RP-020059

3GPP TSG-RAN Meeting #15 Jeju, Korea, 5 – 8, March, 2002

Title: Agreed CRs (R99 and Rel-4 Category A) for the removal of channel coding option of "no coding"

Source: TSG-RAN WG1

Agenda item: 7.1.2

No.	Spec	CR	Rev	R1 T-doc	Subject	Release	Cat	Workitem	V_old	V_new
1	25.201	009	1	R1-02-0495	Removal of channel coding option "no coding" for FDD and 3.84 Mcps TDD	R99	F	TEI	3.2.0	3.3.0
2	25.201	010	-	R1-02-0495	Removal of channel coding option "no coding" for FDD and 3.84 Mcps TDD	Rel-4	A	TEI	4.1.0	4.2.0
3	25.212	127	1	R1-02-0308	Removal of channel coding option "no coding" for FDD	R99	F	TEI	3.8.0	3.9.0
4	25.212	128	1	R1-02-0308	Removal of channel coding option "no coding" for FDD	Rel-4	Α	TEI	4.3.0	4.4.0
5	25.215	110	-	R1-02-0306	Removal of channel coding option "no coding" for FDD	R99	F	TEI	3.9.0	3.10.0
6	25.215	111	-	R1-02-0306	Removal of channel coding option "no coding" for FDD	Rel-4	Α	TEI	4.3.0	4.4.0
7	25.222	067	1	R1-02-0309	Removal of channel coding option "no coding" for 3.84 Mcps TDD	R99	F	TEI	3.7.0	3.8.0
8	25.222	068	1	R1-02-0309	Removal of channel coding option "no coding" for 3.84 Mcps TDD	Rel-4	A	TEI	4.2.0	4.3.0
9	25.225	044	-	R1-02-0307	Removal of channel coding option "no coding" 3.84 Mcps TDD	R99	F	TEI	3.9.0	3.10.0
10	25.225	045	-	R1-02-0307	Removal of channel coding option "no coding" for 3.84 Mcps TDD	Rel-4	A	TEI	4.3.0	4.4.0

_

¥	25.201 CR 009 # rev 1 ^{# Current version:} 3.2.0 [#]								
For HELP on using this form, see bottom of this page or look at the pop-up text over the # symbols.									
Proposed change affects: # (U)SIM ME/UE X Radio Access Network X Core Network									
Title: ೫	Removal of channel coding option "no coding" for FDD and 3.84 Mcps TDD								
Source: ೫	TSG RAN WG1								
Work item code: ೫	TEI Date: # 18.02.2002								
	Category: % F Release: % R99 Use one of the following categories: Ise one of the following releases: 2 (GSM Phase 2) A (corresponds to a correction in an earlier release) R96 (Release 1996) B (addition of feature), R97 (Release 1997) C (functional modification of feature) R98 (Release 1998) D (editorial modification) R99 (Release 1999) Detailed explanations of the above categories can be found in 3GPP TR 21.900. REL-5 (Release 5)								
Reason for change:	At the last joint AdHoc RAN1/RAN2 meeting it has been agreed to remove the channel coding option "no coding" for FDD and 3.84 Mcps TDD. For 1.28 Mcps TDD this option is still valid.								
Summary of change	# The channel coding option "no coding" has been removed in the IE "Type of channel coding".								
	Isolated Impact Analysis								
	This change affects the channel coding type. It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise.								
Consequences if not approved:	An option that is not used by any Radio Bearer would be a mandatory feature f all UEs.	or							
Clauses affected:	¥								
Other specs affected:	X Other core specifications ¥ 25.212, 25.222 Test specifications 0&M Specifications 5								
Other comments:	¥								

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked **#** contain pop-up help information about the field that they are closest to.

4.2.2 Channel coding and interleaving

For the channel coding in UTRA three-two options are supported:

- Convolutional coding.
- Turbo coding.
- ----No coding.

Channel coding selection is indicated by higher layers. In order to randomise transmission errors, bit interleaving is performed further.

3GPP TSG RAN Meeting #15

										CR-Form-v5		
ж	25.	201	CR	010	ж	rev	-	ж	Current ver	sion:	4.1.0	ж
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the $#$ symbols.												
Proposed change affects: # (U)SIM ME/UE X Radio Access Network X Core Network												
Title: #	Rer	noval	<mark>of chan</mark>	nel codi	ng optic	<mark>on "no c</mark>	coding	g" for	FDD and 3	<mark>.84 M</mark> o	cps TDD	
Source: %	TS	<mark>G RAN</mark>	WG1									
Work item code: ೫	TEI								Date: 3	€ <mark>18</mark> .	.02.2002	
Category: # A Release: # REL-4 Use one of the following categories: Use one of the following release. F (correction) 2 A (corresponds to a correction in an earlier release) R96 B (addition of feature), R97 C (functional modification of feature) R98 D (editorial modification) R99 D tetailed explanations of the above categories can REL-4 be found in 3GPP TR 21.900. REL-5						eases:						
Reason for change	. എ	At th	o last ic				2 ma	otina	it has been	agroe	d to romo	vo tho
Reason for change	<i>;,</i> oo	chan	nel cod		on "no c	oding"			ind 3.84 Mcr			
Summary of chang	le: #	chan Isolat This c It wo	nel cod ed Imp hange uld not	ling". act Ana affects th affect im	lysis ne chan nplemer	nel coo	ding t	ype. aving	peen remove like indicate	ed in t	he CR, it v	
Consequences if not approved:	Ħ	An o all U		at is not	used b	y any F	Radio	Bea	rer would be	a ma	ndatory fe	ature for
Clauses affected:	ж	4.2.2	2									
Other specs affected:	æ	Τe	est spec	e specifi cification ecificatio	s	Ħ	25	.212,	25.222			
Other comments:	ж											

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked # contain pop-up help information about the field that they are closest to.

R1-02-0495

4.2.2 Channel coding and interleaving

For the channel coding in UTRA three options are supported:

- Convolutional coding.
- Turbo coding.
- No coding (only 1.28 Mcps TDD).

Channel coding selection is indicated by higher layers. In order to randomise transmission errors, bit interleaving is performed further.

3GPP TSG RAN Meeting #15

										CR-Form-v5	
ж	25.	<mark>212</mark>	CR 12	27	жrev	1	ж	Current ver	sion:	3.8.0	ж
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the # symbols.											
Proposed change	Proposed change affects: # (U)SIM ME/UE X Radio Access Network X Core Network										
Title: ೫	Rer	noval	of channe	<mark>l coding o</mark>	ption "no	coding	g" fo	r FDD			
Source: #	TS	G RAN	WG1								
Work item code: #	TEI							Date: ¥	3 <mark>18</mark> .	.02.2002	
Category: % F Release: % R99 Use one of the following categories: Use one of the following release %											
Baacan far abang	. 49	At the				12 may	oting	it has been	ograa	d to romo	ve the
Reason for change	. т	chan	nel coding		o coding'			nd 3.84 Mcp			
Summary of chang	ye: #	The channel coding option "no coding" has been removed in the IE "Type of channel coding". Isolated Impact Analysis This change affects the channel coding type. It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise.									
Consequences if not approved:	ж	An o all U		is not use	d by any	Radio	Bear	rer would be	a ma	ndatory fe	ature for
Clauses affected:	ж	4.2.2	<mark>, 4.2.3, 4.</mark>	<mark>2.7.1, 4.2</mark> .	<mark>7.2, 4.2.7</mark>	.3, 4.2	2.7.4				
Other specs affected:	æ	Τe	her core s est specifie &M Specifie		ons a	£ 25.	201,	25.215, 25.2	222, 2	25.225	
Other comments:	ж										

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked **#** contain pop-up help information about the field that they are closest to.

