3GPP Meeting TSG RAN WG1#15
Berlin, Germany, 22-25 Aug 2000

Document R1-00-875
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**	**040**	Current Version: 3.3.0

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

For submission to:	RAN#9	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:	Siemens	**Date:**	21-Aug-2000
Subject:	Update of TS 25.222		
Work item:			

Category:	F	Correction	X	**Release:**	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			Release 99	X
					Release 00	X

Reason for change:	Tracking corrective changes in FDD. Specification language improved. Statistical notation clarified
Clauses affected:	4.2.5, 4.2.6, 4.2.7 and subsections, 4.2.9, 4.2.10 and subsections, 4.2.11 and subsections, 4.3 and subsections

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:	

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

4.2.5 1st interleaving

The 1st interleaving is a block interleaver with inter-column permutations. The input bit sequence to the 1st interleaver is denoted by $x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,X_i}$ ~~$x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$~~, 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 C1~~C_I~~ from table 3;

2) determine the number of rows R1~~R_I~~ defined as ~~R_I~~ R1 = X_i/C1~~C_I~~;

3) write the input bit sequence into the ~~R_I~~ R1 $\times$ C1~~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 $x_{i,(\mathrm{R1}\times\mathrm{C1})}$ ~~$x_{i,(R_I C_I)}$~~ in column ~~C_I~~ C1 of row ~~R_I~~ R1;

$$\begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & \ldots & x_{i,\mathrm{C1}} \\ x_{i,(\mathrm{C1}+1)} & x_{i,(\mathrm{C1}+2)} & x_{i,(\mathrm{C1}+3)} & \ldots & x_{i,(2\times\mathrm{C1})} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+1)} & x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+2)} & x_{i,((\mathrm{R1}-1)\times\mathrm{C1}+3)} & \ldots & x_{i,(\mathrm{R1}\times\mathrm{C1})} \end{bmatrix}$$

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

4) ~~p~~Perform the inter-column permutation based on the pattern $\left\langle \mathrm{P1}_{\mathrm{C1}}(j) \right\rangle_{j\in\{0,1,\ldots,\mathrm{C1}-1\}}$ ~~$\{P_I(j)\}$ (j=0,1, …, C-1)~~ shown in table 3, where ~~P_I~~ $\mathrm{P1}_{\mathrm{C1}}(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by $y_{i,k}$:

$$\begin{bmatrix} y_{i,1} & y_{i,(\mathrm{R1}+1)} & y_{i,(2\times\mathrm{R1}+1)} & \ldots & y_{i,((\mathrm{C1}-1)\times\mathrm{R1}+1)} \\ y_{i,2} & y_{i,(\mathrm{R1}+2)} & y_{i,(2\times\mathrm{R1}+2)} & \ldots & y_{i,((\mathrm{C1}-1)\times\mathrm{R1}+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{i,\mathrm{R1}} & y_{i,(2\times\mathrm{R1})} & y_{i,(3\times\mathrm{R1})} & \ldots & y_{i,(\mathrm{C1}\times\mathrm{R1})} \end{bmatrix} \cancel{\begin{bmatrix} y_{i1} & y_{i,(R_I+1)} & y_{i,(2R_I+1)} & \ldots & y_{i,((C_I-1)R_I+1)} \\ y_{i2} & y_{i,(R_I+2)} & y_{i,(2R_I+2)} & \ldots & y_{i,((C_I-1)R_I+2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ y_{iR_I} & y_{i,(2R_I)} & y_{i,(3R_I)} & \ldots & y_{i,(C_I R_I)} \end{bmatrix}}$$

5) ~~r~~Read the output bit sequence $y_{i1}, y_{i2}, y_{i3}, \ldots, y_{i,(\mathrm{C1}\times\mathrm{R1})}$ ~~$y_{i1}, y_{i2}, y_{i3}, \ldots, y_{i,(C_I R_I)}$~~ of the 1st interleaving column by column from the inter-column permuted R1~~R_I~~ $\times$ C1 ~~C_I~~ matrix. Bit $y_{i,1}$ corresponds to the first row of the first column and bit $y_{i,(\mathrm{R1}\times\mathrm{C1})}$ ~~$y_{i,(R_I C_I)}$~~ corresponds to row R1~~R_I~~ of column C1~~C_I~~.

