TSG-RAN Meeting #8 **_RP-000272_**
Düsseldorf, Germany, 21-23 June 2000

Title: **Agreed CRs to TS 25.222**

Source: **TSG-RAN WG1**

Agenda item: **5.1.3**

No.	Doc #	Spec	CR	Rev	Subject	Cat	Current_v	New_v
1	R1-000464	25.222	030	-	Parity bit attachment to 0 size transport block	B	3.2.0	3.3.0
2	R1-000465	25.222	031	-	Correction of the mapping formula	F	3.2.0	3.3.0
3	R1-000513	25.222	034	-	Alignment of Multiplexing for TDD	F	3.2.0	3.3.0
4	R1-000716	25.222	036	2	Bit separation of the Turbo encoded data	D	3.2.0	3.3.0
5	R1-000751	25.222	038	2	Revision of code block segmentation description	D	3.2.0	3.3.0
6	R1-000743	25.222	039	-	Editorial corrections in channel coding section	F	3.2.0	3.3.0

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-00-0464*
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222	**CR**	**030**	Current Version:	3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN# 8	for approval	X	strategic		*(for SMG use only)*
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio	X	Core Network	

Source:	TSG RAN WG1	**Date:**	28, March 2000
Subject:	Parity bit attachment to 0 size transport block		
Work item:	TS 25.222		

Category:	F	Correction		**Release:**	Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature	X		Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification			Release 99	X
					Release 00	

Reason for change:	Alignment with FDD – the attachment of CRC to a transport block of size = 0 was approved for FDD
Clauses affected:	4.2.1.1

Other specs affected:	Other 3G core specifications	X	→ List of CRs:	
	Other GSM core specifications		→ List of CRs:	
	MS test specifications		→ List of CRs:	
	BSS test specifications		→ List of CRs:	
	O&M specifications		→ List of CRs:	

Other comments:	

help.doc

<--------- double-click here for help and instructions on how to create a CR.

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 $g_{CRC8}(D)$.

If no transport blocks are input to the CRC calculation ($M_i = 0$), no CRC attachment shall be done.

If transport blocks are input to the CRC calculation ($M_i \neq 0$) and the size of a transport block is zero ($A_i = 0$), CRC shall be attached, i.e. all parity bits equal to zero.

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:

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

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

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-000465*
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 031 Current Version: V3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG RAN#8 — for approval [X] — for information [ ] — strategic [ ] — non-strategic [ ] *(for SMG use only)*

list expected approval meeting # here ↑

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source:	TSG RAN WG1
Date:	March 28, 2000
Subject:	Correction of the mapping formula
Work item:	TS 25.222

Category: *(only one category shall be marked with an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	
Release 00	

Reason for change: This CR corrects an error in the pseudo code used for channel mapping scheme.

Clauses affected: 4.2.11.1

Other specs affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other GSM core specifications	[ ]	→ List of CRs:
MS test specifications	[ ]	→ List of CRs:
BSS test specifications	[ ]	→ List of CRs:
O&M specifications	[ ]	→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

4.2.11.1 Mapping scheme

Notation used in this section:

P_t: number of physical channels for timeslot t , P_t = 1..2 for uplink ; P_t = 1...16 for downlink

U_{tp}: capacity in bits for the physical channel p in timeslot t

$U_{t.}$: total number of bits to be assigned for timeslot t

bs_p: number of consecutive bits to assign per code

for downlink all $bs_p = 1$

for uplink if SF1 >= SF2 then $bs_1 = 1$; bs_2 = SF1/SF2 ;

if SF2 > SF1 then bs_1 = SF2/SF1; $bs_2 = 1$;

fb_p: number of already written bits for each code

pos: intermediate calculation variable

```
for  p=1  to P_t                                    --  reset number of already written bits for every physical channel
      fb_p = 0
end for

p = 1                                               -- start with PhCH #1
for k=1  to U_t.
     do while (fb_p == U_tp)                        -- physical channel filled up already ?
           ~~p = ((p + 1) mod (P_t +1))  + 1 ;~~
           p = (p mod P_t) + 1;
    end do

    if (p mod 2) == 0
       pos = U_tp - fb_p                            -- reverse order
    else
       pos = fb_p + 1                               -- forward order
    endif

    w_tp,pos = v_t,k                                -- assignment
    fb_p = fb_p + 1                                 -- Increment number of already written bits

    if (fb_p mod bs_p) == 0                         -- Conditional change to the next physical channel
         ~~p = ((p + 1) mod (P_t +1))  + 1;~~
```

p = (p mod P_t) + 1;

end if

end for

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-00-0513*

e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 034 Current Version: 3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN #8	for approval	X	strategic	*(for SMG use only)*
list expected approval meeting # here ↑		for information		non-strategic	

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: (U)SIM [ ] ME [ ] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 07.04.2000

Subject: Alignment of Multiplexing for TDD

Work item:

Category: *(only one category shall be marked with an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: Some modifications have been done in order to align the TDD specification mode with the FDD spec. . The case of TFCS reconfiguration where a transport channel is removed from the CCTrCH is now described. Some formulae have been revised in order to avoid ambiguities in the order of calculation.