R1-02-0308

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 Z, the maximum size of a code block in question, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional coding or $\frac{1}{7}$ 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 $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$ 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 $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$, where *i* is the TrCH number and $X_i = M_i B_i$. They are defined by the following relations:

$$\begin{aligned} x_{ik} &= b_{i1k} \quad k = 1, 2, \dots, B_i \\ x_{ik} &= b_{i,2,(k-B_i)} \quad k = B_i + 1, B_i + 2, \dots, 2B_i \\ x_{ik} &= b_{i,3,(k-2B_i)} \quad k = 2B_i + 1, 2B_i + 2, \dots, 3B_i \\ \dots \\ x_{ik} &= b_{i,M_i,(k-(M_i-1)B_i)} \quad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \dots, M_i B_i \end{aligned}$$

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 blocks. 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, for $C_i \neq 0$, are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and K_i is the number of bits per code block.

Number of code blocks:

$$_{i} = \left\lceil X_{i}/Z \right\rceil$$
 when $Z \neq unlimited$
$$C_{i} = \begin{cases} \left\lceil X_{i}/Z \right\rceil & \text{when } Z \neq unlimited \\ 0 & \text{when } Z = unlimited \text{ and } X_{i} = 0 \\ 1 & \text{when } Z = unlimited \text{ and } X_{i} \neq 0 \end{cases}$$

Number of bits in each code block (applicable for $C_i \neq 0$ only):

if $X_i < 40$ and Turbo coding is used, then

 $K_i = 40$ else $K_i = \langle X_i / C_i \rangle$ end if Number of filler bits: $Y_i = C_i K_i - X_i$ for k = 1 to Y_i -- Insertion of filler bits $o_{i1k} = 0$ end for for $k = Y_i + 1$ to K_i $o_{i1k} = x_{i,(k-Y_i)}$ end for r = 2-- Segmentation while $r \leq C_i$ for k = 1 to K_i $o_{irk} = x_{i,(k+(r-1)\cdot K_i - Y_i)I}$ end for r = r+1end while

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, 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 $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$, 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 TrCHs:

- convolutional coding;
- turbo 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: $Y_i = K_i$.

I

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

Table 1: Usage of channel coding scheme and coding rate

4.2.7.1.2 Determination of parameters needed for calculating the rate matching pattern

The number of bits to be repeated or punctured, $\Delta N_{i,j}$, within one radio frame for each TrCH *i* is calculated with equation 1 for all possible transport format combinations *j* and selected every radio frame. $N_{data,j}$ is given from subclause 4.2.7.1.1.

In a compressed radio frame, $N_{data,j}$ is replaced by $N_{data,j}^{cm}$ in Equation 1. $N_{data,j}^{cm}$ is given as follows:

In a radio frame compressed by higher layer scheduling, $N_{data,j}^{cm}$ is obtained by executing the algorithm in subclause 4.2.7.1.1 but with the number of bits in one radio frame of one PhCH reduced to $\frac{N_{tr}}{15}$ of the value in normal mode.

 N_{tr} is the number of transmitted slots in a compressed radio frame and is defined by the following relation:

$$N_{tr} = \begin{cases} 15 - TGL, \text{ if } N_{first} + TGL \le 15\\ N_{first}, \text{ in first frame if } N_{first} + TGL > 15\\ 30 - TGL - N_{first}, \text{ in second frame if } N_{first} + TGL > 15 \end{cases}$$

 N_{first} and TGL are defined in subclause 4.4.

In a radio frame compressed by spreading factor reduction, $N_{data,j}^{cm} = 2 \times (N_{data,j} - N_{TGL})$, where

$$N_{TGL} = \frac{15 - N_{tr}}{15} \times N_{data, j}$$

If $\Delta N_{i,j} = 0$ then the output data of the rate matching is the same as the input data and the rate matching algorithm of subclause 4.2.7.5 does not need to be executed.

If $\Delta N_{i,j} \neq 0$ the parameters listed in subclauses 4.2.7.1.2.1 and 4.2.7.1.2.2 shall be used for determining e_{ini} , e_{plus} , and e_{minus} (regardless if the radio frame is compressed or not).

4.2.7.1.2.1 Uncoded and cConvolutionally encoded TrCHs

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

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

then
$$\mathbf{q} = \begin{bmatrix} N_{i,i} / R \end{bmatrix}$$

else

$$\mathbf{q} = \left\lceil N_{i,i} / (R - N_{i,i}) \right\rceil$$

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 $F_i - 1$

$$\mathbf{S}[| \mathbf{x} \times \mathbf{q'} \mathbf{j}| \mod F_i] = (| \mathbf{x} \times \mathbf{q'} \mathbf{j}| \dim F_i)$$

end for

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

For each radio frame, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5, where :

$$X_{i} = N_{i,j}, \text{ and}$$

$$e_{ini} = (a \times S[P1_{Fi}(n_{i})] \times |\Delta N_{i}| + 1) \mod (a \cdot N_{ij}).$$

$$e_{plus} = a \times N_{i,j}$$

$$e_{minus} = a \times |\Delta N_{i}|$$

puncturing for $\Delta N < 0$, repetition otherwise.

4.2.7.2.1.3 Determination of rate matching parameters for uncoded and convolutionally encoded TrCHs

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

For compressed mode by puncturing, ΔN_i is defined as: $\Delta N_i = \Delta N_{i,\max}^{TTI,cm,m}$, instead of the previous relation.

$$a=2$$

$$N_{max} = \max_{l \in TFS(i)} N_{il}^{TTI}$$

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5. The following parameters are used as input:

$$X_{i} = N_{il}^{TTI}$$

$$e_{ini} = 1$$

$$e_{plus} = a \times N_{max}$$

$$e_{\min us} = a \times |\Delta N_{i}|$$

Puncturing if $\Delta N_i < 0$, repetition otherwise. The values of $\Delta N_{i,l}^{TTI}$ may be computed by counting repetitions or puncturing when the algorithm of subclause 4.2.7.5 is run. The resulting values of $\Delta N_{i,l}^{TTI}$ can be represented with following expression.

$$\Delta N_{i,l}^{TTI} = \left[\frac{|\Delta N_i| \times X_i}{N_{max}}\right] \times \operatorname{sgn}(\Delta N_i)$$

For compressed mode by puncturing, the above formula produces $\Delta N_{i,l}^{TTI,m}$ instead of $\Delta N_{i,l}^{TTI}$.

4.2.7.2.1.4 Determination of rate matching parameters for Turbo encoded TrCHs

If repetition is to be performed on turbo encoded TrCHs, i.e. $\Delta N_{i,max} > 0$, the parameters in subclause 4.2.7.2.1.3 are used.

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

$$a=2$$
 when $b=2$
 $a=1$ when $b=3$

The bits indicated by b=1 shall not be punctured.

$$\Delta N_i^b = \begin{cases} \left\lfloor \Delta N_{i,max} / 2 \right\rfloor, & \text{for } b = 2\\ \left\lceil \Delta N_{i,max} / 2 \right\rceil, & \text{for } b = 3 \end{cases}$$

In Compressed Mode by puncturing, the following relations are used instead of the previous ones:

$$\Delta N_{i}^{b} = \left[\Delta N_{i,\max}^{TTI,cm,m} / 2 \right], \text{ for } b=2$$
$$\Delta N_{i}^{b} = \left[\Delta N_{i,\max}^{TTI,cm,m} / 2 \right], \text{ for } b=3$$

$$N_{max} = \max_{l \in TFS(i)} (N_{il}^{TTI} / 3)$$

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in subcaluse 4.2.7.5. The following parameters are used as input:

$$X_{i} = N_{il}^{TTI} / 3$$
$$e_{ini} = N_{max}$$
$$e_{plus} = a \times N_{max}$$
$$e_{\min us} = a \times \left| \Delta N_{i}^{b} \right|$$

The values of $\Delta N_{i,l}^{TTI}$ may be computed by counting puncturing when the algorithm of subclause 4.2.7.5 is run. The resulting values of $\Delta N_{i,l}^{TTI}$ can be represented with following expression.

$$\Delta N_{i,l}^{TTI} = -\left\lfloor \frac{\left| \Delta N_i^2 \right| \times X_i}{N_{max}} + 0.5 \right\rfloor - \left\lfloor \frac{\left| \Delta N_i^3 \right| \times X_i}{N_{max}} \right\rfloor$$

In the above equation, the first term of the right hand side represents the amount of puncturing for b=2 and the second term represents the amount of puncturing for b=3.

