2 Allowed CCTrCH combinations on the downlink ... 31
4.2.13 Transport format detection ... 31
4.2.13.1 Blind transport format detection ... 32
4.2.13.2 Explicit transport format detection based on TFCI ... 32
4.2.13.2.1 Transport Format Combination Indicator (TFCI) ... 32
4.3 Coding for layer 1 control ... 32

4.3.1 Coding of transport format combination indicator (TFCI) 32
4.3.1.1 Coding of long TFCI lengths 32
4.3.1.2 Coding of short TFCI lengths 34
4.3.1.2.1 Coding very short TFCIs by repetition 34
4.3.1.2.2 Coding short TFCIs using bi-orthogonal codes 34
4.3.1.3 Mapping of TFCI word 35
4.3.2 Coding of Paging Indicator (PI) 35
4.3.3 Coding of Transmit Power Control (TPC) 35

Annex A (informative): Change history 36

History 37

Foreword

This Technical Specification has been produced by the 3GPP.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version 3.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 Indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the specification;

1 Scope

The present document describes multiplexing, channel coding and interleaving for UTRA Physical Layer TDD mode.

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, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] 3G TS 25.202: "UE capabilities"

[2] 3G TS 25.211: "Transport channels and physical channels (FDD)"

[3] 3G TS 25.212: "Multiplexing and channel coding (FDD)"

[4] 3G TS 25.213: "Spreading and modulation (FDD)"

[5] 3G TS 25.214: "Physical layer procedures (FDD)"

[6] 3G TS 25.215: "Physical layer – Measurements (FDD)"

[7] 3G TS 25.221: "Transport channels and physical channels (TDD)"

[9] 3G TS 25.223: "Spreading and modulation (TDD)"

[10] 3G TS 25.224: "Physical layer procedures (TDD)"

[11] 3G TS 25.225: "Measurements"

[12] 3G TS S2.01: "Radio Interface Protocol Architecture"

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

TrCH number: Transport channel number represents a TrCH ID assigned to L1 by L2. Transport channels are multiplexed to the CCTrCH in the ascending order of these IDs.

3.2 Symbols

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

$\lceil x \rceil$ round towards ∞, i.e. integer such that $x \leq \lceil x \rceil < x+1$

$\lfloor x \rfloor$ round towards -∞, i.e. integer such that $x-1 < \lfloor x \rfloor \leq x$

$|x|$ absolute value of x

Unless otherwise is explicitly stated when the symbol is used, the meaning of the following symbols are:

i	TrCH number
j	TFC number
k	Bit number
l	TF number
m	Transport block number
n	Radio frame number
p	PhCH number
r	Code block number
I	Number of TrCHs in a CCTrCH.
C_i	Number of code blocks in one TTI of TrCH i.
F_i	Number of radio frames in one TTI of TrCH i.
M_i	Number of transport blocks in one TTI of TrCH i.
P	Number of PhCHs used for one CCTrCH.
PL	Puncturing Limit for the uplink. Signalled from higher layers
RM_i	Rate Matching attribute for TrCH i. Signalled from higher layers.

Temporary variables, i.e. variables used in several (sub)sections with different meaning.

x, X
y, Y
z, Z

3.3 Abbreviations

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

<ACRONYM>	<Explanation>
ARQ	Automatic Repeat on Request
BCH	Broadcast Channel
BER	Bit Error Rate
BS	Base Station
BSS	Base Station Subsystem
CBR	Constant Bit Rate
CCCH	Common Control Channel
CCTrCH	Coded Composite Transport Channel
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
DCA	Dynamic Channel Allocation
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Control
FER	Frame Error Rate
GF	Galois Field
JD	Joint Detection
L1	Layer 1
L2	Layer 2
LLC	Logical Link Control
MA	Multiple Access
MAC	Medium Access Control
MS	Mobile Station
MT	Mobile Terminated
NRT	Non-Real Time

OVSF	Orthogonal Variable Spreading Factor
PC	Power Control
PCCC	Parallel Concatenated Convolutional Code
PCH	Paging Channel
PhCH	Physical Channel
PI	Paging Indicator
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RACH	Random Access Channel
RF	Radio Frequency
RLC	Radio Link Control
RRC	Radio Resource Control
RRM	Radio Resource Management
RSC	Recursive Systematic Convolutional Coder
RT	Real Time
RU	Resource Unit
SCCC	Serial Concatenated Convolutional Code
SCH	Synchronization Channel
SNR	Signal to Noise Ratio
TCH	Traffic channel
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TPC	Transmit Power Control
TrBk	Transport Block
TrCH	Transport Channel
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
USCH	Uplink Shared Channel
UTRA	UMTS Terrestrial Radio Access
VBR	Variable Bit Rate

4 Multiplexing, channel coding and interleaving

4.1 General

Data stream from/to MAC and higher layers (Transport block / Transport block set) is encoded/decoded to offer transport services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting (including rate matching), and interleaving and transport channels mapping onto/splitting from physical channels.

In the UTRA-TDD mode, the total number of basic physical channels (a certain time slot one spreading code on a certain carrier frequency) per frame is given by the maximum number of time slots which is 15 and the maximum number of CDMA codes per time slot.

4.2 Transport channel coding/multiplexing

Figure 4-1 illustrates the overall concept of transport-channel coding and multiplexing. Data arrives to the coding/multiplexing unit in form of transport block sets, once every transmission time interval. The transmission time interval is transport-channel specific from the set {10 ms, 20 ms, 40 ms, 80 ms}.

The following coding/multiplexing steps can be identified:

- Add CRC to each transport block (see section 4.2.1)
- TrBk concatenation / Code block segmentation (see section 4.2.2)
- Channel coding (see section 4.2.3)
- Radio frame size equalization (see section 4.2.4)
- Interleaving (two steps, see sections 4.2.5 and 4.2.10)
- Radio frame segmentation (4.2.6)
- Rate matching (see section 4.2.7)
- Multiplexing of transport channels (see section 4.2.8)
- Physical channel segmentation (see section 4.2.9)
- Mapping to physical channels (see section 4.2.11)

The coding/multiplexing steps for uplink and downlink are shown in figure 4-1.

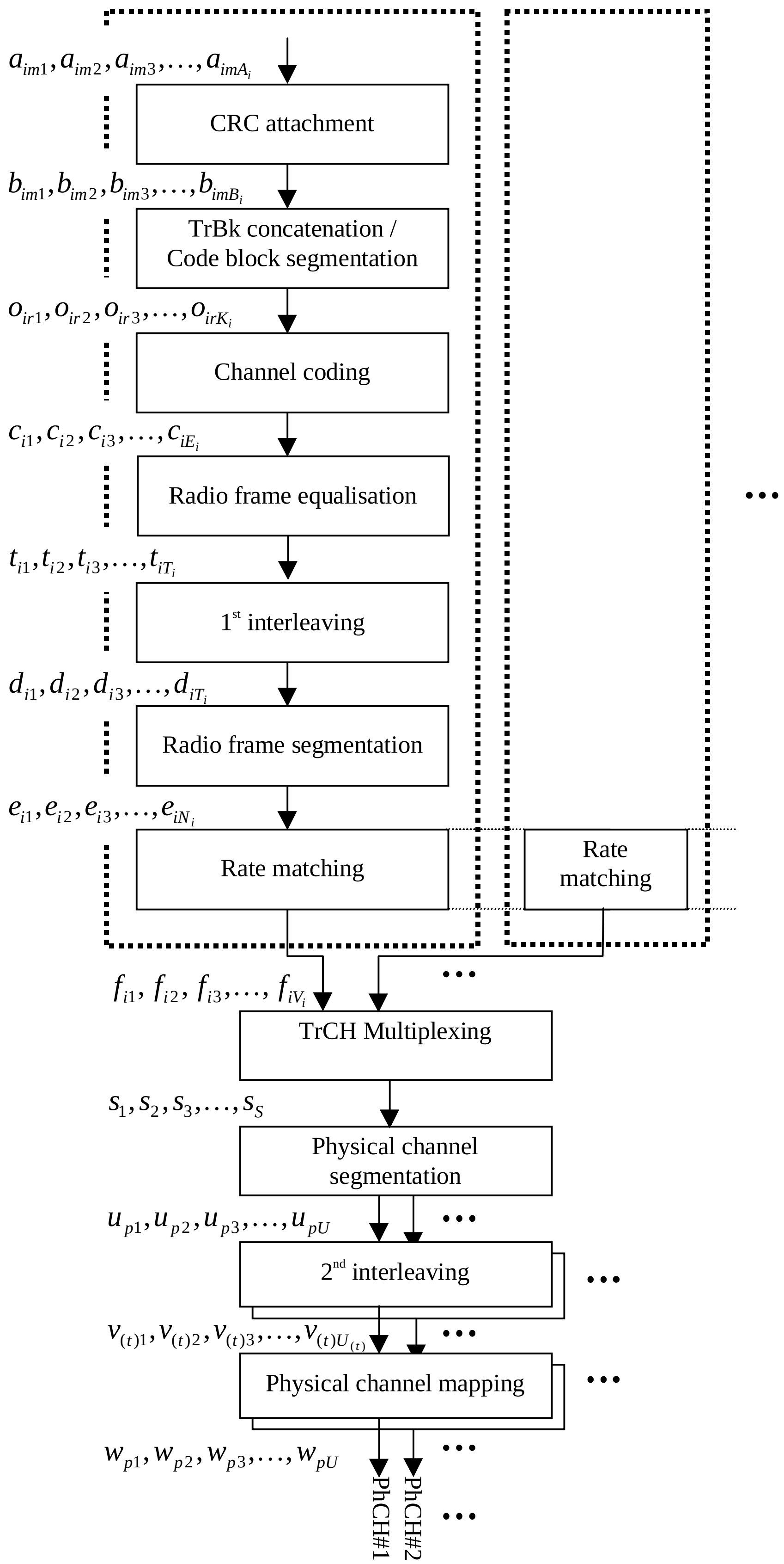

Figure 4–1: Transport channel multiplexing structure for uplink and downlink

Primarily, transport channels are multiplexed as described above, i.e. into one data stream mapped on one or several physical channels. However, an alternative way of multiplexing services is to use multiple CCTrCHs (Coded Composite Transport Channels), which corresponds to having several parallel multiplexing chains as in figure 4-1, resulting in several data streams, each mapped to one or several physical channels.