Clauses affected: 4.2.4, 4.2.6, 4.2.7.1, 4.2.7.2, 4.2.12

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

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 subclause 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

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

where

$T_i = F_i * N_i$ and

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

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 F_i 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 F~~i~~$_i$ 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 to 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, $e_{i,k} = y_{i,n_ik}$ and $N_i = Y_i$.

4.2.7.1 Determination of rate matching parameters

The following relations, defined for all TFC j, are used when calculating the rate matching pattern:

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

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

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

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_{max} , 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 $U_{p,Sp}$, where p refers to the sequence number $1 \le p \le P_{max}$ of this physical channel in the allocation message, and the second index Sp indicates the spreading factor with the possible values {*16, 8, 4, 2, 1*}, respectively. For each physical channel an individual minimum spreading factor Sp_{min} is transmitted by means of the higher layer. Then, for N_{data} one of the following values in ascending order can be chosen:

$$\left\{U_{1,16},\ldots,U_{1,S1_{min}}, U_{1,S1_{min}}+U_{2,16},\ldots,U_{1,S1_{min}}+U_{2,S2_{min}},\ldots,U_{1,S1_{min}}+U_{2,S2_{min}}+\ldots+U_{P_{max},16},\ldots,U_{1,S1_{min}}+U_{2,S2_{min}}+\ldots+U_{P_{max},(SP_{max})_{min}}\right\}$$

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

SET1 = { N_{data} such that $\cancel{N_{data} - PL \cdot \sum_{x=1}^{I} \frac{RM_x}{\min_{1\le y\le I}\{RM_y\}} \cdot N_{x,j}}$ $\underline{\min_{1\le y\le I}\{RM_y\} \cdot N_{data} - PL \cdot \sum_{x=1}^{I} RM_x \cdot N_{x,j}}$ 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 subclause for all possible transport format combinations j and selected every radio frame.

If $\Delta N_{ij} = 0$ then the output data of the rate matching is the same as the input data and the rate matching algorithm of subclause 4.2.7.3 does not need to be executed.

Otherwise, the rate matching pattern is calculated with the algorithm described in subclause 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 subclauses 4.2.7.1.1 and 4.2.7.1.2.

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 and 5.