For compressed mode by puncturing, the above formula produces $\Delta N_{i,l}^{TTI,m}$ instead of $\Delta N_{i,l}^{TTI}$.

4.2.7.2.2.2 Determination of rate matching parameters for uncoded and convolutionally encoded TrCHs

$$\Delta N_i = \Delta N_{il}^{TTI}$$
$$a=2$$

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5. The following parameters are used as input:

$$X_{i} = N_{il}^{TTI}$$

$$e_{ini} = 1$$

$$e_{plus} = a \times N_{il}^{TTI}$$

$$e_{\min us} = a \times |\Delta N_{i}|$$

puncturing for $\Delta N_i < 0$, repetition otherwise.

4.2.7.3 Bit separation and collection in uplink

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits 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.

The first sequence contains:

- All of the systematic bits that are from turbo encoded TrCHs.
- From 0 to 2 first and/or second parity bits that are from turbo encoded TrCHs. These bits come into the first sequence when the total number of bits in a block after radio frame segmentation is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second and third sequences shall be of equal length, whereas the first sequence can contain from 0 to 2 more bits. Puncturing is applied only 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 5 and 6.

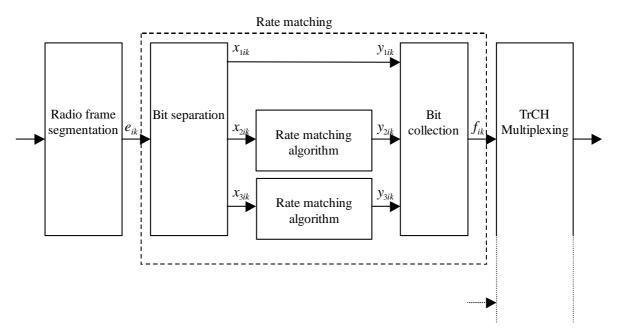


Figure 5: Puncturing of turbo encoded TrCHs in uplink

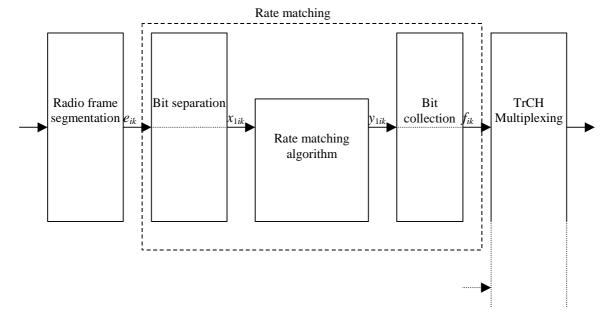


Figure 6: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition in uplink

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. *b* indicates the three sequences defined in this section, with *b*=1 indicating the first sequence, *b* = 2 the second one, and *b* = 3 the third one. The offsets α_b for these sequences are listed in table 5.

TTI (ms)	α1	α2	α3
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 .

TTI (ms)	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
10	0	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

Table 6: Radio frame dependent offset needed for bit separation

4.2.7.3.1 Bit separation

The bits input to the rate matching are denoted by $e_{i1}, e_{i2}, e_{i3}, \dots, e_{iN_i}$, 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 $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.3, with *b*=1 indicating the first sequence, and so forth. 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:

 $x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_n) \mod 3} \qquad k = 1, 2, 3, \dots, X_i \qquad X_i = \lfloor N_i / 3 \rfloor$

$$\begin{aligned} x_{1,i,\lfloor N_i/3 \rfloor+k} &= e_{i,3\lfloor N_i/3 \rfloor+k} & k = 1, ..., N_i \text{ mod } 3 & \text{Note: When } (N_i \text{ mod } 3) = 0 \text{ this row is not needed.} \\ x_{2,i,k} &= e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \\ x_{3,i,k} &= e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \end{aligned}$$

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

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

4.2.7.3.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in subclause 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \dots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i1}, f_{i2}, f_{i3}, \dots, f_{iV_i}$, 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 $(Y_i=X_i)$:

 $\begin{aligned} z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \mod 3} &= y_{1,i,k} & k = 1, 2, 3, ..., Y_i \\ z_{i,3\lfloor N_i/3 \rfloor + k} &= y_{1,i,\lfloor N_i/3 \rfloor + k} & k = 1, ..., N_i \mod 3 & \text{Note: When } (N_i \mod 3) = 0 \text{ this row is not needed.} \\ z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \mod 3} &= y_{2,i,k} & k = 1, 2, 3, ..., Y_i \\ z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \mod 3} &= y_{3,i,k} & k = 1, 2, 3, ..., Y_i \end{aligned}$

After the bit collection, bits $z_{i,k}$ with value δ , where $\delta \not\in \{0, 1\}$, 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:

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

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 $\delta \not\in \{0, 1\}$, 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.

4.2.7.4 Bit separation and collection in downlink

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits 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 of equal lengths.

The first sequence contains :

- All of the systematic bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

Puncturing is applied only 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 7 and 8.

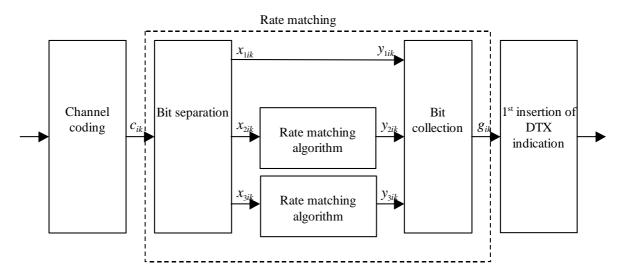


Figure 7: Puncturing of turbo encoded TrCHs in downlink

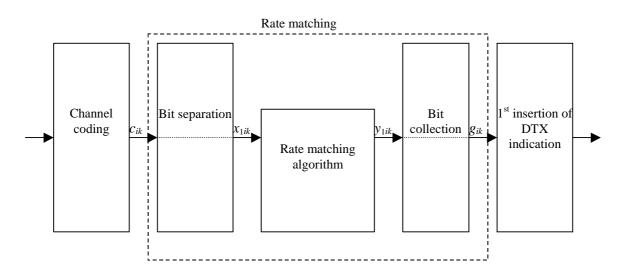


Figure 8: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition in downlink

4.2.7.4.1 Bit separation

The bits input to the rate matching are denoted by $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where *i* is the TrCH number and E_i is the number of bits input to the rate matching block. Note that E_i is a multiple of 3 for turbo encoded TrCHs and that the transport format *l* for simplicity has been left out in the bit numbering, i.e. $E_i = N_{il}^{TTI}$. The bits after separation are

denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.4, with b=1 indicating the first sequence, and so forth. 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 c_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

$x_{1,i,k} = c_{i,3(k-1)+1}$	$k = 1, 2, 3,, X_i$	$X_i = E_i/3$
$x_{2,i,k} = c_{i,3(k-1)+2}$	$k = 1, 2, 3,, X_i$	$X_i = E_i/3$
$x_{3,i,k} = c_{i,3(k-1)+3}$	$k = 1, 2, 3,, X_i$	$X_i = E_i / 3$

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

$$x_{1,i,k} = c_{i,k}$$
 $k = 1, 2, 3, ..., X_i$ $X_i = E_i$

4.2.7.4.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in subclause 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \dots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $g_{i1}, g_{i2}, g_{i3}, \dots, g_{iG_i}$, where *i* is the TrCH number and $G_i = N_{il}^{TTI} + \Delta N_{il}^{TTI}$. The relations between y_{bik}, z_{bik} , and g_{ik} are given below.