The bits input to the 1st interleaving are denoted by $t_{i,1}, t_{i,2}, t_{i,3}, \ldots, t_{i,T_i}$ ~~$t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$~~, where i is the TrCH number and T_i the number of bits. Hence, ~~$x_{ik} = t_{ik}$~~ $x_{i,k} = t_{i,k}$ and $X_i = T_i$.

The bits output from the 1st interleaving are denoted by $d_{i,1}, d_{i,2}, d_{i,3}, \ldots, d_{i,T_i}$ ~~$d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$~~, and ~~$d_{ik} = y_{ik}$~~ $d_{i,k} = y_{i,k}$.

Table 3

TTI	Number of columns ~~C_1~~C1	Inter-column permutation patterns <P1C1(0), …, P1C1(C1-1)>
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 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 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.

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 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.

If no bits are input to the rate matching for all TrCHs within a CCTrCH, the rate matching shall output no bits for all TrCHs within the CCTrCH.

Notation used in subclause 4.2.7 and subclauses:

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

$\Delta N_{i,j}$ ~~ΔN_{ij}~~: If positive – number of bits to be repeated in each radio frame on TrCH i with transport format

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: number of physical channels used in the current frame.

P_{max}: maximum number of physical channels allocated for a CCTrCH.

U_p: Number of data bits in the physical channel p with p = 1...P.

I: Number of TrCHs in a CCTrCH.

Z_{ij}: 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~~$P1_F(n_i)$: The ~~inverse interleaving~~ column permutation function of the 1st interleaver, $P1_F(x)$ is the original position of column with number x after permutation. P1 is defined on table 3 of section 4.2.5 (note that $P1_F$ ~~the inverse interleaving function is identical to the interleaving function itself for the 1st interleaver~~self-inverse).

S[n]~~$S(n_i)$~~: The shift of the puncturing or repetition pattern for radio frame n_i when $n = P1_{F_i}(n_i)$.

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

$TFS(i)$~~$TFS(i)$~~: 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 subclause 4.2.7.3.

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

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

b: Indicates systematic and parity bits.

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

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

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

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:

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

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

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

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 $1£ p£ P_{max}$ 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:

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

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

SET1 = { N_{data} such that

$$\underline{\left(\min_{1\le y\le I}\left\{RM_y\right\}\right)\times N_{data}-PL\times\sum_{x=1}^{I}RM_x\times N_{x,j}}\ \cancel{\min_{1\le y\le I}\left\{RM_y\right\}\cdot N_{data}-PL\cdot\sum_{x=1}^{I}RM_x\cdot N_{x,j}}$$ is non negative }

$N_{data,\,j}$ = min SET1

The number of bits to be repeated or punctured, $\underline{\Delta N_{i,j}}$ ~~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 $\underline{\Delta N_{i,j}}$ ~~ΔN_{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.

4.2.7.1.1 Uncoded and convolutionally encoded TrCHs

a = 2

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

$X_i = N_{i,j}$

R = $\underline{\Delta N_{i,j} \bmod N_{i,j}}$ ~~ΔN_{ij} mod N_{ij}~~ -- note: in this context $\underline{\Delta N_{i,j} \bmod N_{i,j}}$ ~~ΔN_{ij} mod N_{ij}~~ is in the range of 0 to ~~N_{ij}~~ $\underline{N_{i,j}}$-1 i.e. -1 mod 10 = 9.

if $R \neq 0$ and 2$\underline{\times}$R ≤ ~~N_{ij}~~ $\underline{N_{i,j}}$

then q = $\lceil \underline{N_{i,j}}\ \cancel{N_{ij}}\ /\ \cancel{R}\ \underline{R} \rceil$

else

q = $\lceil \underline{N_{i,j}}\ \cancel{N_{ij}}\ /\ (\cancel{R}\underline{R} - \cancel{N_{ij}}\underline{N_{i,j}}) \rceil$

endif

NOTE 1: q is a signed quantity.