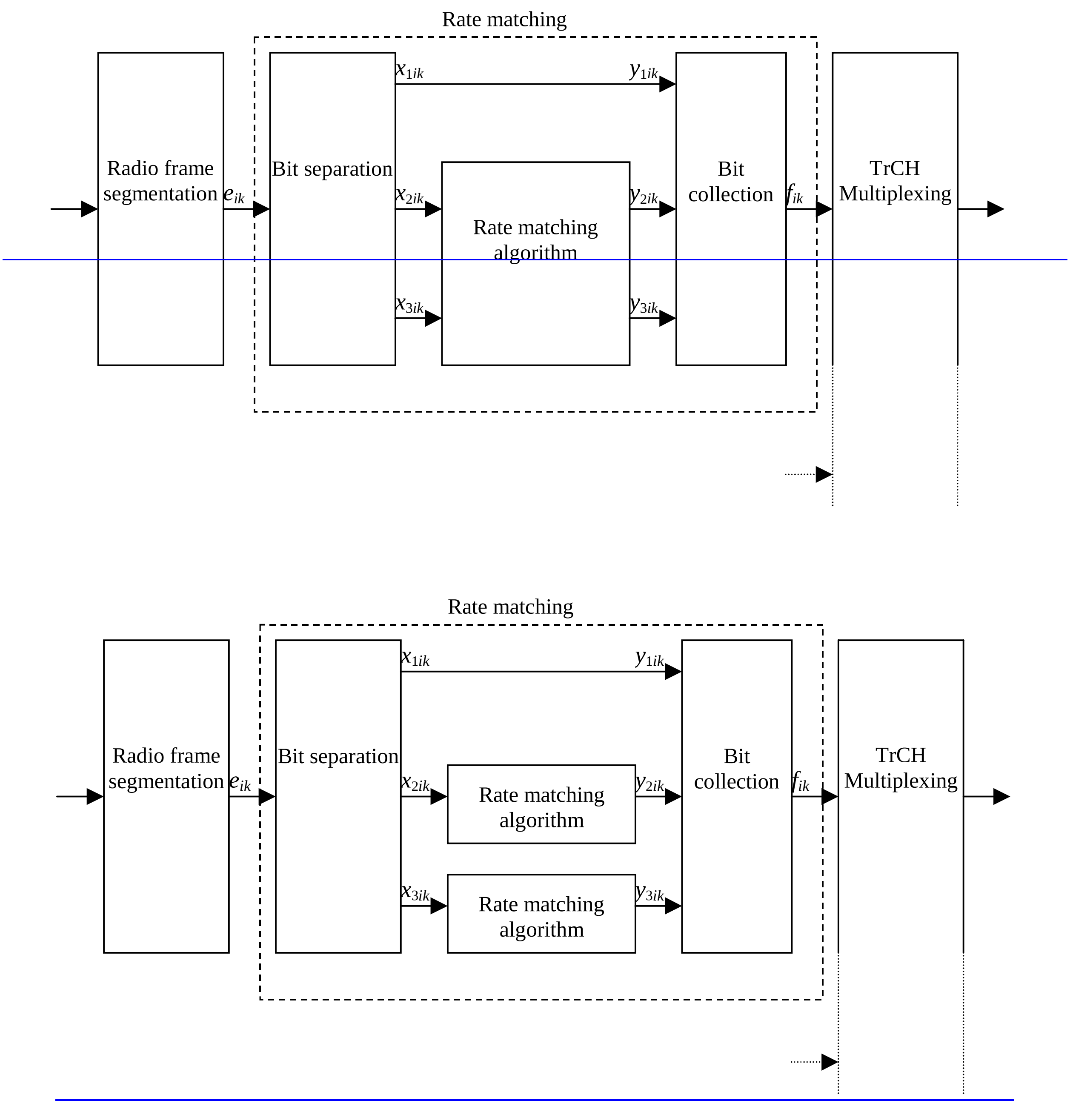


Figure 4: Puncturing of turbo encoded TrCHs

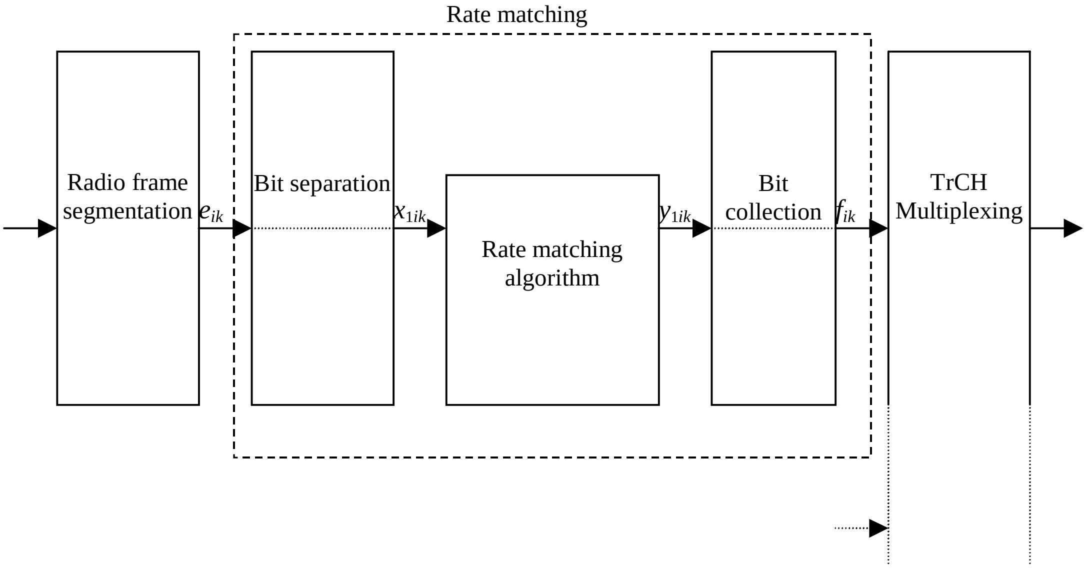


Figure 5: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition

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.

Table 4: 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 5: 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

4.2.12 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 shall have co-ordinated timings. When the TFCS of a CCTrCH is changed because ~~a~~ one or more transport channels ~~*i*~~ are~~is~~ added to the CCTrCH or reconfigured within the CCTrCH, or removed from the CCTrCH, the change may only be made at the ~~TTI of transport channel *i* may only~~ start ~~in~~ of a radio frame~~s~~ with CFN fulfilling the relation

   $CFN_{\cancel{i}} \bmod F_{max} = 0$,

   where F_{max} denotes the maximum number of radio frames within the transmission time intervals of all transport channels which are multiplexed into the same CCTrCH, including any transport channels *i* which ~~is~~are added, ~~or~~ reconfigured or have been removed, and $CFN_{\cancel{i}}$ denotes the connection frame number of the first radio frame of the changed CCTrCH.~~within the transmission time interval of transport channel *i*.~~

   After addition or reconfiguration of a transport channel *i* within a CCTrCH, the TTI of transport channel *i* may only start in radio frames with CFN fulfilling the relation

   $CFN_i \bmod F_i = 0$.

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

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

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.

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-00-0716*
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 036r2 Current Version: 3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG RAN #8	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: (U)SIM [ ] ME [ ] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 23, May 2000

Subject: Bit separation of the Turbo encoded data

Work item: TS 25.222

Category:				**Release:**	
	F	Correction		Phase 2	
	A	Corresponds to a correction in an earlier release		Release 96	
(only one category	B	Addition of feature		Release 97	
shall be marked	C	Functional modification of feature		Release 98	
with an X)	D	Editorial modification	X	Release 99	X
				Release 00	

Reason for change: The introductory description of bit separation and subsequent puncturing of Turbo coded data, specifically trellis termination bits is unclear..