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

$z_{i,3(k-1)+1} = y_{1,i,k}$	$k = 1, 2, 3,, Y_i$
$z_{i,3(k-1)+2} = y_{2,i,k}$	$k = 1, 2, 3,, Y_i$
$z_{i,3(k-1)+3} = y_{3,i,k}$	$k = 1, 2, 3,, Y_i$

After the bit collection, bits $z_{i,k}$ with value δ , where $\delta \not\in \{0, 1\}$, are removed from the bit sequence. Bit $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index *k* after puncturing, bit $g_{i,2}$ 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, $g_{i,k}=z_{i,k}$ and $Y_i=G_i$.

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 $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index *k* after puncturing, bit $g_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index *k* after puncturing, and so on.

CR-Form-v5									
ж	25.212 CR 128 # rev 1 ^{# Current version:} 4.3.0 [#]								
For HELP on using this form, see bottom of this page or look at the pop-up text over the # symbols.									
Proposed change affects: # (U)SIM ME/UE X Radio Access Network X Core Network									
Title: ೫	Removal of channel coding option "no coding" for FDD								
Source: ೫	TSG RAN WG1								
Work item code: #	TEI Date: # 18.02.2002								
Category: % A Release: % REL-4 Use one of the following categories: Use one of the following releases: F (correction) 2 (GSM Phase 2) A (corresponds to a correction in an earlier release) R96 (Release 1996) B (addition of feature), R97 (Release 1997) C (functional modification of feature) R98 (Release 1998) D (editorial modification) R99 (Release 1999) Detailed explanations of the above categories can REL-4 (Release 4) be found in 3GPP TR 21.900. REL-5 (Release 5)									
Summary of change	 channel coding option "no coding" for FDD and 3.84 Mcps TDD. For 1.28 Mcps TDD this option is still valid. The channel coding option "no coding" has been removed in the IE "Type of channel coding". Isolated Impact Analysis This change affects the channel coding type. It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise. 								
Consequences if not approved:	An option that is not used by any Radio Bearer would be a mandatory feature for all UEs.								
Clauses affected:	# 4.2.2, 4.2.3, 4.2.7.1, 4.2.7.2, 4.2.7.3, 4.2.7.4								
Other specs affected:	X Other core specifications % 25.201, 25.215, 25.222, 25.225 Test specifications 0&M Specifications								
Other comments:	¥								

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked **#** contain pop-up help information about the field that they are closest to.

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 Z, the maximum size of a code block in question, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depends on whether convolutional coding or $\frac{1}{7}$ 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 $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$ 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 $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$, where *i* is the TrCH number and $X_i = M_i B_i$. They are defined by the following relations:

$$\begin{aligned} x_{ik} &= b_{i1k} \quad k = 1, 2, \dots, B_i \\ x_{ik} &= b_{i,2,(k-B_i)} \quad k = B_i + 1, B_i + 2, \dots, 2B_i \\ x_{ik} &= b_{i,3,(k-2B_i)} \quad k = 2B_i + 1, 2B_i + 2, \dots, 3B_i \\ \dots \\ x_{ik} &= b_{i,M_i,(k-(M_i-1)B_i)} \quad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \dots, M_i B_i \end{aligned}$$

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, for $C_i \neq 0$, are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and K_i is the number of bits per code block.

Number of code blocks:

$$_{i} = \left\lceil X_{i}/Z \right\rceil$$
 when $Z \neq unlimited$
$$C_{i} = \begin{cases} \left\lceil X_{i}/Z \right\rceil & \text{when } Z \neq unlimited \\ 0 & \text{when } Z = unlimited \text{ and } X_{i} \neq 0 \\ 1 & \text{when } Z = unlimited \text{ and } X_{i} \neq 0 \end{cases}$$

Number of bits in each code block (applicable for $C_i \neq 0$ only):

if $X_i < 40$ and Turbo coding is used, then

$$K_i = 40$$

else

```
K_i = \langle X_i / C_i \rangle
    end if
Number of filler bits: Y_i = C_i K_i - X_i
for k = 1 to Y_i
                                    -- Insertion of filler bits
    o_{i1k} = 0
end for
for k = Y_i + 1 to K_i
    o_{i1k} = x_{i,(k-Y_i)}
end for
r = 2
                                             -- Segmentation
while r \leq C_i
    for k = 1 to K_i
         o_{irk} = x_{i,(k+(r-1)\cdot K_i - Y_i)}
    end for
    r = r + 1
end while
```

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, 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 $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$, 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 TrCHs:

- 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: $Y_i = 3 * K_i + 12$;

I

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

Table 1: Usage of channel coding scheme and coding rate

4.2.7.1.2 Determination of parameters needed for calculating the rate matching pattern

The number of bits to be repeated or punctured, $\Delta N_{i,j}$, within one radio frame for each TrCH *i* is calculated with equation 1 for all possible transport format combinations *j* and selected every radio frame. $N_{data,j}$ is given from subclause 4.2.7.1.1.

In a compressed radio frame, $N_{data,j}$ is replaced by $N_{data,j}^{cm}$ in Equation 1. $N_{data,j}^{cm}$ is given as follows:

In a radio frame compressed by higher layer scheduling, $N_{data,j}^{cm}$ is obtained by executing the algorithm in subclause 4.2.7.1.1 but with the number of bits in one radio frame of one PhCH reduced to $\frac{N_{tr}}{15}$ of the value in normal mode.

 N_{tr} is the number of transmitted slots in a compressed radio frame and is defined by the following relation:

$$N_{tr} = \begin{cases} 15 - TGL, \text{ if } N_{first} + TGL \le 15 \\ N_{first}, \text{ in first frame if } N_{first} + TGL > 15 \\ 30 - TGL - N_{first}, \text{ in second frame if } N_{first} + TGL > 15 \end{cases}$$

 N_{first} and TGL are defined in subclause 4.4.

In a radio frame compressed by spreading factor reduction, $N_{data,j}^{cm} = 2 \times (N_{data,j} - N_{TGL})$, where

$$N_{TGL} = \frac{15 - N_{tr}}{15} \times N_{data,j}$$

If $\Delta N_{i,j} = 0$ then the output data of the rate matching is the same as the input data and the rate matching algorithm of subclause 4.2.7.5 does not need to be executed.

If $\Delta N_{i,j} \neq 0$ the parameters listed in subclauses 4.2.7.1.2.1 and 4.2.7.1.2.2 shall be used for determining e_{ini} , e_{plus} , and e_{minus} (regardless if the radio frame is compressed or not).

4.2.7.1.2.1 Uncoded and cConvolutionally encoded TrCHs

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

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

then $q = \left\lceil N_{i,j} / R \right\rceil$

else

 $q = \left[N_{i,i} / (R - N_{i,i}) \right]$

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 $F_i - 1$

$$\mathbf{S}[| \mathbf{x} \times \mathbf{q'} \mathbf{j}| \mod F_i] = (| \mathbf{x} \times \mathbf{q'} \mathbf{j}| \dim F_i)$$

end for

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

For each radio frame, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5, where :

$$X_{i} = N_{i,j}, \text{ and}$$

$$e_{ini} = (a \times S[P1_{Fi}(n_{i})] \times |\Delta N_{i}| + 1) \mod (a \cdot N_{ij}).$$

$$e_{plus} = a \times N_{i,j}$$

$$e_{minus} = a \times |\Delta N_{i}|$$

puncturing for $\Delta N < 0$, repetition otherwise.

4.2.7.2.1.3 Determination of rate matching parameters for uncoded and convolutionally encoded TrCHs

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

For compressed mode by puncturing, ΔN_i is defined as: $\Delta N_i = \Delta N_{i,\max}^{TTI,cm,m}$, instead of the previous relation.

$$N_{max} = \max_{l \in TFS(i)} N_{il}^{TTI}$$

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5. The following parameters are used as input:

$$X_{i} = N_{il}^{TT}$$

$$e_{ini} = 1$$

$$e_{plus} = a \times N_{max}$$

$$e_{\min us} = a \times |\Delta N_{i}|$$

Puncturing if $\Delta N_i < 0$, repetition otherwise. The values of $\Delta N_{i,l}^{TTI}$ may be computed by counting repetitions or puncturing when the algorithm of subclause 4.2.7.5 is run. The resulting values of $\Delta N_{i,l}^{TTI}$ can be represented with following expression.

$$\Delta N_{i,l}^{TTI} = \left[\frac{\left|\Delta N_{i}\right| \times X_{i}}{N_{max}}\right] \times \operatorname{sgn}(\Delta N_{i})$$

For compressed mode by puncturing, the above formula produces $\Delta N_{i,l}^{TTI,m}$ instead of $\Delta N_{i,l}^{TTI}$.