4.2.1 Error detection

Error detection is provided on transport blocks through a Cyclic Redundancy Check. The CRC is 24, 16, 12, 8 or 0 bits and it is signalled from higher layers what CRC length that should be used for each transport channel.

4.2.1.1 CRC calculation

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

$g_{CRC24}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$

$g_{CRC16}(D) = D^{16} + D^{12} + D^5 + 1$

$g_{CRC12}(D) = D^{12} + D^{11} + D^3 + D^2 + D + 1$

$g_{CRC8}(D) = D^8 + D^7 + D^4 + D^3 + D + 1$

Denote the bits in a transport block delivered to layer 1 by $a_{im1}, a_{im2}, a_{im3}, \ldots, a_{imA_i}$, and the parity bits by $p_{im1}, p_{im2}, p_{im3}, \ldots, p_{imL_i}$. A_i is the length of a transport block of TrCH i, m is the transport block number, and L_i is 24, 16, 8, or 0 depending on what is signalled from higher layers.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial

$$a_{im1}D^{A_i+23} + a_{im2}D^{A_i+22} + \ldots + a_{imA_i}D^{24} + p_{im1}D^{23} + p_{im2}D^{22} + \ldots + p_{im23}D^1 + p_{im24}$$

yields a remainder equal to 0 when divided by $g_{CRC24}(D)$, polynomial

$$a_{im1}D^{A_i+15} + a_{im2}D^{A_i+14} + \ldots + a_{imA_i}D^{16} + p_{im1}D^{15} + p_{im2}D^{14} + \ldots + p_{im15}D^1 + p_{im16}$$

yields a remainder equal to 0 when divided by $g_{CRC16}(D)$, polynomial

$$a_{im1}D^{A_i+11} + a_{im2}D^{A_i+10} + \ldots + a_{imA_i}D^{12} + p_{im1}D^{11} + p_{im2}D^{10} + \ldots + p_{im7}D^1 + p_{im12}$$

yields a remainder equal to 0 when divided by $g_{CRC12}(D)$ and the polynomial

$$a_{im1}D^{A_i+7} + a_{im2}D^{A_i+6} + \ldots + a_{imA_i}D^{8} + p_{im1}D^{7} + p_{im2}D^{6} + \ldots + p_{im7}D^1 + p_{im8}$$

yields a remainder equal to 0 when divided by $g_{CRC8}(D)$.

4.2.1.2 Relation between input and output of the Cyclic Redundancy Check

The bits after CRC attachment are denoted by $b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$, where $B_i=A_i+L_i$. The relation between a_{imk} and b_{imk} is:

$$b_{imk} = a_{imk} \qquad k = 1, 2, 3, \ldots, A_i$$

$$b_{imk} = p_{im(L_i+1-(k-A_i))} \qquad k = A_i + 1, A_i + 2, A_i + 3, \ldots, A_i + L_i$$

4.2.2 Transport block concatenation and code block segmentation

All transport blocks in a TTI are serially concatenated. If the number of bits in a TTI is larger than the maximum size of a code block , then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional, turbo coding or no coding is used for the TrCH.

4.2.2.1 Concatenation of transport blocks

The bits input to the transport block concatenation are denoted by $b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$ where i is the TrCH number, m is the transport block number, and B_i is the number of bits in each block (including CRC). The number of transport blocks on TrCH i is denoted by M_i. The bits after concatenation are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where i is the TrCH number and $X_i=M_iB_i$. They are defined by the following relations:

$$x_{ik} = b_{i1k} \quad k = 1, 2, \ldots, B_i$$

$$x_{ik} = b_{i,2,(k-B_i)} \quad k = B_i + 1, B_i + 2, \ldots, 2B_i$$

$$x_{ik} = b_{i,3,(k-2B_i)} \quad k = 2B_i + 1, 2B_i + 2, \ldots, 3B_i$$

$\ldots$

$$x_{ik} = b_{i,M_i,(k-(M_i-1)B_i)} \quad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \ldots, M_iB_i$$

4.2.2.2 Code block segmentation

NOTE: It is assumed that filler bits are set to 0.

Segmentation of the bit sequence from transport block concatenation is performed if $X_i>Z$. The code blocks after segmentation are of the same size. The number of code blocks on TrCH i is denoted by C_i. If the number of bits input to the segmentation, X_i, is not a multiple of C_i, filler bits are added to the last block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

convolutional coding: $Z = 504$

turbo coding: $Z = 5114$

no channel coding: Z = *unlimited*

The bits output from code block segmentation are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits.

Number of code blocks: $C_i = \lceil X_i / Z \rceil$

Number of bits in each code block: $K_i = \lceil X_i / C_i \rceil$

Number of filler bits: $Y_i = C_iK_i - X_i$

If $X_i \le Z$, then $o_{i1k} = x_{ik}$, and $K_i = X_i$.

If $X_i \ge Z$, then

$$o_{i1k} = x_{ik} \quad k = 1, 2, \ldots, K_i$$

$$o_{i2k} = x_{i,(k+K_i)} \quad k = 1, 2, \ldots, K_i$$

$$o_{i3k} = x_{i,(k+2K_i)} \quad k = 1, 2, \ldots, K_i$$

$\ldots$

$o_{iC_ik} = x_{i(k+(C_i-1)K_i)}$ $k = 1, 2, \ldots, K_i - Y_i$

$o_{iC_ik} = 0$ $k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \ldots, K_i$

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where i is the TrCH number, r is the code block number, and K_i is the number of bits in each code block. The number of code blocks on TrCH i is denoted by C_i. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$. The encoded blocks are serially multiplexed so that the block with lowest index r is output first from the channel coding block. The bits output are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and $E_i = C_iY_i$. The output bits are defined by the following relations:

$c_{ik} = y_{i1k}$ $k = 1, 2, \ldots, Y_i$

$c_{ik} = y_{i,2,(k-Y_i)}$ $k = Y_i + 1, Y_i + 2, \ldots, 2Y_i$

$c_{ik} = y_{i,3,(k-2Y_i)}$ $k = 2Y_i + 1, 2Y_i + 2, \ldots, 3Y_i$

$\ldots$

$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)}$ $k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, \ldots, C_iY_i$

The relation between o_{irk} and Y_{irk} and between K_i and Y_i is dependent on the channel coding scheme.

The following channel coding schemes can be applied to transport channels:

- Convolutional coding
- Turbo coding
- No channel coding

The values of Y_i in connection with each coding scheme:

- Convolutional coding, ½ rate: $Y_i = 2*K_i + 16$; 1/3 rate: $Y_i = 3*K_i + 24$
- Turbo coding, 1/3 rate: $Y_i = 3*K_i + 12$
- No channel coding, $Y_i = K_i$

Table 4.2.3-1: Error Correction Coding Parameters

Transport channel type	Coding scheme	Coding rate
BCH	Convolutional code	1/2
PCH		
FACH		
RACH		
DCH, DSCH, USCH		1/3, 1/2
	Turbo code	1/3
	No coding	

4.2.3.1 Convolutional Coding

- Constraint length K=9. Coding rates 1/2 and 1/3.
- The configuration of the convolutional coder is presented in figure 4-2.

- The output from the convolutional coder shall be done in the order output0, output1,output2, output0, output1,…, output2. (When coding rate is 1/2, output is done up to output 1).
- The initial value of the shift register of the coder shall be "all 0".
- K-1 tail bits (value 0) shall be added to the end of the code block before encoding.