Clauses affected: 4.2.7.2, 4.2.7.2.1

Other specs affected:			
	Other 3G core specifications		→ List of CRs: 25.212
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

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~~.~~, however systematic bits for trellis termination may be punctured. The systematic bits, first parity bits, and second parity bits in the bit sequence input to the rate matching block are therefore separated ~~from each other~~ into three sequences, one sequence containing all of the systematic bits and some systematic, first and second parity trellis termination bits; the second sequence containing all of the first parity bits and some systematic, first and second parity trellis termination bits and the third sequence containing all of the second parity bits and some systematic, first and second parity trellis termination bits. Puncturing is only applied to the second and third sequences. ~~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 and 5.

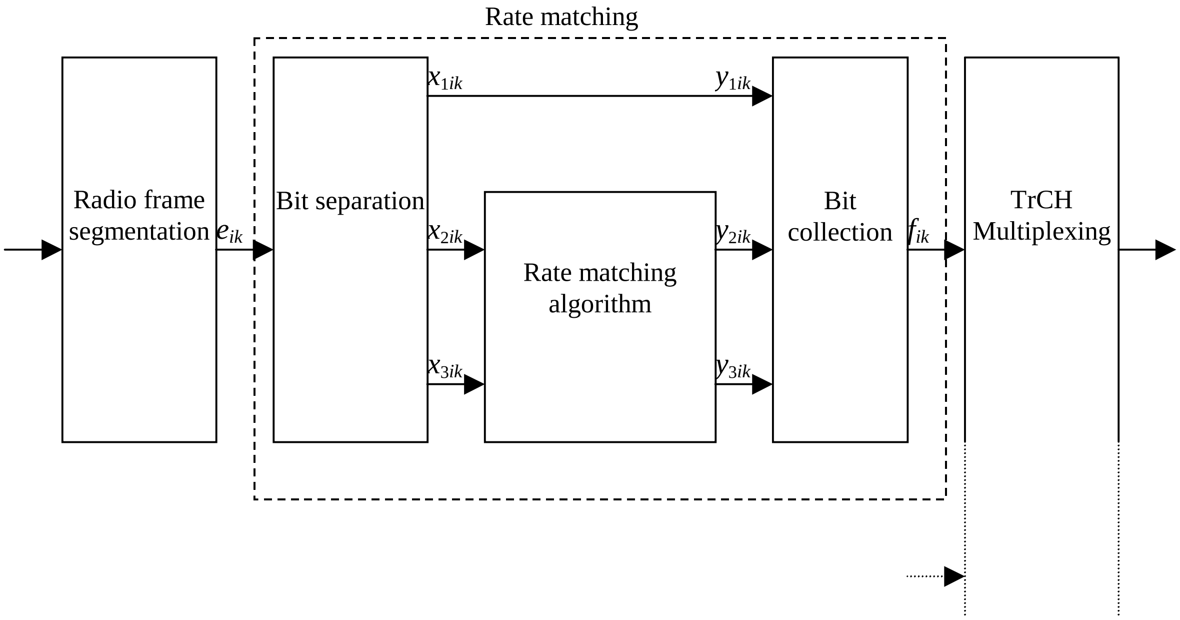


Figure 4: Puncturing of turbo encoded TrCHs

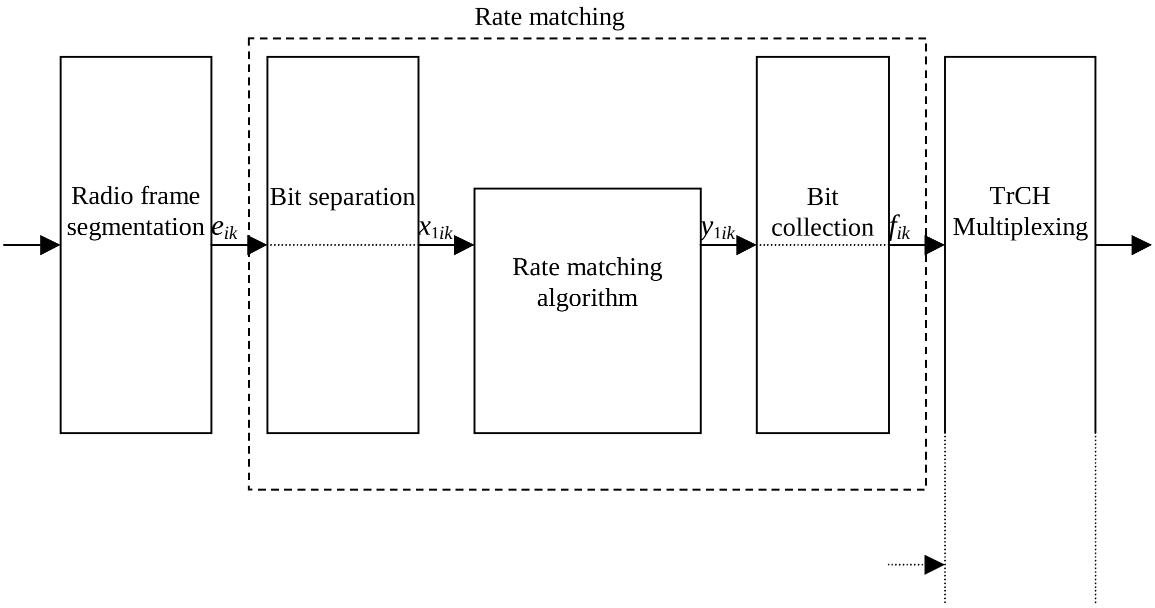


Figure 5: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition.