4.2.7.2.2.2 Determination of rate matching parameters for uncoded and convolutionally encoded TrCHs

$$\Delta N_i = \Delta N_{il}^{TTI}$$
$$a=2$$

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in subclause 4.2.7.5. The following parameters are used as input:

$$X_{i} = N_{il}^{TTI}$$

$$e_{ini} = 1$$

$$e_{plus} = a \times N_{il}^{TTI}$$

$$e_{\min us} = a \times |\Delta N_{i}|$$

puncturing for $\Delta N_i < 0$, repetition otherwise.

4.2.7.3 Bit separation and collection in uplink

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits 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.

The first sequence contains:

- All of the systematic bits that are from turbo encoded TrCHs.
- From 0 to 2 first and/or second parity bits that are from turbo encoded TrCHs. These bits come into the first sequence when the total number of bits in a block after radio frame segmentation is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second and third sequences shall be of equal length, whereas the first sequence can contain from 0 to 2 more bits. Puncturing is applied only 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 5 and 6.

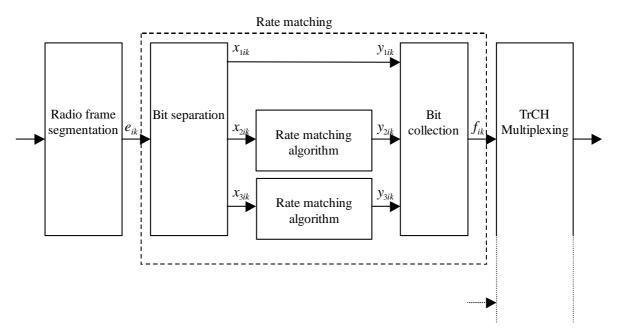


Figure 5: Puncturing of turbo encoded TrCHs in uplink

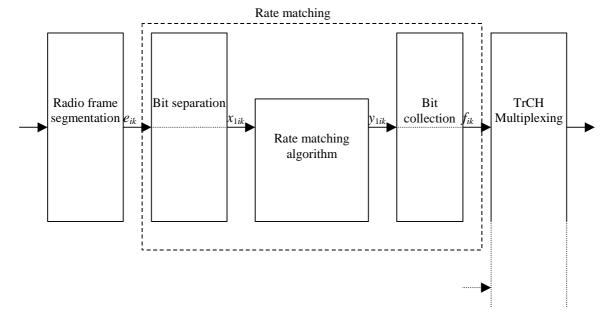


Figure 6: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition in uplink

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. *b* indicates the three sequences defined in this section, with *b*=1 indicating the first sequence, *b* = 2 the second one, and *b* = 3 the third one. The offsets α_b for these sequences are listed in table 5.

Table 5: TTI dependent offset needed for bit separation

TTI (ms)	α1	α2	α3
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 .

TTI (ms)	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
10	0	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

Table 6: Radio frame dependent offset needed for bit separation

4.2.7.3.1 Bit separation

The bits input to the rate matching are denoted by $e_{i1}, e_{i2}, e_{i3}, \dots, e_{iN_i}$, 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 $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.3, with *b*=1 indicating the first sequence, and so forth. 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:

 $x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_n) \mod 3} \qquad k = 1, 2, 3, \dots, X_i \qquad X_i = \lfloor N_i / 3 \rfloor$

$$\begin{aligned} x_{1,i,\lfloor N_i/3 \rfloor+k} &= e_{i,3\lfloor N_i/3 \rfloor+k} & k = 1, ..., N_i \text{ mod } 3 & \text{Note: When } (N_i \text{ mod } 3) = 0 \text{ this row is not needed.} \\ x_{2,i,k} &= e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \\ x_{3,i,k} &= e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \end{aligned}$$

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

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

4.2.7.3.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in subclause 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \dots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i1}, f_{i2}, f_{i3}, \dots, f_{iV_i}$, 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 $(Y_i=X_i)$:

 $\begin{aligned} z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \mod 3} &= y_{1,i,k} & k = 1, 2, 3, ..., Y_i \\ z_{i,3\lfloor N_i/3 \rfloor + k} &= y_{1,i,\lfloor N_i/3 \rfloor + k} & k = 1, ..., N_i \mod 3 & \text{Note: When } (N_i \mod 3) = 0 \text{ this row is not needed.} \\ z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \mod 3} &= y_{2,i,k} & k = 1, 2, 3, ..., Y_i \\ z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \mod 3} &= y_{3,i,k} & k = 1, 2, 3, ..., Y_i \end{aligned}$

After the bit collection, bits $z_{i,k}$ with value δ , where $\delta \not\in \{0, 1\}$, 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:

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

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 $\delta \not\in \{0, 1\}$, 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.

4.2.7.4 Bit separation and collection in downlink

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits 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 of equal lengths.

The first sequence contains :

- All of the systematic bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

Puncturing is applied only 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 7 and 8.

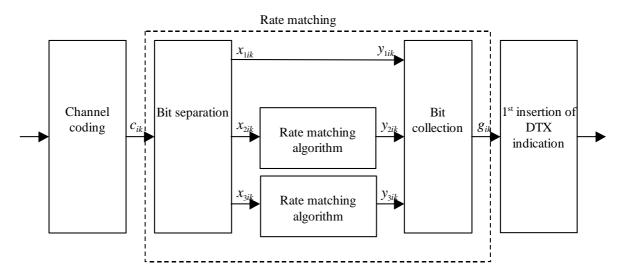


Figure 7: Puncturing of turbo encoded TrCHs in downlink

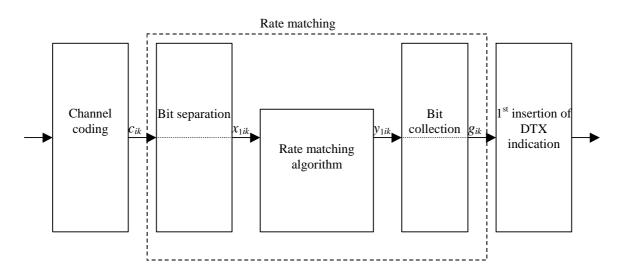


Figure 8: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition in downlink

4.2.7.4.1 Bit separation

The bits input to the rate matching are denoted by $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where *i* is the TrCH number and E_i is the number of bits input to the rate matching block. Note that E_i is a multiple of 3 for turbo encoded TrCHs and that the transport format *l* for simplicity has been left out in the bit numbering, i.e. $E_i = N_{il}^{TTI}$. The bits after separation are

denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \dots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.4, with b=1 indicating the first sequence, and so forth. 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 c_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

$x_{1,i,k} = c_{i,3(k-1)+1}$	$k = 1, 2, 3,, X_i$	$X_i = E_i/3$
$x_{2,i,k} = c_{i,3(k-1)+2}$	$k = 1, 2, 3,, X_i$	$X_i = E_i/3$
$x_{3,i,k} = c_{i,3(k-1)+3}$	$k = 1, 2, 3,, X_i$	$X_i = E_i / 3$

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

$$x_{1,i,k} = c_{i,k}$$
 $k = 1, 2, 3, ..., X_i$ $X_i = E$

4.2.7.4.2 Bit collection

The bits x_{bik} are input to the rate matching algorithm described in subclause 4.2.7.5. The bits output from the rate matching algorithm are denoted $y_{bi1}, y_{bi2}, y_{bi3}, \dots, y_{biY_i}$.

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{bi1}, z_{bi2}, z_{bi3}, \dots, z_{biY_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $g_{i1}, g_{i2}, g_{i3}, \dots, g_{iG_i}$, where *i* is the TrCH number and $G_i = N_{il}^{TTI} + \Delta N_{il}^{TTI}$. The relations between y_{bik}, z_{bik} , and g_{ik} are given below.