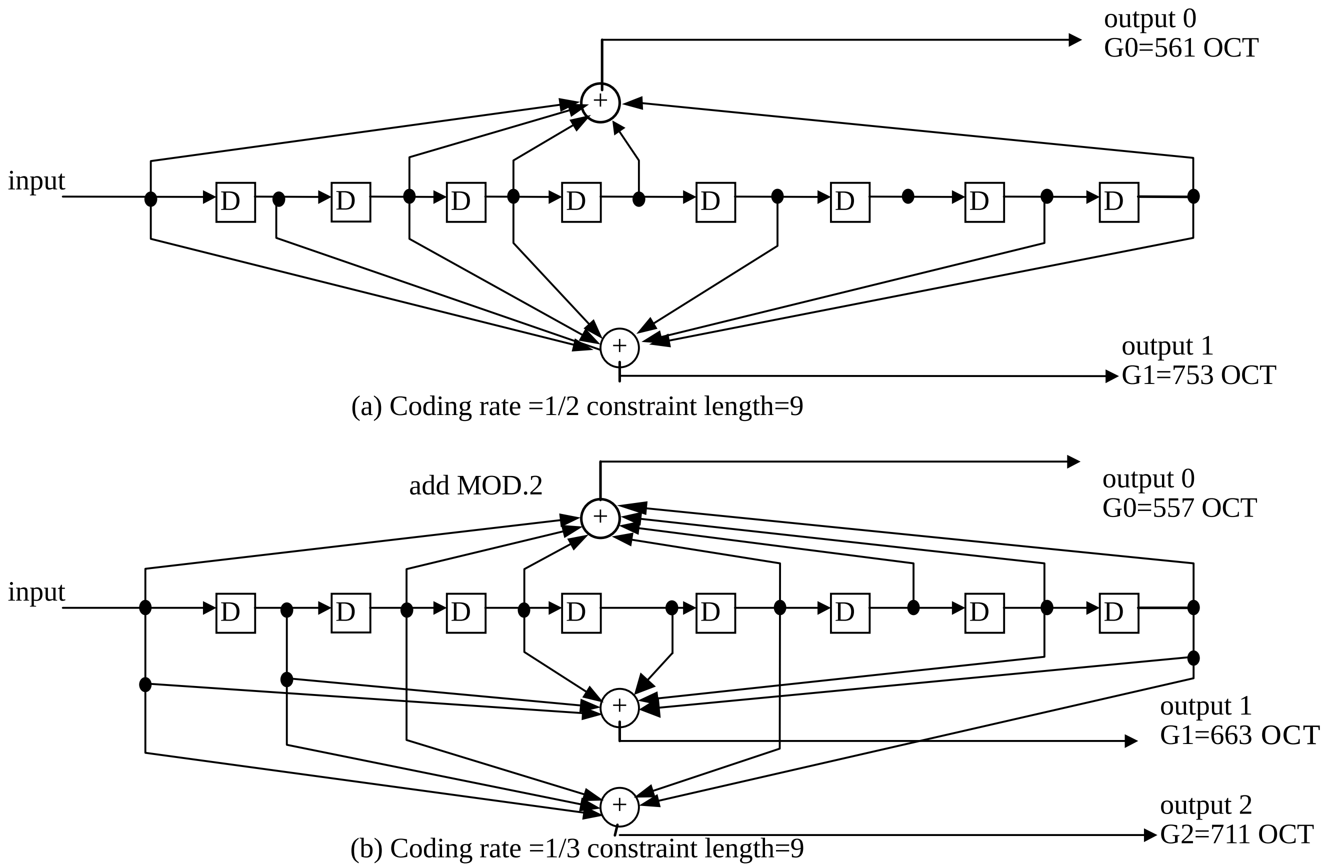

Figure 4-2: Convolutional Coder

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

For data services requiring quality of service between 10^{-3} and 10^{-6} BER inclusive, parallel concatenated convolutional code (PCCC) with 8-state constituent encoders is used.

The transfer function of the 8-state constituent code for PCCC is

$$G(D)=\left[1, \frac{n(D)}{d(D)}\right]$$

where,

$d(D)=1+D^2+D^3$

$n(D)=1+D+D^3$.

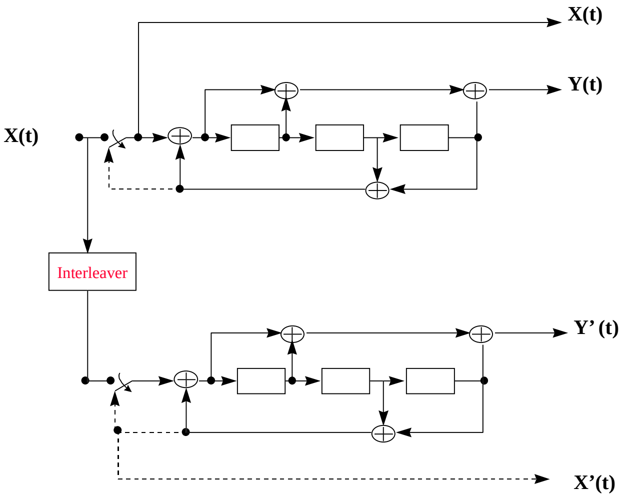

Figure 4-3: Structure of the 8-state PCCC encoder (dotted lines effective for trellis termination only)

The initial value of the shift registers of the PCCC encoder shall be all zeros.

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc.

4.2.3.2.2 Trellis termination in turbo code

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are added after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of figure 4-3 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of figure 4-3 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X'(t) Y'(t) X'(t+1) Y'(t+1) X'(t+2) Y'(t+2).

4.2.3.2.3 Turbo code internal interleaver

Figure 4-4 depicts the overall 8-State PCCC Turbo coding scheme including Turbo code internal interleaver. The Turbo code internal interleaver consists of mother interleaver generation and pruning. For arbitrary given block length K, one mother interleaver is selected from the 134 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l*-bits are pruned in order to adjust the mother interleaver to the block length K. Tail bits T_1 and T_2 are added for constituent encoders RSC1 and RSC2, respectively. The definition of *l* is shown in section 4.2.3.2.3.2..

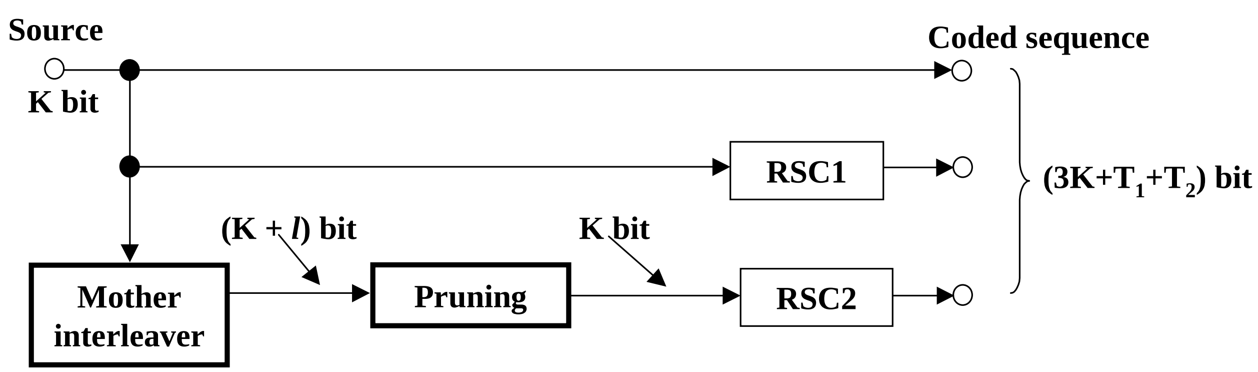

Figure 4-4: Overall 8 State PCCC Turbo Coding

4.2.3.2.3.1 Mother interleaver generation

The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320 to 5114 bits).

First Stage:

(1) Determine the number of rows R such that

R=10 (K = 481 to 530 bits; Case-1)

R=20 (K = any other block length except 481 to 530 bits; Case-2)

(2) Determine the number of columns C such that

Case-1; C = p = 53

Case-2;

(i) find minimum prime p such that,

0 =< (p+1)-K/R

(ii) if (0 =< p-K/R) then go to (iii)

else C = p+1.

(iii) if (0 =< p-1-K/R) then C=p-1.

Else C = p.

(3) The input sequence of the interleaver is written into the RxC rectangular matrix row by row starting from row 0.

Second Stage:

A. If C = p

(A-1) Select a primitive root g_0 from table 4.2.2-2.

(A-2) Construct the base sequence c(i) for intra-row permutation as:

$c(i) = [g_0 \times c(i-1)] \bmod p$, i =1,2,…(p-2)., c(0) = 1.

(A-3) Select the minimum prime integer set $\{q_j\}$ (j=1,2,…R-1) such that

g.c.d$\{q_j, p-1\}$ =1

$q_j > 6$

$q_j > q_{(j-1)}$

where g.c.d. is greatest common divider. And $q_0 = 1$.

(A-4) The set $\{q_j\}$ is permuted to make a new set $\{p_j\}$ such that

$p_{P(j)} = q_j$, j = 0, 1, …. R-1,

where P(j) is the inter-row permutation pattern defined in the third stage.

(A-5) Perform the j-th (j = 0,1, 2, …, C-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$, i =0,1,2,…, (p-2)., and $c_j(p-1) = 0$,

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

If C = p+1

(B-1) Same as case A-1.

(B-2) Same as case A-2.

(B-3) Same as case A-3.

(B-4) Same as case A-4.

(B-5) Perform the *j*-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$, i =0,1,2,…, (p-2)., $c_j(p\text{-}1) = 0$, and $c_j(p) = p$,

where $c_j(i)$ is the input bit position of *i*-th output after the permutation of *j*-th row.

(B-6) If (K = C x R) then exhange $c_{R-1}(p)$ with $c_{R-1}(0)$.

<u>If C = p-1</u>

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the *j*-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$ -1, i =0,1,2,…, (p-2).,

where $c_j(i)$ is the input bit position of *i*-th output after the permutation of *j*-th row.

<u>Third Stage:</u>

Perform the inter-row permutation based on the following P(j) (j=0,1, …, R-1) patterns, where P(j) is the original row position of the *j*-th permuted row.

P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R=20

P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R=20

P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R=10

The usage of these patterns is as follows:

Block length K: P(j)

320 to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

2281 to 2480-bit: P_B

2481 to 3160-bit: P_A

3161 to 3210-bit: P_B

3211 to 5114-bit: P_A

(2) The output of the mother interleaver is the sequence read out column by column from the permuted R $\times$ C matrix starting from column 0.