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. The sequence denoted as b=1 contains all of the systematic bits and some systematic, first and second parity trellis termination bits; the sequence denoted as b=2 contains all of the first parity bits and some systematic, first and second parity trellis termination bits; the sequence denoted as b=3 contains all of the second parity bits and some systematic, first and second parity trellis termination bits. The offsets α_b for ~~the~~ these sequences ~~systematic (b=1) and parity bits ($b\in\{2, 3\}$)~~ are listed in table 4.

Table 4 : 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 5 : 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 the three sequences defined in section 4.2.7.2~~systematic, first parity, or second parity bit.~~ The sequence denoted as b=1 contains all of the systematic bits and some systematic, first and second parity trellis termination bits; the sequence denoted as b=2 contains all of the first parity bits and some systematic, first and second parity trellis termination bits; the sequence denoted as b=3 contains all of the second parity bits and some systematic, first and second parity trellis termination bits. For all other cases *b* is defined to be 1. *X* 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:

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

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

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

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

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

$x_{1,i,k} = e_{i,k}$ $\quad k = 1, 2, 3, \ldots, X_i \quad X_i = N_i$

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-00-0751*
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 038r2 Current Version: 3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG RAN #8	for approval	X	strategic		(for SMG
list expected approval meeting # here ↑	for information		non-strategic		use only)

Form: CR cover sheet, version 2 for 3GPP and SMG The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio	X	Core Network	

Source: TSG RAN WG1 **Date:** 8.5.2000

Subject: Revision of Code Block Segmentation Description

Work item:

Category:				**Release:**		
	F	Correction			Phase 2	
	A	Corresponds to a correction in an earlier release			Release 96	
(only one category	B	Addition of feature			Release 97	
shall be marked	C	Functional modification of feature			Release 98	
with an X)	D	Editorial modification	X		Release 99	X
					Release 00	

Reason for change: The insertion of filler bits within the first block of code block segmentation does not depend on the conditions $X_i \le Z$ or $X_i > Z$. In that way the current specification is redundant. This is changed in the proposed modification. Additionally by reformulating some formulas a more precise description is given.

Clauses affected: Chapter 4.2.2.2

Other specs affected:			
	Other 3G core specifications		→ List of CRs:
	Other GSM core specifications		→ List of CRs:
	MS test specifications		→ List of CRs:
	BSS test specifications		→ List of CRs:
	O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

4.2.2.2 Code block segmentation

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 beginning of the first block. If turbo coding is selected and X_i < 40, filler bits are added to the beginning of the code 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:

if X_i < 40 and Turbo coding is used, then

K_i = 40

else

$K_i = \lceil X_i / C_i \rceil$

end if

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

~~If $X_i \leq Z$, then~~

$\cancel{o_{i1k} = 0 \quad k = 1, 2, \ldots, Y_i}$

for k = 1 to Y_i -- Insertion of filler bits

$o_{i1k} = 0$

end for

$\cancel{o_{i1k} = x_{i,(k-Y_i)} \quad k = Y_i+1, Y_i+2, \ldots, K_i}$

for k = Y_i+1 to K_i

$o_{i1k} = x_{i,(k-Y_i)}$

end for

~~end if~~

~~If $X_i > Z$, then~~

$\cancel{o_{i1k} = 0 \quad k = 1, 2, \ldots, Y_i}$

$\cancel{o_{i1k} = x_{i,(k-Y_i)} \quad k = Y_i+1, Y_i+2, \ldots, K_i}$

r = 2 -- Segmentation

while $r \leq C_i$

for k = 1 to K_i

$\cancel{o_{i2k} = x_{i,(k+K_i-Y_i)}}$ $o_{irk} = x_{i,(k+(r-1)\cdot K_i-Y_i)}$ ~~k = 1, 2, …, K_i~~

end for

$r = r+1$

end while $\cancel{o_{i3k} = x_{i,(k+2K_i-Y_i)}}$ ~~k = 1, 2, …, K_i~~

~~…~~

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

~~end if~~

3GPP TSG RAN Meeting #8
Düsseldorf, Germany, 21-23 June 2000

Document *R1-00-0743*
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

25.222 CR 039 Current Version: 3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: TSG RAN #8	for approval	X	Strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic		

Proposed change affects: (U)SIM [ ] ME [X] UTRAN / Radio [X] Core Network [ ]
(at least one should be marked with an X)

Source: TSG RAN WG1 **Date:** 24-May-2000

Subject: Editorial corrections in channel coding section

Work item:

Category: *(only one category shall be marked with an X)*

F	Correction	X	
A	Corresponds to a correction in an earlier release		
B	Addition of feature		
C	Functional modification of feature		
D	Editorial modification		

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
Release 00	

Reason for change: To give editorial corrections on descriptions of channel coding including Turbo code internal interleaver.