If q is even

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

NOTE 2: q' is not an integer, but a multiple of 1/8.

else

q' = q

endif

for x = 0 to ~~F_i~~ F_i-1

S[~~(I_F (~~ $\lfloor\lfloor x$ ~~×~~ $*q' \rfloor\rfloor$ mod F_i ~~F_i))~~] = ($\lfloor\lfloor x*q' \rfloor\rfloor$ div F_i ~~F_i~~)

end for

~~e_{ini} = (a·S(n_i)·|ΔN$_i$| + 1) mod (a·X_i)~~ e_{ini} = (a×S[P1$_{Fi}$(n_i)]×|ΔN_i | + 1) mod (a·$N_{i,j}$).

~~e_{plus} = a·X_i~~ e_{plus} = a × X_i

~~e_{minus} = a·|ΔN$_i$|~~ e_{minus} = a×|ΔN$_i$|

puncturing for ~~DN_i~~ 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. ~~D~~Δ$N_{i,j}$ >0, the parameters in subclause 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

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

If ΔN_i is calculated as 0 for *b*=2 or *b*=3, then the following procedure and the rate matching algorithm of subclause 4.2.7.3 don't need to be performed for the corresponding parity bit stream.

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

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

if(q ≤ 2)

for ~~x~~r=0 to F_i-1

S[~~I_F[~~(3×r~~x~~+b-1) mod F_i~~Fi]~~] = ~~x~~r mod 2;

end for

else

if ~~q~~q is even

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

NOTE: ~~q'~~q' is not an integer, but a multiple of 1/8.

else ~~q'~~q' = ~~q~~q

endif

for *x*=0 to F_i –1

r = $\lceil$ ~~x*~~x×q'$\rceil$ mod F_i ~~F_i~~;

S[~~I_F[~~(3×r+b-1) mod F_i ~~F_i]~~] = $\lceil$ ~~x*~~x×q'~~q'~~$\rceil$ div F_i ~~F_i~~;

endfor

endif

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

X_i is as above,

$\cancel{e_{ini}}\ e_{ini} = (a\cancel{\cdot}\times S[P1\ F_i\ (n_i)]\times\cancel{(n_i)}\ |\Delta N_i \cancel{N_i}| + X_i \cancel{X_i}) \bmod (a\cdot X_i)$, if $\cancel{e_{ini}}\ e_{ini} = 0$ then $\cancel{e_{ini} = a\cdot X_i}.\ e_{ini} = a\times X_i$.

$\cancel{e_{plus} = a\cdot X_i}\ e_{plus} = a\times X_i$

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

4.2.7.2 Bit separation and collection for rate matching

The systematic bits 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 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.

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.