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

$z_{i,3(k-1)+1} = y_{1,i,k}$	$k = 1, 2, 3,, Y_i$
$z_{i,3(k-1)+2} = y_{2,i,k}$	$k = 1, 2, 3,, Y_i$
$z_{i,3(k-1)+3} = y_{3,i,k}$	$k = 1, 2, 3,, Y_i$

After the bit collection, bits $z_{i,k}$ with value δ , where $\delta \not\in \{0, 1\}$, are removed from the bit sequence. Bit $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index *k* after puncturing, bit $g_{i,2}$ 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, $g_{i,k}=z_{i,k}$ and $Y_i=G_i$.

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 $g_{i,1}$ corresponds to the bit $z_{i,k}$ with smallest index *k* after puncturing, bit $g_{i,2}$ corresponds to the bit $z_{i,k}$ with second smallest index *k* after puncturing, and so on.

_

	CHANGE REQUEST
x	25.215 CR 110 # rev - ^{# Current version:} 3.9.0 [#]
For <u>HELP</u> on usi	ing this form, see bottom of this page or look at the pop-up text over the $lpha$ symbols.
Proposed change af	fects: # (U)SIM ME/UE X Radio Access Network X Core Network
Title: ೫	Removal of channel coding option "no coding" for FDD
Source: ೫	TSG RAN WG1
Work item code: ೫	TEI Date: # 18.02.2002
E	Jse one of the following categories: Use one of the following releases: F (correction) 2 (GSM Phase 2) A (corresponds to a correction in an earlier release) R96 (Release 1996) B (addition of feature), R97 (Release 1997) C (functional modification of feature) R98 (Release 1998) D (editorial modification) R99 (Release 1999) Defailed explanations of the above categories can REL-4 (Release 4) De found in 3GPP TR 21.900. REL-5 (Release 5)
Summary of change	 The channel coding option "no coding" has been removed in the IE "Type of channel coding". Isolated Impact Analysis This change affects the channel coding type. It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise.
Consequences if not approved:	An option that is not used by any Radio Bearer would be a mandatory feature for all UEs.
Clauses affected:	¥ 5.2.6
Other specs affected:	XOther core specificationsX25.201, 25.212, 25.222, 25.225Test specifications0&M Specifications
Other comments:	ж.

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked **#** contain pop-up help information about the field that they are closest to.

5.2.6 Transport channel BER

Definition The transport channel BER is an estimation of the average bit error rate (BER) of the DPDCH data of a Radio Link Set. The transport channel (TrCH) BER is measured from the data considering only non-punctured bits at the input of the channel decoder in Node B. It shall be possible to report an estimate of the transport channel BER for a TrCH after the end of each TTI of the TrCH. The reported TrCH BER shall be an estimate of the BER during the latest TTI for that TrCH. Transport channel BER is only required to be reported for TrCHs that are channel coded.

[#] 25	5.215 CR 111 # rev - ^{# Current version:} 4.3.0 [#]		
For <u>HELP</u> on using	this form, see bottom of this page or look at the pop-up text over the # symbols.		
Proposed change affect	cts: # (U)SIM ME/UE X Radio Access Network X Core Network		
Title: # Re	emoval of channel coding option "no coding" for FDD		
Source: ೫ TS	SG RAN WG1		
Work item code: # TE	El Date: # 18.02.2002		
Det	channel coding option "no coding" for FDD and 3.84 Mcps TDD. For 1.28 Mcps TDD this option is still valid.		
	It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise.		
Consequences if # not approved:	An option that is not used by any Radio Bearer would be a mandatory feature for all UEs.		
Clauses affected: #	5.2.6		
Other specs # affected:	X Other core specifications # 25.201, 25.212, 25.222, 25.225 Test specifications 0&M Specifications		
Other comments: #	8		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

1) Fill out the above form. The symbols above marked **#** contain pop-up help information about the field that they are closest to.

5.2.6 Transport channel BER

Definition The transport channel BER is an estimation of the average bit error rate (BER) of the DPDCH data of a Radio Link Set. The transport channel (TrCH) BER is measured from the data considering only non-punctured bits at the input of the channel decoder in Node B. It shall be possible to report an estimate of the transport channel BER for a TrCH after the end of each TTI of the TrCH. The reported TrCH BER shall be an estimate of the BER during the latest TTI for that TrCH. Transport channel BER is only required to be reported for TrCHs that are channel coded.

		CHA		EQU	EST	•		CR-Form-v5
^ж 2	<mark>5.222</mark>	CR 067	ж r	ev 🥤	^ж	Current vers	ion: 3.7 .	. <mark>0</mark> [#]
For <u>HELP</u> on using	g this for	m, see botton	n of this pag	ie or loc	ok at th	e pop-up text	over the X	symbols.
Proposed change affe	ects: ೫	(U)SIM	ME/UE	X Ra	adio Ad	ccess Network	X Core	e Network
Title: % R	emoval	of channel co	ding option	<mark>"no cod</mark>	<mark>ing" fo</mark>	<mark>r 3.84 Mcps T</mark>	DD	
Source: ೫ T	SG RAN	WG1						
Work item code: 🕷 🕇	El					Date: ೫	18.02.200)2
De	e <u>one</u> of t <i>F</i> (corr <i>A</i> (corr <i>B</i> (ada <i>C</i> (func <i>D</i> (edit tailed exp found in 3	responds to a c lition of feature, ctional modificat orial modification lanations of the 3GPP <u>TR 21.90</u>	orrection in a , tion of featur on) e above cate <u>)0</u> .	e) gories ca	an	2 e) R96 R97 R98 R99 REL-4 REL-5	R99 the following (GSM Phase (Release 19 (Release 19 (Release 19 (Release 19 (Release 4) (Release 5)	9 2) 96) 97) 98) 99)
Reason for change: S	chan		tion "no coc			it has been a and 3.84 Mcps		
Summary of change:	chan	nel coding".		o coding	" has l	been removed	in the IE "T	Гуре of
		ed Impact Ar	-		1 type			
	It wo	uld not affect	implementa	tions be	ehaving	g like indicated		
Consequences if a solution of approved:	# An o all U		ot used by a	any Rad	io Bea	rer would be a	amandator	y feature for
Clauses affected:	<mark>₩ 4.2.2</mark>	<mark>, 4.2.3, 4.2.7.</mark>	1, 4.2.7.2					
Other specs	Te	her core spec est specificatio &M Specificat	ons	ж 2	25.201	, 25.212, 25.2	15, 25.225	
Other comments:	ж							

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

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 or $\frac{1}{7}$ 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 $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$ 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 $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$, where *i* is the TrCH number and $X_i = M_i B_i$. They are defined by the following relations:

$$\begin{aligned} x_{ik} &= b_{i1k} \qquad k = 1, 2, \dots, B_i \\ x_{ik} &= b_{i,2,(k-B_i)} \qquad k = B_i + 1, B_i + 2, \dots, 2B_i \\ x_{ik} &= b_{i,3,(k-2B_i)} \qquad k = 2B_i + 1, 2B_i + 2, \dots, 3B_i \\ \dots \\ x_{ik} &= b_{i,M_i,(k-(M_i-1)B_i)} \qquad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \dots, M_i B_i \end{aligned}$$

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, for $C_i \neq 0$, are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and K_i is the number of bits per code block.