Clauses affected: 4.2.3 of TS25.222

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

end if

If $X_i > Z$, then

$o_{i1k} = 0 \quad k = 1, 2, \ldots, Y_i$

$o_{i1k} = x_{i,(k-Y_i)} \qquad k = Y_i+1, Y_i+2, \ldots, K_i$

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

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

…

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

end if

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 $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$, where Y_i is the number of encoded bits. 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 coding.

Usage of coding scheme and coding rate for the different types of TrCH is shown in table 1. The values of Y_i in connection with each coding scheme:

- convolutional coding with rate 1/2: $Y_i = 2*K_i + 16$; rate 1/3: $Y_i = 3*K_i + 24$;
- turbo coding with rate 1/3: $Y_i = 3*K_i + 12$;
- no coding: $Y_i = K_i$.

Table 1: Usage of channel coding scheme and coding rate

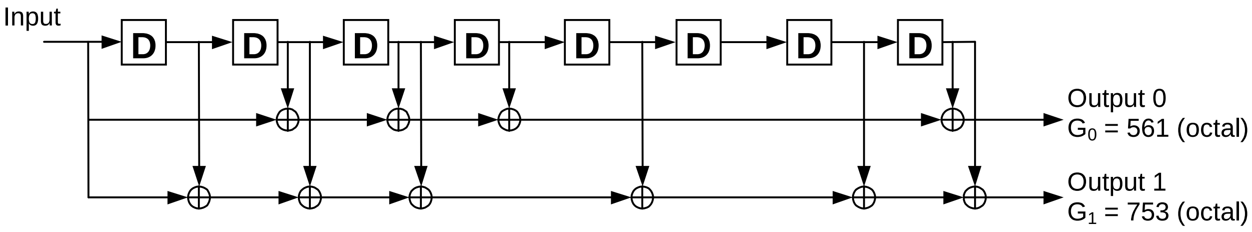
Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		
RACH		
DCH, DSCH, FACH, USCH		1/3, 1/2
	Turbo coding	1/3
	No coding	

~~If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.~~

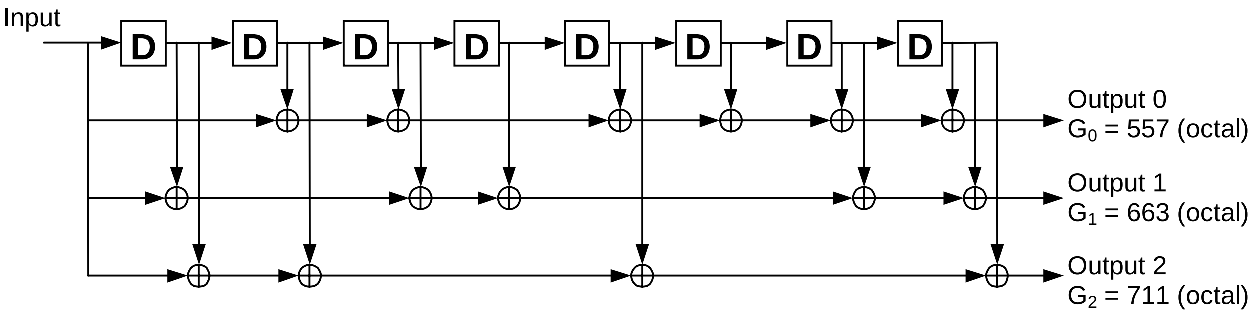
4.2.3.1 Convolutional coding

Convolutional codes with constraint length 9 and coding rates 1/3 and 1/2 are defined.
The configuration of the convolutional coder is presented in figure 2.
Output from the rate 1/3 convolutional coder shall be done in the order output0, output1, output2, output0, output1, output 2, output 0,…,output2. Output from the rate 1/2 convolutional coder shall be done in the order output 0, output 1, output 0, output 1, output 0, …, output 1.
8 tail bits with binary value 0 shall be added to the end of the code block before encoding.

The initial value of the shift register of the coder shall be "all 0" when starting to encode the input bits.



(a) Rate 1/2 convolutional coder



(b) Rate 1/3 convolutional coder

Figure 2: Rate 1/2 and rate 1/3 convolutional coders

4.2.3.2 Turbo coding

4.2.3.2.1 Turbo coder

The scheme of Turbo coder is a Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one Turbo code internal interleaver. The coding rate of Turbo coder is 1/3. The structure of Turbo coder is illustrated in figure 3.

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

$$\mathrm{G(D)}=\left[1, \frac{g_1(D)}{g_0(D)}\right],$$

where

$$g_0(D) = 1 + D^2 + D^3,$$

$$g_1(D) = 1 + D + D^3.$$

The initial value of the shift registers of the 8-state constituent encoders shall be all zeros when starting to encode the input bits.

Output from the Turbo coder is , Y'(0), X(1), Y(1), Y'(1), etc:

$$x_1, z_1, z'_1, x_2, z_2, z'_2, \ldots, x_K, z_K, z'_K,$$

where $x_1, x_2, \ldots, x_K$ are the bits input to the Turbo coder i.e. both first 8-state constituent encoder and Turbo code internal interleaver, and K is the number of bits, and $z_1, z_2, \ldots, z_K$ and $z'_1, z'_2, \ldots, z'_K$ are the bits output from first and second 8-state constituent encoders, respectively.