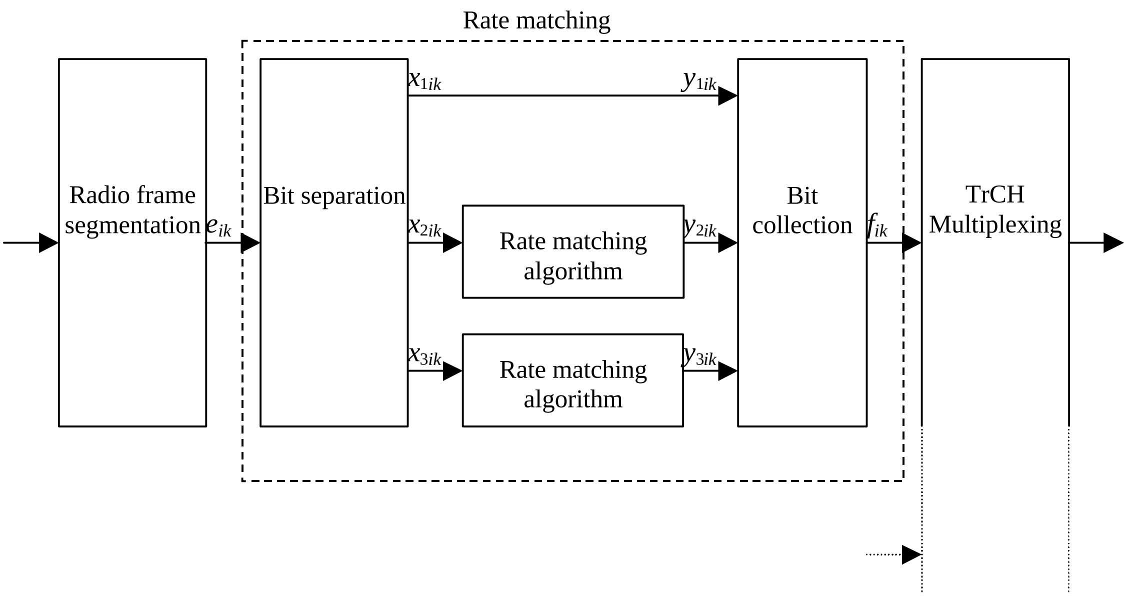

Figure 4: Puncturing of turbo encoded TrCHs

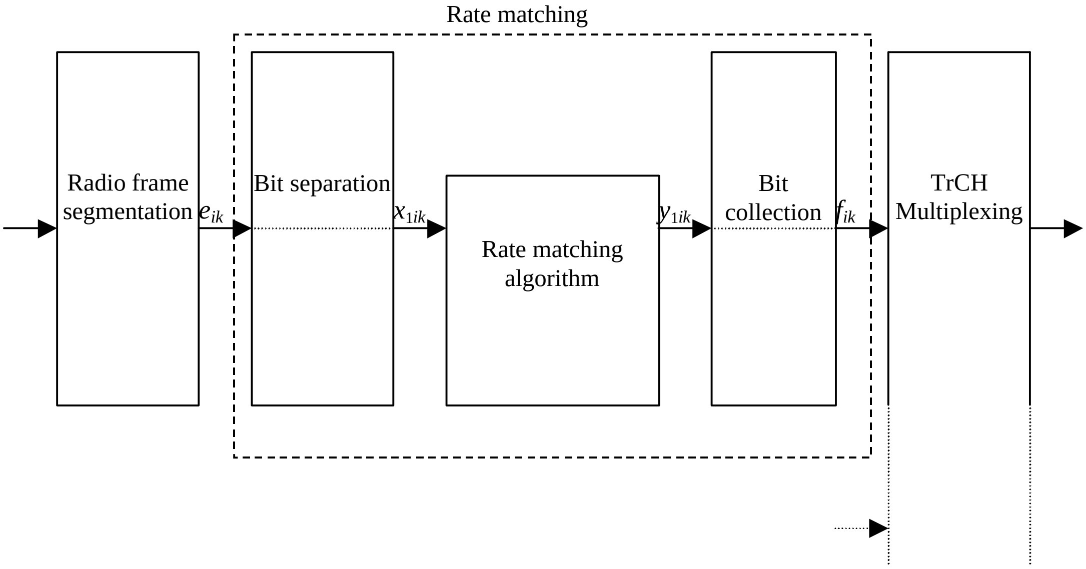

Figure 5: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition

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 a_b for these sequences are listed in table 4.

Table 4: TTI dependent offset needed for bit separation

TTI (ms)	a_1	a_2	a_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 b_{n_i} .

Table 5: Radio frame dependent offset needed for bit separation

TTI (ms)	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_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 $e_{i,1}, e_{i,2}, e_{i,3}, \ldots, e_{i,N_i}$ $\cancel{e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}}$, 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. $N_i=N_{i,j}$. The bits after separation are denoted by $x_{b,i,1}, x_{b,i,2}, x_{b,i,3}, \ldots, x_{b,i,X_i}$ $\cancel{x_{bi1}, x_{bi2}, x_{bi3}, \ldots, x_{biX_i}}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.2. 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*~~b~~=2 contains all of the first parity bits and some systematic, first and second parity trellis termination bits; the sequence denoted as ~~b~~*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_i is the number of bits in each separated bit sequence. The relation between $e_{i,k}$ and $x_{b,i,k}$ is given below.

For turbo encoded TrCHs with puncturing:

$x_{1,i,k} = e_{i,3(k-1)+1+(a_1+b_{n_i}) \bmod 3}$ $\quad k = 1, 2, 3, \ldots, X_i \quad X_i = \lfloor N_i / 3 \rfloor$

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

$x_{2,i,k} = e_{i,3(k-1)+1+(a_2+b_{n_i}) \bmod 3}$ $\quad k = 1, 2, 3, \ldots, X_i \quad X_i = \lfloor N_i / 3 \rfloor$

$x_{3,i,k} = e_{i,3(k-1)+1+(a_3+b_{n_i}) \bmod 3}$ $\quad k = 1, 2, 3, \ldots, X_i \quad X_i = \lfloor N_i / 3 \rfloor$