Table 4.2.3-2: Table of prime p and associated primitive root

p	g_o	P	g_o	p	g_o	P	g_o	p	g_o
17	3	59	2	103	5	157	5	211	2
19	2	61	2	107	2	163	2	223	3
23	5	67	2	109	6	167	5	227	2
29	2	71	7	113	3	173	2	229	6
31	3	73	5	127	3	179	2	233	3
37	2	79	3	131	2	181	2	239	7
41	6	83	2	137	3	191	19	241	7
43	3	89	3	139	2	193	5	251	6
47	5	97	5	149	2	197	2	257	3
53	2	101	2	151	6	199	3		

4.2.3.2.3.2 Definition of the number of pruning bits

The output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence. The pruning bits number *l* is defined as:

$l = R \times C - K$,

where R is the row number and C is the column number defined in section 4.2.3.2.3.1.

4.2.4 Radio frame size equalisation

Radio frame size equalisation is padding the input bit sequence in order to ensure that the output can be segmented in F_i data segments of same size as described in the section 4.2.6.

The input bit sequence to the radio frame size equalisation is denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where *i* is TrCH number and E_i the number of bits. The output bit sequence is denoted by $t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$, where T_i is the number of bits. The output bit sequence is derived as follows:

$t_{ik} = c_{ik}$, for k = 1… E_i and

t_{ik} = {0 | 1} for k= E_i +1… T_i, if $E_i < T_i$

where

$T_i = F_i * N_i$ and

$N_i = \lfloor (E_i - 1)/F_i \rfloor + 1$ is the number of bits per segment after size equalisation.

4.2.5 1st interleaving

The 1[st] interleaving is a block interleaver with inter-column permutations. The input bit sequence to the 1[st] interleaver is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where *i* is TrCH number and X_i the number of bits (at this stage X_i is assumed and guaranteed to be an integer multiple of TTI). The output bit sequence is derived as follows:

1) Select the number of columns C_I from table 4.2.5-1.
2) Determine the number of rows R_I defined as $R_I = X_i/C_I$
3) Write the input bit sequence into the $R_I \times C_I$ rectangular matrix row by row starting with bit $x_{i,1}$ in the first column of the first row and ending with bit $x_{i,(R_I C_I)}$ in column C_I of row R_I:

$$\begin{bmatrix} x_{i1} & x_{i2} & x_{i3} & \ldots & x_{iC_I} \\ x_{i,(C_I+1)} & x_{i,(C_I+2)} & x_{i,(C_I+3)} & \ldots & x_{i,(2C_I)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{i,((R_I-1)C_I+1)} & x_{i,((R_I-1)C_I+2)} & x_{i,((R_I-1)C_I+3)} & \ldots & x_{i,(R_I C_I)} \end{bmatrix}$$

4) Perform the inter-column permutation based on the pattern $\{P_1(j)\}$ (j=0,1, …, C-1) shown in table 4.2.5-1, where $P_1(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{ik}:

$$\begin{bmatrix} y_{i1} & y_{i,(R_I+1)} & y_{i,(2R_I+1)} & \ldots & y_{i,((C_I-1)R_I+1)} \\ y_{i2} & y_{i,(R_I+2)} & y_{i,(2R_I+2)} & \ldots & y_{i,((C_I-1)R_I+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{iR_I} & y_{i,(2R_I)} & y_{i,(3R_I)} & \ldots & y_{i,(C_I R_I)} \end{bmatrix}$$

5) Read the output bit sequence $y_{i1}, y_{i2}, y_{i3}, \ldots, y_{i,(C_I R_I)}$ of the 1st interleaving column by column from the inter-column permuted $R_I \times C_I$ matrix. Bit $y_{i,1}$ corresponds to the first row of the first column and bit $y_{i,(R_I C_I)}$ corresponds to row R_I of column C_I.

The bits input to the 1st interleaving are denoted by $t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$, where i is the TrCH number and T_i the number of bits. Hence, $x_{ik} = t_{ik}$ and $X_i = T_i$.

The bits output from the 1st interleaving are denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$, and $d_{ik} = y_{ik}$.

Table 4.2.5-1

TTI	Number of columns C_1	Inter-column permutation patterns
10 ms	1	{0}
20 ms	2	{0,1}
40 ms	4	{0,2,1,3}
80 ms	8	{0,4,2,6,1,5,3,7}

4.2.6 Radio frame segmentation

When the transmission time interval is longer than 10 ms, the input bit sequence is segmented and mapped onto consecutive radio frames. Following radio frame size equalisation the input bit sequence length is guaranteed to be an integer multiple of F_i.

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where i is the TrCH number and X_i is the number bits. The Fi output bit sequences per TTI are denoted by $y_{i,n_i1}, y_{i,n_i2}, y_{i,n_i3}, \ldots, y_{i,n_iY_i}$ where n_i is the radio frame number in current TTI and Y_i is the number of bits per radio frame for TrCH i. The output sequences are defined as follows:

$y_{i,n_ik} = x_{i,((n_i-1)Y_i)+k}$, $n_i = 1\ldots F_i$, $k = 1\ldots Y_i$

where

$Y_i = (X_i / F_i)$ is the number of bits per segment,

x_{ik} is the k^th bit of the input bit sequence and

y_{i,n_ik} is the k^th bit of the output bit sequence corresponding to the n^th radio frame

The n_i –th segment is mapped to the n_i –th radio frame of the transmission time interval.

The input bit sequence to the radio frame segmentation is denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$, where *i* is the TrCH number and T_i the number of bits. Hence, $x_{ik} = d_{ik}$ and $X_i = T_i$.

The output bit sequence corresponding radio frame n_i is denoted by $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$, where *i* is the TrCH number and N_i is the number of bits. Hence, $e_{i,k} = y_{i,n_i k}$ and $N_i = Y_i$.

4.2.7 Rate matching

Rate matching means that bits on a TrCH are repeated or punctured. Higher layers assign a rate-matching attribute for each TrCH. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

The number of bits on a TrCH can vary between different transmission time intervals. When the number of bits between different transmission time intervals is changed, bits are repeated to ensure that the total bit rate after TrCH multiplexing is identical to the total channel bit rate of the allocated physical channels.

Notation used in section 4.2.7 and subsections:

N_{ij}: Number of bits in a radio frame before rate matching on TrCH *i* with transport format combination *j* .

ΔN_{ij} : If positive – number of bits to be repeated in each radio frame on TrCH *i* with transport format combination *j*.

If negative – number of bits to be punctured in each radio frame on TrCH *i* with transport format combination *j*.

RM_i: Semi-static rate matching attribute for TrCH *i*. Signalled from higher layers.

PL: Puncturing limit. This value limits the amount of puncturing that can be applied in order to minimise the number of physical channels. Signalled from higher layers.

$N_{data,j}$: Total number of bits that are available for a CCTrCH in a radio frame with transport format combination j.

P: maximum number of physical channels for a CCTrCH.

I: Number of TrCHs in a CCTrCH.

Z_{mj}: Intermediate calculation variable.

F_i: Number of radio frames in the transmission time interval of TrCH *i*.

n_i: Radio frame number in the transmission time interval of TrCH *i* ($0 \le n_i < F_i$).

q: Average puncturing or repetition distance(normalised to only show the remaining rate matching on top of an integer number of repetitions).

$I_F(n_i)$: The inverse interleaving function of the 1st interleaver (note that the inverse interleaving function is identical to the interleaving function itself for the 1st interleaver).

$S(n_i)$: The shift of the puncturing or repetition pattern for radio frame n_i.

$TF_i(j)$: Transport format of TrCH i for the transport format combination j.

TFS(i): The set of transport format indexes *l* for TrCH i.

e_{ini}: Initial value of variable e in the rate matching pattern determination algorithm of section 4.2.7.3.

e_{plus} Increment of variable *e* in the rate matching pattern determination algorithm of section 4.2.7.3.

e_{minus} Decrement of variable *e* in the rate matching pattern determination algorithm of section 4.2.7.3.

b: Indicates systematic and parity bits.

b=1: Systematic bit. *X(t)* in 4.2.3.2.1.

Y: *b*=2: 1st parity bit (from the upper Turbo constituent encoder). *Y(t)* in section 4.2.3.2.1.

Y': b=3: 2nd parity bit (from the lower Turbo constituent encoder). *Y'(t)* in section 4.2.3.2.1.

4.2.7.1 Determination of rate matching parameters

The following relations are used when calculating the rate matching pattern:

$$Z_{0,j} = 0$$

$$Z_{ij} = \left\lfloor \frac{\sum_{m=1}^{i} RM_m \cdot N_{mj}}{\sum_{m=1}^{I} RM_m \cdot N_{mj}} \cdot N_{data,j} \right\rfloor \quad \text{for all i = 1 .. I}$$

$$\Delta N_{ij} = Z_{ij} - Z_{i-1,j} - N_{ij} \quad \text{for all i = 1 .. I}$$

Puncturing can be used to minimise the required transmission capacity. The maximum amount of puncturing that can be applied is signalled from higher layers and denoted by PL. The possible values for N_{data} depend on the number of physical channels P, allocated to the respective CCTrCH, and on their characteristics (spreading factor, length of midamble and TFCI, usage of TPC and multiframe structure), which is given in [7].

Denote the number of data bits in each physical channel by $N_{k,Sk}$, where k refers to the sequence number $1\leq k\leq P$ of this physical channel in the allocation message, and the second index Sk indicates the spreading factor with the possible values $\{16, 8, 4, 2, 1\}$, respectively. For each physical channel an individual minimum spreading factor Sk_{min} is transmitted by means of the higher layer. Then, for N_{data} one of the following values in ascending order can be chosen: $\{N_{1,16}, ..., N_{1,S1min}, N_{1,S1min}+N_{2,16}, ..., N_{1,S1min}+N_{2,S2min}, ..., N_{1,S1min}+N_{2,S2min}+...+N_{P,16}, ..., N_{1,S1min}+N_{2,S2min}+...+N_{P,SPmin}\}$.