The bits output from Turbo code internal interleaver are denoted by $x'_1, x'_2, \ldots, x'_K$, and these bits are to be input to the second 8-state constituent encoder.

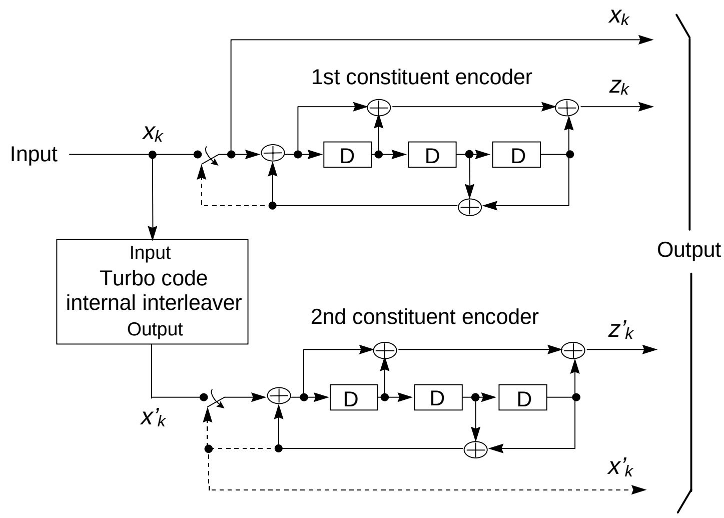


Figure 3: Structure of rate 1/3 Turbo coder (dotted lines apply for trellis termination only)

4.2.3.2.2 Trellis termination for Turbo coder

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

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

The transmitted bits for trellis termination shall then be:

$$x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x_{K+3}, z_{K+3}, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}, x'_{K+3}, z'_{K+3}).$$

4.2.3.2.3 Turbo code internal interleaver

The Turbo code internal interleaver consists of bits-input to a rectangular matrix, intra-row and inter-row permutations of the rectangular matrix, and bits-output from the rectangular matrix with pruning. The bits input to the Turbo code internal interleaver are denoted by $x_1, x_2, x_3, \ldots, x_K$, where K is the integer number of the bits and takes one value of 40 $\leq K \leq$ 5114. The relation between the bits input to the Turbo code internal interleaver and the bits input to the channel coding is defined by $x_k = o_{irk}$ and $K = K_i$.

The following subclause specific symbols are used in subclauses 4.2.3.2.3.1 to 4.2.3.4.3.3:

K	Number of bits input to Turbo code internal interleaver
R	Number of rows of rectangular matrix
C	Number of columns of rectangular matrix
p	Prime number
v	Primitive root
$s(i)$	Base sequence for intra-row permutation
q_j	Minimum prime integers
r_j	Permuted prime integers
$T(j)$	Inter-row permutation pattern
$U_j(i)$	Intra-row permutation pattern
i	Index of matrix
j	Index of matrix
k	Index of bit sequence

4.2.3.2.3.1 Bits-input to rectangular matrix

The bit sequence input to the Turbo code internal interleaver x_k is written into the rectangular matrix as follows.

(1) Determine the number of rows R of the rectangular matrix such that:

$$R = \begin{cases} 5, \text{if } (40 \le K \le 159) \\ 10, \text{if } ((160 \le K \le 200) \text{ or } (481 \le K \le 530)) \\ 20, \text{if } (K = \text{any other value}) \end{cases}$$

where the rows of rectangular matrix are numbered 0, 1, 2, …, R - 1 from top to bottom.

(2) Determine the number of columns C of rectangular matrix such that:
if (481 ≤ K ≤ 530) then

p = 53 and $C = p$.

else

Find minimum prime p such that,

$(p+1)-K/R \geq 0$,

and determine C such that

if $p-K/R \geq 0$) then

if ($p - 1 - K/R \geq 0$) then

$C = p - 1$.

else

$C = p$.

end if

else

C = p+1.

end if

end if

where the columns of rectangular matrix are numbered 0, 1, 2, …, C - 1 from left to right.

(3) Write the input bit sequence x_k into the $R \times C$ rectangular matrix row by row starting with bit x_1 in column 0 of: row 0:

$$\begin{bmatrix} x_1 & x_2 & x_3 & \cdots x_C \\ x_{(C+1)} & x_{(C+2)} & x_{(C+3)} & \cdots x_{2C} \\ \vdots & \vdots & \vdots & \cdots \vdots \\ x_{((R-1)C+1)} & x_{((R-1)C+2)} & x_{((R-1)C+3)} & \cdots x_{RC} \end{bmatrix}.$$

4.2.3.2.3.2 Intra-row and inter-row permutations

After the bits-input to the $R \times C$ rectangular matrix, the intra-row and inter-row permutations for the $R \times C$ rectangular matrix are performed by using the following algorithm:

(1) Select a primitive root v from table 2.

(2) Construct the base sequence $s(i)$ for intra-row permutation as:
$s(i) = [v \times s(i - 1)] \bmod p$, i =1,2,…($p$-2)., and $s(0) = 1$.