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

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

4.2.7.2.2 Bit collection

The bits $x_{b,i,k}$ are input to the rate matching algorithm described in subclause 4.2.7.3. The bits output from the rate matching algorithm are denoted $y_{b,i,1}, y_{b,i,2}, y_{b,i,3}, \ldots, y_{b,i,Y_i}$ $\cancel{y_{bi1}, y_{bi2}, y_{bi3}, \ldots, y_{biY_i}}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{b,i,1}, z_{b,i,2}, z_{b,i,3}, \ldots, z_{b,i,Y_i}$ $\cancel{z_{bi1}, z_{bi2}, z_{bi3}, \ldots, z_{biY_i}}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i,1}, f_{i,2}, f_{i,3}, \ldots, f_{i,V_i}$ $\cancel{f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}}$, where i is the TrCH number and V_i= $N_{i,j}+\Delta N_{i,j}$ $\cancel{N_{ij}+DN_{ij}}$. The relations between ~~y_{bik}, z_{bik}~~ $y_{b,i,k}$, $z_{b,i,k}$, and ~~f_{ik}~~ $f_{i,k}$ are given below.

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

$z_{i,3(k-1)+1+(a_1+b_{n_i}) \bmod 3} = y_{1,i,k}$ $\quad k = 1, 2, 3, \ldots, Y_I$

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

$z_{i,3(k-1)+1+(a_2+b_{n_i}) \bmod 3} = y_{2,i,k}$ $\quad k = 1, 2, 3, \ldots, Y_i$

$z_{i,3(k-1)+1+(a_3+b_{n_i}) \bmod 3} = y_{3,i,k}$ $\quad k = 1, 2, 3, \ldots, Y_i$

After the bit collection, bits $z_{i,k}$ with value d, where dÏ{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:

$z_{i,k} = y_{1,i,k}$ $\quad k = 1, 2, 3, \ldots, Y_i$

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 d, where dÏ{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 $x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,X_i}$ $\cancel{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}}$, where i is the TrCH and X_i is the parameter given in subclauses 4.2.7.1.1 and 4.2.7.1.2.

NOTE: 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 = ~~e_ini~~ e_ini     -- initial error between current and desired puncturing ratio
    m = 1                        -- index of current bit
    do while m <= X_i
        ~~e~~e = ~~e~~e – e_minus ~~e_minus~~     -- update error
        if e <= 0 then                            -- check if bit number m should be punctured
            set bit x_i,m to d where d Ï {0, 1}
            ~~e~~e = ~~e~~e + e_plus ~~e_plus~~   -- update error
        end if
        m = m + 1                                 -- next bit
    end do
else
    e = e_ini ~~e_ini~~          -- initial error between current and desired puncturing ratio
    m = 1                        -- index of current bit
    do while m <= X_i
        ~~e~~e = ~~e~~e – e_minus ~~e_minus~~     -- update error
        do while e <= 0                           -- check if bit number m should be repeated
            repeat bit x_i,m
            ~~e~~e = ~~e~~e + e_plus ~~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 $f_{i,1}, f_{i,2}, f_{i,3}, \ldots, f_{i,V_i}$ $\cancel{f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}}$, 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:

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

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

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

…

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

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 $u_{p,1}, u_{p,2}, u_{p,3}, \ldots, u_{p,U_p}$ $\cancel{u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU_p}}$, 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 $\cancel{u_{pk}}$ $u_{p,k}$ is given below.

Bits on first PhCH after physical channel segmentation:

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

Bits on second PhCH after physical channel segmentation:

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

…

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

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

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 2nd 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 with $S = U = \sum_p U_p$.

The relation between x_k and the bits $u_{p,k}$ in the respective physical channels is given below:

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

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

…

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

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

(1) Set the number of columns ~~C_2~~ C2 = 30. The columns are numbered 0, 1, 2, …, ~~C_2~~ C2-1 from left to right.

(2) Determine the number of rows ~~R_2~~ R2 by finding minimum integer ~~R_2~~ R2 such that:

U ≤ ~~R_2C_2~~R2 X C2.