$N_{data,j}$ for the transport format combination j is determined by executing the following algorithm:

SET1 = { N_{data} such that $N_{data} - PL \cdot \sum_{x=1}^{I} \frac{RM_x}{\min_{1\leq y\leq I}\{RM_y\}} \cdot N_{x,j}$ is non negative }

$N_{data,j}$ = min SET1

The number of bits to be repeated or punctured, ΔN_{ij}, within one radio frame for each TrCH i is calculated with the relations given at the beginning of this section for all possible transport format combinations j and selected every radio frame.

If ΔN_{ij} = 0 then the output data of the rate matching is the same as the input data and the rate matching algorithm of section 4.2.7.3 does not need to be executed.

Otherwise, the rate matching pattern is calculated with the algorithm described in section 4.2.7.3. For this algorithm the parameters e_{ini}, e_{plus}, e_{minus}, and X_i are needed, which are calculated according to the equations in section 4.2.7.1.1 and 4.2.7.1.2

4.2.7.1.1 Uncoded and convolutionally encoded TrCHs

a = 2

$\Delta N_i = \Delta N_{i,j}$

$X_i = N_{i,j}$

R = ΔN_{ij} mod N_{ij} -- note: in this context ΔN_{ij} mod N_{ij} is in the range of 0 to N_{ij}-1 i.e. -1 mod 10 = 9.

if R $\neq$ 0 and 2R $\leq$ N_{ij}

then q = $\lceil N_{ij} / R \rceil$

else

$q = \lceil N_{ij} / (R - N_{ij}) \rceil$

endif

-- note: q is a signed quantity.

If q is even

then $q' = q + gcd(|q|, F_i)/F_i$ -- where gcd ($|q|, F_i$) means greatest common divisor of $|q|$ and F_i

-- note that q' is not an integer, but a multiple of 1/8

else

q' = q

endif

for x = 0 to F_i-1

$S(I_F(|\lfloor x*q' \rfloor| \bmod F_i)) = (|\lfloor x*q' \rfloor| \operatorname{div} F_i)$

end for

$e_{ini} = (a \cdot S(n_i) \cdot |\Delta N_i| + 1) \bmod (a \cdot X_i)$

$e_{plus} = a \cdot X_i$

$e_{minus} = a \cdot |\Delta N_i|$

puncturing for ΔN_i<0, repetition otherwise.

4.2.7.1.2 Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{i,j} > 0$, the parameters in section 4.2.7.1.1 are used.

If puncturing is to be performed, the parameters below shall be used. Index *b* is used to indicate systematic (*b*=1), 1st parity (*b*=2), and 2nd parity bit (*b*=3).

a = 2 when *b*=2

a = 1 when *b*=3

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{i,j}/2 \rfloor, & b=2 \\ \lceil \Delta N_{i,j}/2 \rceil, & b=3 \end{cases}$$

$X_i = \lfloor N_{i,j}/3 \rfloor$,

$q = \lfloor X_i / |\Delta N_i| \rfloor$

if($q \leq 2$)

for x=0 to F_i-1

$S[I_F[(3x+b-1) \bmod F_i]] = x \bmod 2$; end for

else

if *q* is even

then $q' = q - gcd(q, F_i)/F_i$ -- where $gcd(q, F_i)$ means greatest common divisor of *q* and F_i

-- note that *q'* is not an integer, but a multiple of 1/8

else $q' = q$

endif

for x=0 to $F_i - 1$

$r = \lceil x*q' \rceil \bmod F_i$;

$S[I_F[(3r+b-1) \bmod F_i]] = \lceil x*q' \rceil \text{ div } F_i$;

endfor

endif

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 4.2.7.3, where:

X_i is as above,

$e_{ini} = (a \cdot S(n_i) \cdot |\Delta N_i| + X_i) \bmod (a \cdot X_i)$, if $e_{ini} = 0$ then $e_{ini} = a \cdot X_i$.

$e_{plus} = a \cdot X_i$

$e_{minus} = a \cdot |\Delta N_i|$

4.2.7.2 Bit separation and collection for rate matching

The systematic bits (excluding bits for trellis termination) of turbo encoded TrCHs shall not be punctured. The systematic bit, first parity bit, and second parity bit in the bit sequence input to the rate matching block are therefore separated from each other. Puncturing is only applied to the parity bits and systematic bits used for trellis termination.

The bit separation function is transparent for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition. The bit separation and bit collection are illustrated in figures 4-5 and 4-6.

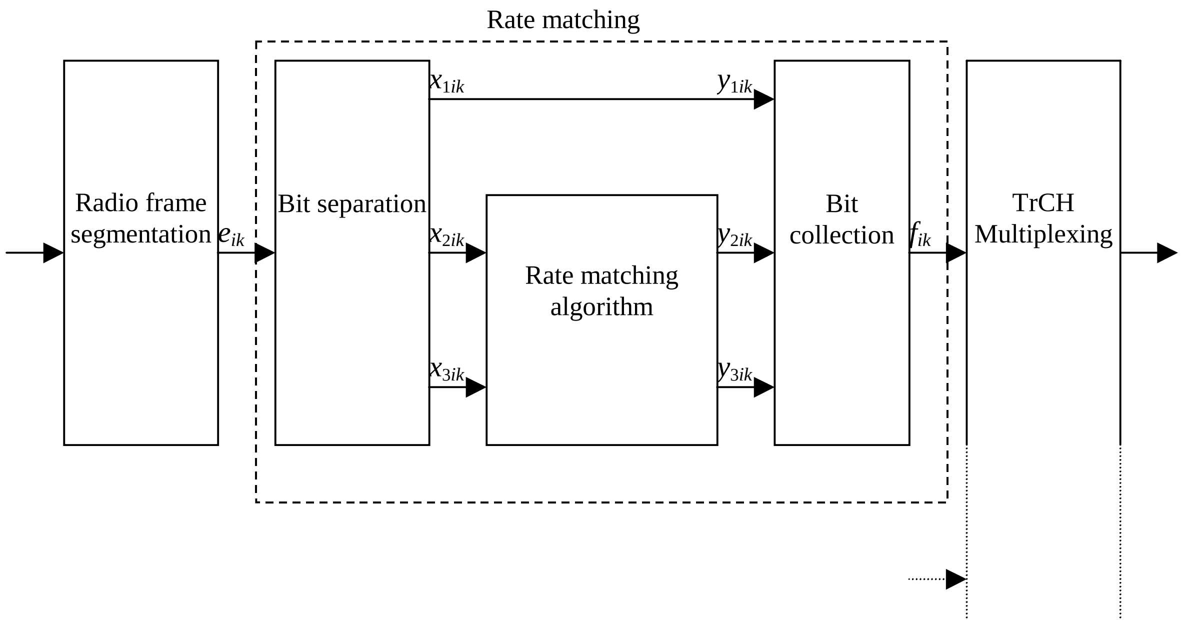

Figure 4-5: Puncturing of turbo encoded TrCHs

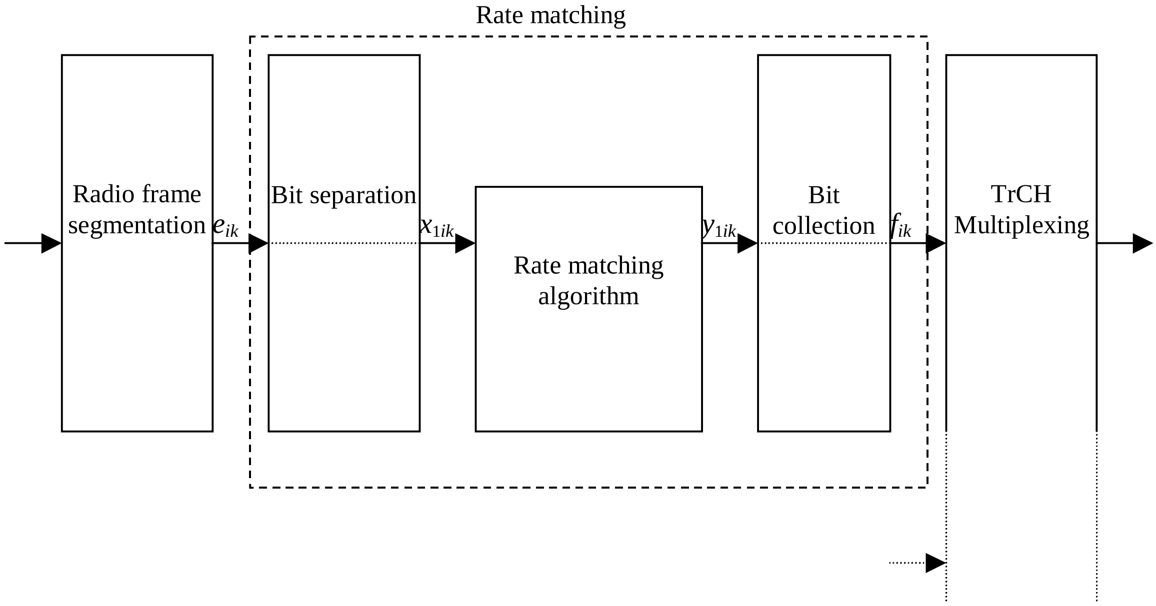

Figure 4-6: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition.