Number of code blocks:

$$_{i} = \left[X_{i} / Z \right]$$

$$C_{i} = \begin{cases} \left[X_{i} / Z \right] & \text{when } Z \neq \textit{unlimited} \\ 0 & \text{when } Z = \textit{unlimited} \text{ and } X_{i} = 0 \\ 1 & \text{when } Z = \textit{unlimited} \text{ and } X_{i} \neq 0 \end{cases}$$

Number of bits in each code block (applicable for $C_i \neq 0$ only):

if $X_i < 40$ and Turbo coding is used, then

$$K_i = 40$$

else

$$K_i = \overline{X_i} / C_i \overline{A}$$

end if

Number of filler bits: $Y_i = C_i K_i - X_i$

for k = 1 to Y_i

-- Insertion of filler bits

$$o_{i1k} = 0$$

end for

for $k = Y_i + 1$ to K_i

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

end for

r = 2 -- Segmentation while $r \le C_i$

for k = 1 to K_i

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

end for

r = r+1

end while

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, 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 $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$, 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: $Y_i = 3*K_i + 12$;

I

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

Table 1: Usage of channel coding scheme and coding rate

4.2.7.1.1 Uncoded and cConvolutionally encoded TrCHs

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

 $X_i = N_{i,j} \\$

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

if $\mathbf{R} \neq 0$ and $2 \times \mathbf{R} \leq N_{i,j}$

then
$$\mathbf{q} = \left\lceil N_{i,j} / R \right\rceil$$

else

$$\mathbf{q} = \left\lceil N_{i,j} / (R - N_{i,j}) \right\rceil$$

endif

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

else

q'= q

endif

for $\mathbf{x} = 0$ to F_i -1

```
S[|\lfloor x \times q' \rfloor| \mod F_i] = (|\lfloor x^*q' \rfloor| \dim F_i)
```

end for

 $e_{ini} = (\mathbf{a} \times \mathbf{S}[\mathbf{P1}_{Fi}(n_i)] \times |\Delta N_i| + 1) \mod (\mathbf{a} \times N_{i,j})$

 $e_{plus} = \mathbf{a} \times X_i$

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

puncturing for $\Delta N_i < 0$, repetition otherwise.

4.2.7.2 Bit separation and collection for rate matching

The systematic bits of turbo encoded TrCHs shall not be punctured, the other bits 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.

The first sequence contains:

- All of the systematic bits that are from turbo encoded TrCHs.
- From 0 to 2 first and/or second parity bits that are from turbo encoded TrCHs. These bits come into the first sequence when the total number of bits in a block after radio frame segmentation is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second sequence contains:

- All of the first parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The third sequence contains:

- All of the second parity bits that are from turbo encoded TrCHs, except those that go into the first sequence when the total number of bits is not a multiple of three.
- Some of the systematic, first parity and second parity bits that are for trellis termination.

The second and third sequences shall be of equal length, whereas the first sequence can contain from 0 to 2 more bits. Puncturing is applied only 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.

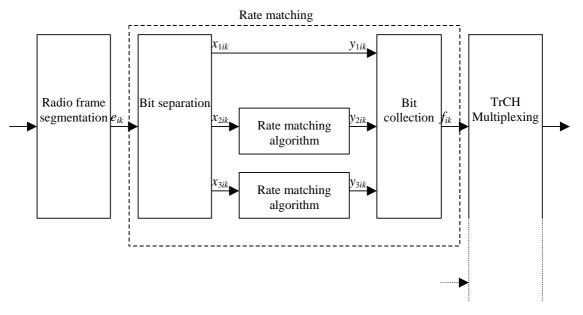


Figure 4: Puncturing of turbo encoded TrCHs

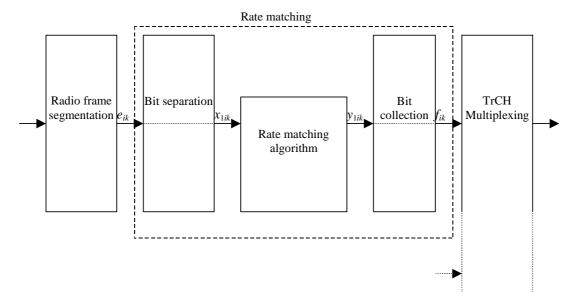


Figure 5: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition

The bit separation is dependent on the 1st interleaving and offsets are used to define the separation for different TTIs. *b* indicates the three sequences defined in this section, with b=1 indicating the first sequence, b=2 the second one, and b=3 the third one.

The offsets α_b for these sequences are listed in table 5.

Table 5: TTI dependent offset needed for bit separation

TTI (ms)	α1	<i>α</i> ₂	<i>α</i> 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} .

TTI (ms)	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
10	0	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

Table 6: Radio frame dependent offset needed for bit separation

4.2.7.2.1 Bit separation

The bits input to the rate matching are denoted by $e_{i,1}, e_{i,2}, e_{i,3}, \dots, e_{i,N_i}$, 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 $x_{b,i,1}, x_{b,i,2}, x_{b,i,3}, \dots, x_{b,i,X_i}$. For turbo encoded TrCHs with puncturing, *b* indicates the three sequences defined in section 4.2.7.2, with *b*=1 indicating the first sequence, and so forth. 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:

 $x_{1,i,k} = e_{i,3(k-1)+1+(\alpha_1+\beta_n) \mod 3} \qquad k = 1, 2, 3, ..., X_i \qquad X_i = \lfloor N_i / 3 \rfloor$

$$\begin{aligned} x_{1,i,\lfloor N_i/3 \rfloor+k} &= e_{i,3\lfloor N_i/3 \rfloor+k} & k = 1, ..., N_i \text{ mod } 3 & \text{Note: When } (N_i \text{ mod } 3) = 0 \text{ this row is not needed.} \\ x_{2,i,k} &= e_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \\ x_{3,i,k} &= e_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \text{ mod } 3} & k = 1, 2, 3, ..., X_i & X_i = \lfloor N_i/3 \rfloor \end{aligned}$$

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

Bit collection is the inverse function of the separation. The bits after collection are denoted by $z_{b,i,1}, z_{b,i,2}, z_{b,i,3}, \dots, z_{b,i,Y_i}$. After bit collection, the bits indicated as punctured are removed and the bits are then denoted by $f_{i,1}, f_{i,2}, f_{i,3}, \dots, f_{i,V_i}$, where *i* is the TrCH number and $V_i = N_{i,j} + \Delta N_{i,j}$. The relations between $y_{b,i,k}, z_{b,i,k}$, and $f_{i,k}$ are given below.

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

$$\begin{aligned} z_{i,3(k-1)+1+(\alpha_1+\beta_{n_i}) \mod 3} &= y_{1,i,k} & k = 1, 2, 3, ..., Y_I \\ z_{i,3\lfloor N_i/3 \rfloor + k} &= y_{1,i,\lfloor N_i/3 \rfloor + k} & k = 1, ..., N_i \mod 3 & \text{Note: When } (N_i \mod 3) = 0 \text{ this row is not needed.} \\ z_{i,3(k-1)+1+(\alpha_2+\beta_{n_i}) \mod 3} &= y_{2,i,k} & k = 1, 2, 3, ..., Y_i \\ z_{i,3(k-1)+1+(\alpha_3+\beta_{n_i}) \mod 3} &= y_{3,i,k} & k = 1, 2, 3, ..., Y_i \end{aligned}$$

After the bit collection, bits $z_{i,k}$ with value δ , where $\delta \not \leq \{0, 1\}$, 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.