(3) The bits input to the 2nd interleaving are written into the ~~R_2~~ R2 × ~~C_2~~ C2 rectangular matrix row by row.

$$\begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_{30} \\ x_{31} & x_{32} & x_{33} & \dots & x_{60} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{(R2-1)\times 30+1} & x_{(R2-1)\times 30+2} & x_{(R2-1)\times 30+3} & \dots & x_{R2\times 30} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_{30} \\ x_{31} & x_{32} & x_{33} & \dots & x_{60} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{(R_2-1)30+1} & x_{(R_2-1)30+2} & x_{(R_2-1)30+3} & \dots & x_{R_2\cdot 30} \end{bmatrix}$$

4) Perform the inter-column permutation based on the pattern {~~P_2~~P2(j)} (j = 0, 1, …, ~~C_2~~C2-1) that is shown in table 6, where ~~P_2~~P2(j) is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_k.

$$\begin{bmatrix} y_1 & y_{R2+1} & y_{2\times R2+1} & \dots y_{29\times R2+1} \\ y_2 & y_{R2+2} & y_{2\times R2+2} & \dots y_{29\times R2+2} \\ \vdots & \vdots & \vdots & \dots \quad \vdots \\ y_{R2} & y_{2\times R2} & y_{3\times R2} & \dots \ y_{30\times R2} \end{bmatrix} \begin{bmatrix} y_1 & y_{R_2+1} & y_{2R_2+1} & \dots y_{29R_2+1} \\ y_2 & y_{R_2+2} & y_{2R_2+2} & \dots y_{29R_2+2} \\ \vdots & \vdots & \vdots & \dots \quad \vdots \\ y_{R_2} & y_{2R_2} & y_{3R_2} & \dots \ y_{30R_2} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted ~~R_2~~ R2 × ~~C_2~~ C2 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 $v_1, v_2, \dots, v_U$, 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 $x_{t,1}, x_{t,2}, x_{t,3}, \dots, x_{t,U_t}$ ~~$x_{t1}, x_{t2}, x_{t3}, \dots, x_{tU_t}$~~, 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}~~ $x_{t,k}$ and ~~u_{tpk}~~ $u_{t,p,k}$ is given below with P_t refering to the number of physical channels within the respective timeslot:

~~$x_{tk} = u_{t1k}$~~ $x_{t,k} = u_{t,1,k}$ $k = 1, 2, \dots, U_{t1}$

~~$x_{t(k+U_{t1})} = u_{t2k}$~~ $x_{t,(k+U_{t1})} = u_{t,2,k}$ $k = 1, 2, \dots, U_{t2}$

…

~~$x_{t(k+U_{t1}+\dots+U_{t(P_t-1)})} = u_{tP_tk}$~~ $x_{t,(k+U_{t1}+\dots+U_{t(P_t-1)})} = u_{t,P_t,k}$ $k = 1, 2, \dots, U_{tP_t}$

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~~ C2 = 30. The columns are numbered 0, 1, 2, …, ~~C_2~~ C2 - 1 from left to right.

(2) Determine the number of rows ~~R_2~~ R2 by finding minimum integer ~~R_2~~ R2 such that:

U_t ≤ ~~R_2C_2~~R2×C2.

(3) The bits input to the 2nd interleaving are written into the ~~R_2~~ R2 × C2~~C_2~~ rectangular matrix row by row.

$$\begin{bmatrix} x_{t,1} & x_{t,2} & x_{t,3} & \cdots & x_{t,30} \\ x_{t,31} & x_{t,32} & x_{t,33} & \cdots & x_{t,60} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{t,((R2-1)\times 30+1)} & x_{t,((R2-1)\times 30+2)} & x_{t,((R2-1)\times 30+3)} & \cdots & x_{t,(R2\times 30)} \end{bmatrix}$$

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

(4) Perform the inter-column permutation based on the pattern ~~$\{P_2(j)\}$~~ P2(j) ~~(~~(j = 0, 1, …, ~~C_2~~C2-1~~)~~) that is shown in table 6, where ~~$P_2(j)$~~ P2(j) is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by $y_{t,k}$.