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. The offsets α_b for the systematic (b=1) and parity bits ($b\in\{2, 3\}$) are listed in table 4.2.7-1.

Table 4.2.7-1: TTI dependent offset needed for bit separation

TTI (ms)	α_1	α_2	α_3
10, 40	0	1	2
20, 80	0	2	1

The bit separation is different for different radio frames in the TTI. A second offset is therefore needed. The radio frame number for TrCH i is denoted by n_i. and the offset by β_{n_i}.

Table 4.2.7-2: Radio frame dependent offset needed for bit separation

TTI (ms)	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
10	0	NA	NA	NA	NA	NA	NA	NA
20	0	1	NA	NA	NA	NA	NA	NA
40	0	1	2	0	NA	NA	NA	NA
80	0	1	2	0	1	2	0	1

4.2.7.2.1 Bit separation

The bits input to the rate matching are denoted by $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$, where i is the TrCH number and N_i is the number of bits input to the rate matching block. Note that the transport format combination number j for simplicity has been left out in the bit numbering, i.e. $N_i=N_{ij}$. The bits after separation are denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \ldots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, b indicates systematic, first parity, or second parity bit. For all other cases b is defined to be 1. X_i is the number of bits in each separated bit sequence. The relation between e_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

$x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3}$ k = 1, 2, 3, …, X_i $X_i = \lfloor N_i/3 \rfloor$

$x_{1,i,\lfloor N_i/3 \rfloor+k} = e_{i,3\lfloor N_i/3 \rfloor+k}$ k = 1, …, N_i mod 3   Note: When (N_i mod 3) = 0 this row is not needed.

$x_{2,i,k} = e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3}$ k = 1, 2, 3, …, X_i $X_i = \lfloor N_i/3 \rfloor$

$x_{3,i,k} = e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3}$ k = 1, 2, 3, …, X_i $X_i = \lfloor N_i/3 \rfloor$

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$x_{1,i,k} = e_{i,k}$ k = 1, 2, 3, …, X_i $X_i = N_i$

4.2.7.2.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in section 4.2.7.3. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \ldots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \ldots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}$, where i is the TrCH number and $V_i = N_{ij} + \Delta N_{ij}$. The relations between y_{bik}, z_{bik}, and f_{ik} are given below.

For turbo encoded TrCHs with puncturing ($Y_i=X_i$):

$z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \bmod 3} = y_{1,i,k}$ k = 1, 2, 3, …, Y_I

$z_{i,3\lfloor N_i/3 \rfloor+k} = y_{1,i,\lfloor N_i/3 \rfloor+k}$ k = 1, …, N_i mod 3 Note: When (N_i mod 3) = 0 this row is not needed.

$z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \bmod 3} = y_{2,i,k}$ k = 1, 2, 3, …, Y_i

$z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \bmod 3} = y_{3,i,k}$ k = 1, 2, 3, …, Y_i

After the bit collection, bits $z_{i,k}$ with value δ, where $\delta \notin \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

$z_{i,k} = y_{1,i,k}$ k = 1, 2, 3, …, Y_i

When repetition is used, $f_{i,k}=z_{i,k}$ and $Y_i=V_i$.

When puncturing is used, $Y_i=X_i$ and bits $z_{i,k}$ with value δ, where $\delta \notin \{0, 1\}$, are removed from the bit sequence. Bit $f_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index k after puncturing, bit $f_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index k after puncturing, and so on.

4.2.7.3 Rate matching pattern determination

The bits input to the rate matching are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where i is the TrCH and X_i is the parameter given in section 4.2.7.1.1 and 4.2.7.1.2.The bits output from the rate matching are denoted by $f_{i1}, f_{i2}, f_{13}, \ldots, f_{iV_i}$, where i is the TrCH number and $V_i=N+\Delta N$.

Note that the transport format combination number j for simplicity has been left out in the bit numbering.

The rate matching rule is as follows:

if puncturing is to be performed

e = e_{ini} -- initial error between current and desired puncturing ratio

```
        m = 1            -- index of current bit
        do while m <= X_i
            e = e – e_minus        -- update error
            if e <= 0 then         -- check if bit number m should be punctured
                set bit x_i,m to δ where δ∉{0, 1}
                e = e + e_plus     -- update error
            end if
            m = m + 1              --  next bit
        end do
    else
        e = e_ini    -- initial error between current and desired puncturing ratio
        m = 1              -- index of current bit
        do while m <= X_i
            e = e – e_minus        -- update error
            do while e <= 0        -- check if bit number m should be repeated
                repeat bit x_i,m
                e = e + e_plus   -- update error
            end do
            m = m + 1              --  next bit
        end do
    end if
```

A repeated bit is placed directly after the original one.

4.2.8 TrCH multiplexing

Every 10 ms, one radio frame from each TrCH is delivered to the TrCH multiplexing. These radio frames are serially multiplexed into a coded composite transport channel (CCTrCH).

The bits input to the TrCH multiplexing are denoted by $f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}$, where i is the TrCH number and V_i is the number of bits in the radio frame of TrCH i. The number of TrCHs is denoted by I. The bits output from TrCH multiplexing are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits, i.e. $S = \sum_i V_i$. The TrCH multiplexing is defined by the following relations:

$s_k = f_{1k}$ $k = 1, 2, \ldots, V_1$

$s_k = f_{2,(k-V_1)}$ $k = V_1+1, V_1+2, \ldots, V_1+V_2$

$s_k = f_{3,(k-(V_1+V_2))}$ $k = (V_1+V_2)+1, (V_1+V_2)+2, \ldots, (V_1+V_2)+V_3$

$\ldots$

$s_k = f_{I,(k-(V_1+V_2+\ldots+V_{I-1}))}$ $k = (V_1+V_2+\ldots+V_{I-1})+1, (V_1+V_2+\ldots+V_{I-1})+2, \ldots, (V_1+V_2+\ldots+V_{I-1})+V_I$

4.2.9 Physical channel segmentation

When more than one PhCH is used, physical channel segmentation divides the bits among the different PhCHs. The bits input to the physical channel segmentation are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits input to the physical channel segmentation block. The number of PhCHs is denoted by *P*.

The bits after physical channel segmentation are denoted $u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU_p}$, where *p* is PhCH number and U_p is the in general variable number of bits in the respective radio frame for each PhCH. The relation between s_k and u_{pk} is given below.

Bits on first PhCH after physical channel segmentation:

$u_{1k} = s_k$ $k = 1, 2, \ldots, U_1$

Bits on second PhCH after physical channel segmentation:

$u_{2k} = s_{(k+U_1)}$ $k = 1, 2, \ldots, U_2$

…

Bits on the P^{th} PhCH after physical channel segmentation:

$u_{Pk} = s_{(k+U_1+\ldots+U_{P-1})}$ $k = 1, 2, \ldots, U_P$

4.2.10 2nd interleaving

The 2nd interleaving can be applied jointly to all data bits transmitted during one frame, or separately within each timeslot, on which the CCTrCH is mapped. The selection of the 2nd interleaving scheme is controlled by higher layer.

4.2.10.1 Frame related 2nd interleaving

In case of frame related interleaving, the bits input to the 2^{nd} interleaver are denoted $x_1, x_2, x_3, \ldots, x_U$, where *U* is the total number of bits after TrCH multiplexing transmitted during the respective radio frame.

The relation between x_k and the bits u_{pk} in the respective physical channels is given below:

$x_k = u_{1k}$ $k = 1, 2, \ldots, U_1$

$x_{(k+U_1)} = u_{2k}$ $k = 1, 2, \ldots, U_2$

…

$x_{(k+U_1+\ldots+U_{P-1})} = u_{Pk}$ $k = 1, 2, \ldots, U_P$

The following steps have to be performed once for each CCTrCH:

(1) Set the number of columns $C_2 = 30$. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that $U \leq R_2C_2$.

(3) The bits input to the 2^{nd} interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_{30} \\ x_{31} & x_{32} & x_{33} & \ldots & x_{60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{(R_2-1)30+1} & x_{(R_2-1)30+2} & x_{(R_2-1)30+3} & \ldots & x_{R_2 \cdot 30} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern $\{P_2(j)\}$ (j = 0, 1, …, C_2-1) that is shown in table 4.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_k.

$$\begin{bmatrix} y_1 & y_{R_2+1} & y_{2R_2+1} & \cdots y_{29R_2+1} \\ y_2 & y_{R_2+2} & y_{2R_2+2} & \cdots y_{29R_2+2} \\ \vdots & \vdots & \vdots & \ldots \quad \vdots \\ y_{R_2} & y_{2R_2} & y_{3R_2} & \cdots \; y_{30R_2} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_k that corresponds to bits x_k with k>U are removed from the output. The bits after 2nd interleaving are denoted by $v_1, v_2, \ldots, v_U$, where v_1 corresponds to the bit y_k with smallest index k after pruning, v_2 to the bit y_k with second smallest index k after pruning, and so on.

4.2.10.2 Timeslot related 2nd interleaving

In case of timeslot related 2nd interleaving, the bits input to the 2nd interleaver are denoted $x_{t1}, x_{t2}, x_{t3}, \ldots, x_{tU_t}$, where *t* refers to a certain timeslot, and U_t is the number of bits transmitted in this timeslot during the respective radio frame.

In each timeslot *t* the relation between x_{tk} and u_{pk} is given below with P_t refering to the number of physical channels within the respective timeslot:

$x_{tk} = u_{1k}$ k = 1, 2 , …, U_1

$x_{t(k+U_1)} = u_{2k}$ k = 1, 2 , …, U_2

…

$x_{t(k+U_1+\ldots+U_{P_t-1})} = u_{P_t k}$ k = 1, 2 , …, U_{P_t}

The following steps have to be performed for each timeslot *t*, on which the respective CCTrCH is mapped:

(1) Set the number of columns C_2 = 30. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that $U_t \le R_2 C_2$.