(3) Let $q_0 = 1$ be the first prime integer in $\{q_j\}$, and select the consecutive minimum prime integers $\{q_j\}$ (j=1,2,…R-1): such that

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

$q_j > 6$, and

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

where g.c.d. is greatest common divisor.

(4) Permute $\{q_j\}$ to make $\{r_j\}$ such that:

$r_{T(j)} = q_j$, j = 0, 1, …. R-1,

where ~~$T(j)$ indicates the original row position of the j-th permuted row, and~~ $T(j)$ (j = 0, 1, 2, …, R - 1) is the inter-row permutation pattern defined as the one of the following four kind of patterns: Pat_1, Pat_2, Pat_3 and Pat_4 depending on the number of input bits K.

$$\cancel{T(j)}\ \{T(0), T(1), T(2), ..., T(R-1)\} = \begin{cases} Pat_4 & \text{if } (40 \le K \le 159) \\ Pat_3 & \text{if } (160 \le K \le 200) \\ Pat_1 & \text{if } (201 \le K \le 480) \\ Pat_3 & \text{if } (481 \le K \le 530) \\ Pat_1 & \text{if } (531 \le K \le 2280) \\ Pat_2 & \text{if } (2281 \le K \le 2480) \\ Pat_1 & \text{if } (2481 \le K \le 3160) \\ Pat_2 & \text{if } (3161 \le K \le 3210) \\ Pat_1 & \text{if } (3211 \le K \le 5114) \end{cases},$$

where Pat_1, Pat_2, Pat_3 and Pat_4 have the following patterns respectively.

Pat_1: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}

Pat_2: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10}

Pat_3: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0}

Pat_4: {4, 3, 2, 1, 0}

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

$U_j(i) = s([i \times r_j] \bmod (p - 1))$, i =0,1,2,…, (p-2)., and $U_j(p-1) = 0$,

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

end if

if (C = p+1) then

$U_j(i) = s([i \times r_j] \bmod (p - 1))$, i =0,1,2,…, (p-2)., $U_j(p-1) = 0$, and $U_j(p) = p$,

where $U_j(i)$ is the input bit position of i-th output after the permutation of j-th row, and

if ($K = C \times R$) then

Exhange $U_{R-1}(p)$ with $U_{R-1}(0)$.

end if

end if

if ($C = p-1$) then

$U_j(i) = s([i \times r_j] \bmod (p - 1)) - 1$, i =0,1,2,…, (p-2),

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

end if
(6) Perform the inter-row permutation based on the pattern $T(j)$ (j = 0, 1, 2, …, R - 1),

where $T(j)$ is the original row position of the j-th permuted row.

Table 2: Table of prime *p* and associated primitive root *v*

p	v	p	v	p	v	p	v	p	v
7	3	47	5	101	2	157	5	223	3
11	2	53	2	103	5	163	2	227	2
13	2	59	2	107	2	167	5	229	6
17	3	61	2	109	6	173	2	233	3
19	2	67	2	113	3	179	2	239	7
23	5	71	7	127	3	181	2	241	7
29	2	73	5	131	2	191	19	251	6
31	3	79	3	137	3	193	5	257	3
37	2	83	2	139	2	197	2		
41	6	89	3	149	2	199	3		
43	3	97	5	151	6	211	2		

4.2.3.2.3.3 Bits-output from rectangular matrix with pruning

After intra-row and inter-row permutations, the bits of the permuted rectangular matrix are denoted by y'_k:

$$\begin{bmatrix} y'_1 & y'_{(R+1)} & y'_{(2R+1)} & \cdots y'_{((C-1)R+1)} \\ y'_2 & y'_{(R+2)} & y'_{(2R+2)} & \cdots y'_{((C-1)R+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y'_R & y'_{2R} & y'_{3R} & \cdots & y'_{CR} \end{bmatrix}.$$

The output of the Turbo code internal interleaver is the bit sequence read out column by column from the intra-row and inter-row permuted $R \times C$ matrix starting with bit y'_1 in row 0 of column 0 and ending with bit y'_{CR} in row R - 1 of column C - 1. 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 $k > K$ are removed from the output. The bits output from Turbo code internal interleaver are denoted by x'_1, x'_2, …, x'_K, where x'_1 corresponds to the bit y'_k with smallest index k after pruning, x'_2 to the bit y'_k with second smallest index k after pruning, and so on. The number of bits output from Turbo code internal interleaver is K and the total number of pruned bits is:

$R \times C - K$.

4.2.3.3 Concatenation of encoded blocks

After the channel coding for each code block, if C_i is greater than 1, the encoded blocks are serially concatenated so that the block with lowest index r is output first from the channel coding block, otherwise the encoded block is output from channel coding block as it is. 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:

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

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

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

…

$c_{ik} = y_{i,C_i,(k-(C_i-1)Y_i)}$ $\quad k = (C_i - 1)Y_i + 1, (C_i - 1)Y_i + 2, …, C_i\cancel{Y_i}\underline{Y_i}$

If no code blocks are input to the channel coding ($C_i = 0$), no bits shall be output from the channel coding, i.e. $E_i = 0$.