$$\begin{bmatrix} y_{t,1} & y_{t,(R2+1)} & y_{t,(2\times R2+1)} & \cdots y_{t,(29\times R2+1)} \\ y_{t,2} & y_{t,(R2+2)} & y_{t,(2\times R2+2)} & \cdots y_{t,(29\times R2+2)} \\ \vdots & \vdots & \vdots & \cdots \quad \vdots \\ y_{t,R2} & y_{t,(2\times R2)} & y_{t,(3\times R2)} & \cdots \; y_{t,(30\times R2)} \end{bmatrix} \begin{bmatrix} y_{t1} & y_{t,(R_2+1)} & y_{t,(2R_2+1)} & \cdots y_{t,(29R_2+1)} \\ y_{t2} & y_{t,(R_2+2)} & y_{t,(2R_2+2)} & \cdots y_{t,(29R_2+2)} \\ \vdots & \vdots & \vdots & \cdots \quad \vdots \\ y_{tR_2} & y_{t,(2R_2)} & y_{t,(3R_2)} & \cdots \; y_{t,(30R_2)} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted ~~R_2~~R2 $\times$ C2 ~~C_2~~ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits ~~y_{tk}~~ $y_{t,k}$ that corresponds to bits ~~x_{tk}~~ $x_{t,k}$ with $k>U_t$ are removed from the output. The bits after 2nd interleaving are denoted by $v_{t,1}, v_{t,2}, \ldots, v_{t,U_t}$ ~~$v_{t1}, v_{t2}, \ldots, v_{tU_t}$~~, where ~~$v_{t1}$~~ $v_{t,1}$ corresponds to the bit ~~y_{tk}~~ $y_{t,k}$ with smallest index k after pruning, ~~v_{t2}~~ $v_{t,2}$ to the bit ~~y_{tk}~~ $y_{t,k}$ with second smallest index k after pruning, and so on.

Table 6

Column number ~~C_2~~C2	Inter-column permutation pattern < P2(0), P2(1), …,P2(29) >
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 $w_{p,1}, w_{p,2}, \ldots, w_{p,U_p}$ ~~$w_{p1}, w_{p2}, \ldots, w_{pU_p}$~~, 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}~~ $w_{p,k}$ 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 of the bits $v_{(t),1}, v_{(t),2}, \ldots, v_{(t),U_{(t)}}$ ~~$v_{(t)1}, v_{(t)2}, \ldots, v_{(t)U_{(t)}}$~~ is performed like block interleaving, writing the bits into columns, 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 mapping scheme, as described in the following subclause, shall be applied individually for each timeslot t used in the current frame. Therefore, the bits $v_{t,1}, v_{t,2}, \ldots, v_{t,U_t}$ ~~$v_{t1}, v_{t2}, \ldots, v_{tU_t}$~~ are assigned to the bits of the physical channels $w_{t,1,1\ldots U_{t1}}, w_{t,2,1\ldots U_{t2}}, \ldots, w_{t,P_t,1\ldots U_{tP_t}}$ ~~$w_{t1,1\ldots U_{t1}}, w_{t2,1\ldots U_{t2}}, \ldots, w_{tP_t,1\ldots U_{tP_t}}$~~ in each timeslot.

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. Denote SF1 and SF2 the spreading factors used for code 1 and 2, respectively. For the number of consecutive bits to assign per code bs_k the following rule is applied:

if

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

else

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

end if

In the downlink case bs_p is 1 for all physical channels.

4.2.11.1 Mapping scheme

Notation used in this subclause:

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

$\underline{U_{t,p}}$~~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}~~$\underline{U_{t,p}}$) -- physical channel filled up already ?

p = (p mod P_t) + 1;

end do

if (p mod 2) == 0

pos = ~~U_{tp}~~ $\underline{U_{t,p}}$ - fb_p -- reverse order

else

pos = fb_p + 1 -- forward order

endif

~~$w_{tp,pos}$~~ $\underline{w_{t,p,pos}}$ = $v_{t,k}$ -- 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 mod P_t) + 1;

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~~its length. If there are 6-10 bits of TFCI the channel encoding is done as described in subclause 4.3.1.1. Also specific coding of less than 6 bits is possible as explained in subclause 4.3.1.2.

4.3.1.1 Coding of long TFCI lengths

The TFCI ~~bits are~~is encoded using a (32, 10) sub-code of the second order Reed-Muller code. The coding procedure is as shown in figure 6.