		CHAI		EQU	EST	-			CR-Form-v5
ж 2	2 <mark>5.222</mark>	CR 068	жr	ev 1	ж	Current vers	ion: 4.	2.0	ж
For <u>HELP</u> on usir	ng this for	rm, see bottorr	of this pag	e or loo	k at th	e pop-up text	over the	Ж syn	nbols.
Proposed change aff	fects: Ж	(U)SIM	ME/UE	X Ra	adio Ad	ccess Network	K <mark>X</mark> Co	ore Ne	twork
Title: ೫	Removal	of channel coo	ding option	<mark>'no cod</mark> i	<mark>ng" fo</mark>	<mark>r 3.84 Mcps T</mark>	DD		
Source: #	TSG RAN	WG1							
Work item code: 🕱 🧧	TEI					<i>Date:</i> ೫	18.02.2	2002	
D	lse <u>one</u> of F (cor A (cor B (ado C (fun D (edi etailed ex	the following ca rection) responds to a c dition of feature) ctional modificat torial modificatio blanations of the 3GPP <u>TR 21.90</u>	orrection in a , tion of featur on) above cates	e)		Release: # Use <u>one</u> of 2 e) R96 R97 R98 R99 REL-4 REL-5		ase 2) 1996) 1997) 1998) 1999) 4)	ases:
Reason for change:	char	e last joint Adl nel coding opt this option is a	ion "no cod						
Summary of change:		channel coding nel coding".	g option "no	coding	" has l	been removed	I in the IE	Е "Туре	e of
		ed Impact An	-						
	lt wo	hange affects uld not affect i t implementat	mplementa	tions be	having	g like indicated			vould
Consequences if not approved:	₩ An o all U	ption that is no Es.	ot used by a	iny Rad	io Bea	irer would be a	a mandat	ory fea	ature for
Clauses affected:	೫ <mark>4.2.2,</mark>	4.2.3							
Other specs affected:	Т	ther core spec est specificatio &M Specificati	ns	ж 2	25.201	, 25.212, 25.2	15, 25.22	25	
Other comments:	ж								

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

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 or, turbo coding or no coding is used for the TrCH. For 1.28 Mcps TDD also "no coding" is supported.

4.2.2.1 Concatenation of transport blocks

The bits input to the transport block concatenation are denoted by $b_{im1}, b_{im2}, b_{im3}, \dots, b_{imB_i}$ 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 $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iX_i}$, where *i* is the TrCH number and $X_i = M_i B_i$. They are defined by the following relations:

$$\begin{aligned} x_{ik} &= b_{i1k} \quad k = 1, 2, \dots, B_i \\ x_{ik} &= b_{i,2,(k-B_i)} \quad k = B_i + 1, B_i + 2, \dots, 2B_i \\ x_{ik} &= b_{i,3,(k-2B_i)} \quad k = 2B_i + 1, 2B_i + 2, \dots, 3B_i \\ \dots \\ x_{ik} &= b_{i,M_i,(k-(M_i-1)B_i)} \quad k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \dots, M_i B_i \end{aligned}$$

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 (only for 1.28 Mcps TDD) : *Z* = *unlimited*.

The bits output from code block segmentation, for $C_i \neq 0$, are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and K_i is the number of bits per code block.

Number of code blocks:

$$C_{i} = \begin{cases} \begin{bmatrix} X_{i} / Z \end{bmatrix} & \text{when } Z \neq \textit{unlimited} \\ 0 & \text{when } Z = \textit{unlimited} \text{ and } X_{i} = 0 \\ 1 & \text{when } Z = \textit{unlimited} \text{ and } X_{i} \neq 0 \end{cases}$$

Number of bits in each code block (applicable for $C_i \neq 0$ only):

if $X_i < 40$ and Turbo coding is used, then

 $K_i = 40$

else

$$K_i = /\overline{X_i} / C_i$$

end if

Number of filler bits: $Y_i = C_i K_i - X_i$

for k = 1 to Y_i

-- Insertion of filler bits

$$o_{i1k} = 0$$

end for

for $k = Y_i + 1$ to K_i

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

end for

r = 2 -- Segmentation

while $r \leq C_i$

for k = 1 to K_i

 $O_{irk} = X_{i,(k+(r-1)K_i-Y_i)}$

end for

r = r+1

end while

4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \dots, o_{irK_i}$, 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 $y_{ir1}, y_{ir2}, y_{ir3}, \dots, y_{irY_i}$, 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 (only for 1.28 Mcps TDD).

Usage of coding scheme and coding rate for the different types of TrCH is shown in tables 1 and 2. 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: $Y_i = K_i$ (only for 1.28 Mcps TDD).

Table 1: Usage of channel coding scheme and coding rate for 3.84Mcps TD

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

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

Table 2: Usage of channel codin	a scheme and codin	a rate for 1 28Mcns TDD
Table 2. Usaye of charmer could	y scheme and count	grate for 1.20micps 100

	CR-Form-v5							
ж	25.225 CR 044 * rev - [*] Current version: 3.9.0 [*]							
For <u>HELP</u> on us	For HELP on using this form, see bottom of this page or look at the pop-up text over the # symbols.							
Proposed change a	affects: # (U)SIM ME/UE X Radio Access Network X Core Network							
Title: %	Removal of channel coding option "no coding" 3.84 Mcps TDD							
Source: #	TSG RAN WG1							
Work item code: ೫	TEI Date: # 18.02.2002							
Category: # Reason for change Summary of chang	channel coding option "no coding" for FDD and 3.84 Mcps TDD. For 1.28 Mcps TDD this option is still valid.							
	This change affects the channel coding type. It would not affect implementations behaving like indicated in the CR, it would affect implementations supporting the corrected functionality otherwise.							
Consequences if not approved:	An option that is not used by any Radio Bearer would be a mandatory feature for all UEs.							
Clauses affected:	¥ 5.2.5							
Other specs affected:	XOther core specificationsX25.201, 25.212, 25.215, 25.222Test specifications0&M Specifications							
Other comments:	¥							

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

5.2.5 Transport channel BER

Definition	The transport channel BER is an estimation of the average bit error rate (BER) of DCH or USCH data. The transport channel (TrCH) BER is measured from the data considering only non-punctured bits at the input of the channel decoder in Node B. It shall be possible to report an estimate of the transport channel BER for a TrCH after the end of each TTI of the TrCH. The reported TrCH BER shall be an estimate of the BER during the latest TTI for that TrCH. Transport channel BER is only required to be reported for TrCHs that are channel coded.
------------	---

		CHAN	GE REC	UEST			CR-Form-v5
^ж 2	2 <mark>5.225</mark>	CR <mark>045</mark>	ж rev	- [#]	Current vers	^{ion:} 4.3.0	ж
For <u>HELP</u> on usin	ng this fori	m, see bottom	of this page o	r look at the	e pop-up text	over the # syn	nbols.
Proposed change aff	fects: Ж	(U)SIM	ME/UE X	Radio Ac	cess Network	X Core Ne	twork
Title: %	Removal o	o <mark>f channel codi</mark>	ng option "no	coding" for	3.84 Mcps T	DD	
Source: ೫ <mark>-</mark>	<mark>TSG RAN</mark>	WG1					
Work item code: # 📑	TEI				Date: ೫	18.02.2002	
D	se <u>one</u> of t F (corr A (corr B (add C (fund D (edite etailed exp	he following cate ection) responds to a col ition of feature), stional modification orial modification lanations of the BGPP <u>TR 21.900</u>	rection in an e on of feature)) above categori		2 R96 R97 R98 R99 REL-4	REL-4 the following rele (GSM Phase 2) (Release 1996) (Release 1997) (Release 1998) (Release 1999) (Release 4) (Release 5)	eases:
Reason for change:	chan		on "no coding			agreed to removes TDD. For 1.28	
Summary of change:	chan	nel coding".		ding" has b	een removec	t in the IE "Type	e of
		ed Impact Ana					
		hange affects t					
		uld not affect in t implementation				d in the CR, it v lity otherwise.	vould
Consequences if not approved:	Han op all UE		used by any	Radio Bear	rer would be a	a mandatory fe	ature for
Clauses affected:	೫ <mark>5.2.5</mark>						
Other specs affected:	Те	her core specif st specification &M Specificatio	S	\$ 25.201,	25.212, 25.2	15, 25.222	
Other comments:	ж						

Comprehensive information and tips about how to create CRs can be found at: <u>http://www.3gpp.org/3G_Specs/CRs.htm</u>. Below is a brief summary:

5.2.5 Transport channel BER

Definition	The transport channel BER is an estimation of the average bit error rate (BER) of DCH or USCH data. The transport channel (TrCH) BER is measured from the data considering only non-punctured bits at the input of the channel decoder in Node B. It shall be possible to report an estimate of the transport channel BER for a TrCH after the end of each TTI of the TrCH. The reported TrCH BER shall be an estimate of the BER during the latest TTI for that TrCH.
	For 1.28 Mcps TDD: Transport channel BER is only required to be reported for TrCHs that are channel coded.