(3) The bits input to the 2nd interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_{t1} & x_{t2} & x_{t3} & \ldots & x_{t30} \\ x_{t31} & x_{t32} & x_{t33} & \ldots & x_{t60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{t,((R_2-1)30+1)} & x_{t,((R_2-1)30+2)} & x_{t,((R_2-1)30+3)} & \ldots & x_{t,(R_2 30)} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern $\{P_2(j)\}$ (j = 0, 1, …, C_2-1) that is shown in table 4.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{tk}.

$$\begin{bmatrix} y_{t1} & y_{t,(R_2+1)} & y_{t,(2R_2+1)} & \cdots y_{t,(29R_2+1)} \\ y_{t2} & y_{t,(R_2+2)} & y_{t,(2R_2+2)} & \cdots y_{t,(29R_2+2)} \\ \vdots & \vdots & \vdots & \cdots \vdots \\ y_{tR_2} & y_{t,(2R_2)} & y_{t,(3R_2)} & \cdots y_{t,(30R_2)} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_{tk} that corresponds to bits x_{tk} with $k>U_t$ are removed from the output. The bits after 2nd interleaving are denoted by $v_{t1}, v_{t2}, \ldots, v_{tU_t}$, where v_{t1} corresponds to the bit y_{tk} with smallest index k after pruning, v_{t2} to the bit y_{tk} with second smallest index k after pruning, and so on.

Table 4.2.10-1

Column number C_2	Inter-column permutation pattern
30	{0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17}

4.2.11 Physical channel mapping

The PhCH for both uplink and downlink is defined in [6]. The bits after physical channel mapping are denoted by $w_{p1}, w_{p2}, \ldots, w_{pU_p}$, where p is the PhCH number and U_p is the number of bits in one radio frame for the respective PhCH. The bits w_{pk} are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to k. The mapping scheme depends on the applied 2nd interleaving scheme.

4.2.11.1 Mapping scheme after frame related 2nd interleaving

4.2.11.1.1 Mapping scheme after frame related 2nd interleaving in uplink

In uplink there are at most two codes allocated (P≤2). If there is only one code, the same mapping as for downlink is applied, see section 6.2.11.1.2. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. Then denote the inverse relation of the spreading factors s1: s2 = SF2: SF1, where the smallest possible integers are used for s1 and s2.

The following mapping rule is applied:

Bits are mapped on the first PhCH (in forward order) if *(k-1)mod(s1+s2) = 0*, …, s1-1:

$$w_{1,(k\,div(s1+s2))\cdot s1 + k \bmod (s1+s2)} = v_k$$

else bits are mapped on the second PhCH (in reverse order):

$$w_{2,U_2-(k\,div(s1+s2))\cdot s2 + k \bmod (s1+s2) - s1} = v_k$$

This formula is applied starting with k=1 and increasing k until one of the PhCH is completely filled. From then on, the remaining bits are mapped on the PhCH which has not been filled in the same order (forward or reverse depending on the PhCH) as used previously on that PhCH.

4.2.11.1.2 Mapping scheme after frame related 2nd interleaving in downlink

The mapping is equivalent to block interleaving, writing in colomns, but a PhCH with an odd number is filled in forward order, were as a PhCH with an even number is filled in reverse order.

The following mapping rule is applied:

Bits are mapped on an odd numbered PhCH (in forward order) according to the following rule, if (k mod P)+1 is odd:

$$w_{k \bmod P+1,\, k\,div\,P} = v_k$$

Bits are mapped on an even numbered PhCH (in reverse order) according to the following rule, if (k mod P)+1 is even:

$$w_{k \bmod P+1, U_P - 1 - k \, div \, P} = v_k$$

This formula is applied starting with k=1 and increasing k until all the PhCHs which carry TFCI are completely filled. From then on, the remaining bits are mapped on the remaining PhCHs in the same order (forward or reverse depending on the PhCH) as previously on these PhCHs.

4.2.11.2 Mapping scheme after timeslot related 2nd interleaving

For each timeslot only those physical channels with $p = 1, 2, \ldots, P_t$ are considered respectively, which are transmitted in that timeslot, and the following mapping scheme is applied:

6.2.11.2.1 Mapping scheme after timeslot related 2nd interleaving in uplink

In uplink there are at most two codes allocated (P≤2). If there is only one code, the same mapping as for downlink is applied, see section 6.2.11.1.2. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. Then denote the inverse relation of the spreading factors s1: s2 = SF2: SF1, where the smallest possible integers are used for s1 and s2.

The following mapping rule is applied:

Bits are mapped on the first PhCH (in forward order) if *(k-1)mod(s1+s2) = 0*, …, s1-1:

$$w_{1,(k \, div (s1+s2)) \cdot s1 + k \bmod (s1+s2)} = v_{tk}$$

else bits are mapped on the second PhCH (in reverse order):

$$w_{2, U_2 - (k \, div (s1+s2)) \cdot s2 + k \bmod (s1+s2) - s1} = v_{tk}$$

This formula is applied starting with k=1 and increasing k until one of the PhCH is completely filled. From then on, the remaining bits are mapped on the PhCH which has not been filled in the same order (forward or reverse depending on the PhCH) as used previously on that PhCH.

6.2.11.2.2 Mapping scheme after timeslot related 2nd interleaving in downlink

The mapping is equivalent to block interleaving, writing in colomns, but a PhCH with an odd number is filled in forward order, were as a PhCH with an even number is filled in reverse order.

The following mapping rule is applied:

Bits are mapped on an odd numbered PhCH (in forward order) according to the following rule, if (k mod P_t)+1 is odd:

$$w_{k \bmod P_t + 1, k \, div \, P_t} = v_{tk}$$

Bits are mapped on an even numbered PhCH (in reverse order) according to the following rule, if (k mod P_t)+1 is even:

$$w_{k \bmod P_t + 1, U_{P_t} - 1 - k \, div \, P_t} = v_{tk}$$

This formula is applied starting with k=1 and increasing k until all the PhCHs which carry TFCI are completely filled. From then on, the remaining bits are mapped on the remaining PhCHs in the same order (forward or reverse depending on the PhCH) as previously on these PhCHs.

4.2.12 Multiplexing of different transport channels onto one CCTrCH, and mapping of one CCTrCH onto physical channels

Different transport channels can be encoded and multiplexed together into one Coded Composite Transport Channel (CCTrCH). The following rules shall apply to the different transport channels which are part of the same CCTrCH:

1) Transport channels multiplexed into one CCTrCh should have co-ordinated timings in the sense that transport blocks arriving from higher layers on different transport channels of potentially different transmission time intervals shall have aligned transmission time instants as shown in figure 4-6.

2) Different CCTrCHs cannot be mapped onto the same physical channel.

3) One CCTrCH shall be mapped onto one or several physical channels.

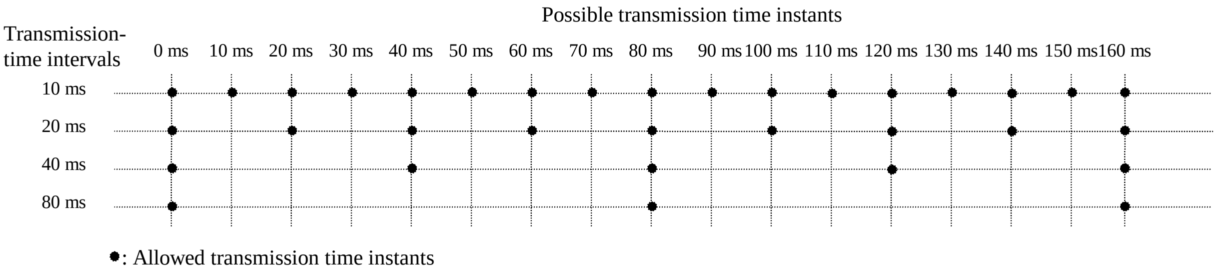

Figure 4-6: Possible transmission time instants regarding CCTrCH

4) Dedicated Transport channels and common transport channels cannot be multiplexed into the same CCTrCH.

5) For the common transport channels, only the FACH and PCH may belong to the same CCTrCH.

6) Each CCTrCH carrying a BCH shall carry only one BCH and shall not carry any other Transport Channel.

7) Each CCTrCH carrying a RACH shall carry only one RACH and shall not carry any other Transport Channel.

Hence, there are two types of CCTrCH

CCTrCH of dedicated type, corresponding to the result of coding and multiplexing of one or several DCH.

CCTrCH of common type, corresponding to the result of the coding and multiplexing of a common channel, i.e. RACH and USCH in the uplink and DSCH, BCH, FACH or PCH in the downlink, respectively.

Transmission of TFCI is possible for CCTrCH containing Transport Channels of:

- Dedicated type
- USCH type
- DSCH type
- FACH and/or PCH type

4.2.12.1 Allowed CCTrCH combinations for one UE

4.2.12.1.1 Allowed CCTrCH combinations on the uplink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

1) several CCTrCH of dedicated type

2) several CCTrCH of common type

4.2.12.1.2 Allowed CCTrCH combinations on the downlink

The following CCTrCH combinations for one UE are allowed, also simultaneously:

3) several CCTrCH of dedicated type

4) several CCTrCH of common type

4.2.13 Transport format detection

Transport format detection can be performed both with and without Transport Format Combination Indicator (TFCI). If a TFCI is transmitted, the receiver detects the transport format combination from the TFCI. When no TFCI is transmitted, so called blind transport format detection may be used, i.e. the receiver side uses the possible transport format combinations as a priori information.