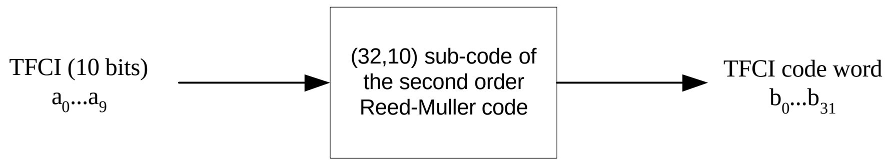

Figure 6: Channel coding of TFCI information bits

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 7.

Table 7: 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	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0
2	1	1	0	0	0	1	0	0	0	1
3	0	0	1	0	0	1	1	0	1	1
4	1	0	1	0	0	1	0	0	0	1
5	0	1	1	0	0	1	0	0	1	0
6	1	1	1	0	0	1	0	1	0	0
7	0	0	0	1	0	1	0	1	1	0
8	1	0	0	1	0	1	1	1	1	0
9	0	1	0	1	0	1	1	0	1	1
10	1	1	0	1	0	1	0	0	1	1
11	0	0	1	1	0	1	0	1	1	0
12	1	0	1	1	0	1	0	1	0	1
13	0	1	1	1	0	1	1	0	0	1
14	1	1	1	1	0	1	1	1	1	1
15	1	0	0	0	1	1	1	1	0	0
16	0	1	0	0	1	1	1	1	0	1
17	1	1	0	0	1	1	1	0	1	0
18	0	0	1	0	1	1	0	1	1	1
19	1	0	1	0	1	1	0	1	0	1
20	0	1	1	0	1	1	0	0	1	1
21	1	1	1	0	1	1	0	1	1	1
22	0	0	0	1	1	1	0	1	0	0
23	1	0	0	1	1	1	1	1	0	1
24	0	1	0	1	1	1	1	0	1	0
25	1	1	0	1	1	1	1	0	0	1
26	0	0	1	1	1	1	0	0	1	0
27	1	0	1	1	1	1	1	1	0	0
28	0	1	1	1	1	1	1	1	1	0
29	1	1	1	1	1	1	1	1	1	1
30	0	0	0	0	0	1	0	0	0	0
31	0	0	0	0	1	1	1	0	0	0

~~Let's define t~~The TFCI information bits ~~as~~ a_0 , a_1 , a_2 , a_3 , a_4 , a_5 , a_6 , a_7 , a_8 , a_9 (where a_0 is LSB and a_9 is MSB)~~. The TFCI information bits~~ shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

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*~~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.~~ Let's define the TFCI information bit(s) as b_0 (or b_0 and b_1).~~ The TFCI information bit(s) b_0 (or b_0 and b_1 where b_0 is the LSB) shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame. 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~~is encoded using a (16, 5) bi-orthogonal (or first order Reed-Muller) code. The coding procedure is as shown in figure 7.

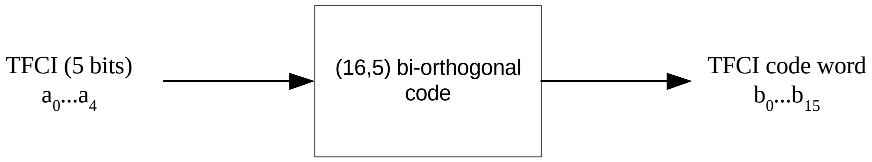

Figure 7: Channel coding of short length TFCI information bits

The code words of the (16,5) bi-orthogonal code are linear combinations of 5 basis sequences as defined in table 8.

Table 8: 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	0	0	0	1
1	0	1	0	0	1
2	1	1	0	0	1
3	0	0	1	0	1
4	1	0	1	0	1
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	0	1	1
8	1	0	0	1	1
9	0	1	0	1	1
10	1	1	0	1	1
11	0	0	1	1	1
12	1	0	1	1	1
13	0	1	1	1	1
14	1	1	1	1	1
15	0	0	0	0	1

~~Let's define t~~The TFCI information bits ~~as~~ a_0 , a_1 , a_2 , a_3 , a_4 (where a_0 is LSB and a_4 is MSB)~~. The TFCI information bits~~ shall correspond to the TFC index (expressed in unsigned binary form) defined by the RRC layer to reference the TFC of the CCTrCH in the associated DPCH radio frame.

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~~i~~=0,…,15. N_{TFCI}=16.