4.2.13.1 Blind transport format detection

Blind transport format detection may be performed in the receiver by trying all possible combinations of the transport format.

4.2.13.2 Explicit transport format detection based on TFCI

4.2.13.2.1 Transport Format Combination Indicator (TFCI)

The Transport Format Combination Indicator (TFCI) informs the receiver of the transport format combination of the CCTrCHs . As soon as the TFCI is detected, the transport format combination, and hence the individual transport channels' transport formats are known, and decoding of the transport channels can be performed.

4.3 Coding for layer 1 control

4.3.1 Coding of transport format combination indicator (TFCI)

Encoding of the TFCI bits depends on the number of them. If there are 6-10 bits of TFCI the channel encoding is done as described in section 4.3.1.1. Also specific coding of less than 6 bits is possible as explained in section 4.3.1.2.

4.3.1.1 Coding of long TFCI lengths

The TFCI bits are encoded using a (32, 10) sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 4.3.3.1-1.

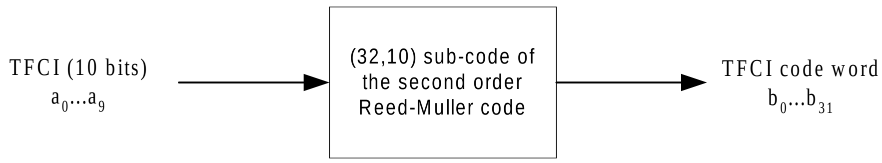

Figure 4.3.3.1-1: Channel coding of TFCI bits

TFCI is encoded by the (32,10) sub-code of second order Reed-Muller code. The code words of the (32,10) sub-code of second order Reed-Muller code are linear combination of some among 10 basis sequences. The basis sequences are as follows in table 4.3.1-1.

Table 4.3.1-1: Basis sequences for (32,10) TFCI code

I	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{I,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$
0	1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	1	0	0	0
2	1	1	1	0	0	0	0	0	0	1
3	1	0	0	1	0	0	1	0	1	1
4	1	1	0	1	0	0	0	0	0	1
5	1	0	1	1	0	0	0	0	1	0
6	1	1	1	1	0	0	0	1	0	0
7	1	0	0	0	1	0	0	1	1	0
8	1	1	0	0	1	0	1	1	1	0
9	1	0	1	0	1	0	1	0	1	1
10	1	1	1	0	1	0	0	0	1	1
11	1	0	0	1	1	0	0	1	1	0
12	1	1	0	1	1	0	0	1	0	1
13	1	0	1	1	1	0	1	0	0	1
14	1	1	1	1	1	0	1	1	1	1
15	1	1	0	0	0	1	1	1	0	0
16	1	0	1	0	0	1	1	1	0	1
17	1	1	1	0	0	1	1	0	1	0
18	1	0	0	1	0	1	0	1	1	1
19	1	1	0	1	0	1	0	1	0	1
20	1	0	1	1	0	1	0	0	1	1
21	1	1	1	1	0	1	0	1	1	1
22	1	0	0	0	1	1	0	1	0	0
23	1	1	0	0	1	1	1	1	0	1
24	1	0	1	0	1	1	1	0	1	0
25	1	1	1	0	1	1	1	0	0	1
26	1	0	0	1	1	1	0	0	1	0
27	1	1	0	1	1	1	1	1	0	0
28	1	0	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	1	0	0	0	0	0	0	0	0	0
31	1	0	0	0	0	1	1	0	0	0

For TFCI bits a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9 (a_0 is LSB and a_9 is MSB), the output code word bits b_i are given by:

$$b_i = \sum_{n=0}^{9} (a_n \times M_{i,n}) \bmod 2$$

where i=0…31. N_{TFCI}=32.

4.3.1.2 Coding of short TFCI lengths

4.3.1.2.1 Coding very short TFCIs by repetition

If the number of TFCI bits is 1 or 2, then repetition will be used for coding. In this case each bit is repeated to a total of 4 times giving 4-bit transmission (N_{TFCI}=4) for a single TFCI bit and 8-bit transmission (N_{TFCI}=8) for 2 TFCI bits. In the case of two TFCI bits denoted b_0 and b_1 the TFCI word shall be { $b_0, b_1, b_0, b_1, b_0, b_1, b_0, b_1$ }.

4.3.1.2.2 Coding short TFCIs using bi-orthogonal codes

If the number of TFCI bits is in the range 3 to 5 the TFCI bits are encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 4-8.

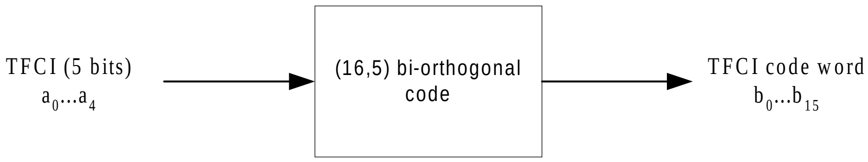

Figure 4-8: Channel coding of short length TFCI bits

The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 4.3.1-2 below.

Table 4.3.1-2: Basis sequences for (16,5) TFCI code

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$
0	1	1	0	0	0
1	1	0	1	0	0
2	1	1	1	0	0
3	1	0	0	1	0
4	1	1	0	1	0
5	1	0	1	1	0
6	1	1	1	1	0
7	1	0	0	0	1
8	1	1	0	0	1
9	1	0	1	0	1
10	1	1	1	0	1
11	1	0	0	1	1
12	1	1	0	1	1
13	1	0	1	1	1
14	1	1	1	1	1
15	1	0	0	0	0

For TFCI information bits a_0 , a_1 , a_2 , a_3 , a_4 (a_0 is LSB and a_4 is MSB), the), the output code word bits b_j are given by:

$$b_i = \sum_{n=0}^{4} (a_n \times M_{i,n}) \bmod 2$$

where i=0…15. N_{TFCI}=16.

4.3.1.3 Mapping of TFCI word

The mapping of the TFCI word to the TFCI bit positions in a timeslot shall be as follows.

Denote the number of bits in the TFCI word by N_{TFCI}, denote the code word bits by b_k where k=0… N_{TFCI}-1.

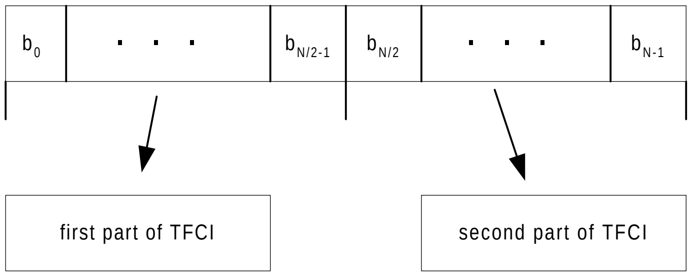

Figure 4-9: Mapping of TFCI word bits to timeslot

The locations of the first and second parts of the TFCI in the timeslot is defined in [7].

If the shortest transmission time interval of any constituent TrCH is at least 20 ms the successive TFCI words in the frames in the TTI shall be identical. If TFCI is transmitted on multiple timeslots in a frame each timeslot shall have the same TFCI word.

4.3.2 Coding of Paging Indicator (PI)

The PI is an identifier to instruct the UE whether there is a paging message for the groups of mobiles that are associated to the PI. The length L_{PI} of the PI is L_{PI}=2, L_{PI}=4 or L_{PI}=8 symbols. The coding of the PI is shown in table 4.3.3-1.

Table 4.3.3-1: Coding of the PI

Bits	PI	Content
All ‘0’	Not set	There is no necessity to receive PCH
All ‘1’	Set	There is necessity to receive PCH-

4.3.3 Coding of Transmit Power Control (TPC)

The TPC command is an identifier sent in uplink transmission only, to instruct the NodeB whether Tx power has to be increased or decreased. The length of the TPC command is one symbol. The coding of the TPC command is shown in table 4.3.3-1.

Table 4.3.4-1: Coding of the TPC

TPC	TPC Bits	Meaning
‘Down’	00	Decrease Tx Power
‘Up’	11	Increase Tx Power

Annex A (informative): Change history

Change history					
TSG RAN#	Version	CR	Tdoc RAN	New Version	Subject/Comment
RAN_05	-	-	RP-99592	3.0.0	Approved at TSG RAN #5 and placed under Change Control
RAN_06	3.0.0	001	RP-99694	3.1.0	Correction of rate matching parameters for repetition after 1st Interleaving in 25.222
RAN_06	3.0.0	002	RP-99694	3.1.0	Clarification of bit separation and collection
RAN_06	3.0.0	003	RP-99694	3.1.0	Changing the initial offset value for convolutional code rate matching
RAN_06	3.0.0	004	RP-99693	3.1.0	Editorial corrections to TS 25.222
RAN_06	3.0.0	007	RP-99694	3.1.0	Update of rate matching rule for TDD
RAN_06	3.0.0	009	RP-99694	3.1.0	Modified physical channel mapping scheme
RAN_06	3.0.0	013	RP-99694	3.1.0	Introduction of TFCI for S-CCPCH in TDD mode
RAN_06	3.0.0	015	RP-99694	3.1.0	TFCI coding and mapping in TDD
-	3.1.0	-	-	3.1.1	Change history was added by the editor

History

Document history		
V3.1.1	January 2000	Publication