TSG-RAN Working Group 1(Radio) meeting #9 ***TSGR1#9(99) k07***

30 November – 3 December 1999, Dresden, Germany

Source : Samsung and LGIC

Title : Revised CR to 25.222 for clarification of bit separation and collection

Document for : Approval

1 Introduction

Bit separation and bit collection are currently not sufficiently described in 25.222. We proposed the modifications in TSGR1#8(99)f28, which were agreed in the WG1 meeting in NY. But, it is proposed that instead of making the limited changes proposed in f28, the section should be rewritten with notations more similar to the rest of 25.222. The changes proposed in this document are listed below.

Entire section 4.2.7	The following notation is currently used for the number of bits before rate matching: $N = N_i = N_{ij}$ It is proposed that *N* is replaced by X_i so that it is possible to distinguish between different TrCHs. Similarly, it is proposed that *DN* is replaced by DN_i.
Entire section 4.2.7	Index *b* is introduced to indicate systematic or parity bits.
Section 4.2.7.1.1 / 4.2.7.1.2	Divided into subsections to ease understanding. (Note that current text does not define rate matching for uncoded TrCHs.)
Section 4.2.7.2	Completely rewritten. A new notation is introduced for the bits. The section is divided into subsections. Clearly states what happens when N_{ij} is not a multiple of three (the last 1 or 2 bits can not be punctured).

2 Text Proposal

3GPP TSG RAN WG1 Meeting #9
Dresden, Germany, Nov 30 – Dec 3, 1999

Document ***R1-99K07***
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**	**002r1**	Current Version:	3.0.0

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

For submission to:	TSG-RAN #6	For approval	X	strategic		*(for SMG*
list expected approval meeting # here ↑		for information		non-strategic		*use only)*

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

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

Source: Samsung and LGIC   **Date:** 1999-11-22

Subject: Clarification of bit separation and collection

Work item:

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

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

Release:

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

Reason for change: Current description of bit separation and collection can easily be misinterpreted.

Clauses affected: 4.2.7

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: It would be desired that the specification of rate matching section be consistent in notations and the way of description throughout the entire TS 25.222 and be aligned with that of TS 25.212.

help.doc

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

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 ~~second~~TrCH multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

Notation used in section 4.2.7 and subsections:

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

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

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

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

PL: Puncturing limit for uplink. This value limits the amount of puncturing that can be applied in order to minimise the number of dedicated 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.

I: Number of TrCHs in a CCTrCH.

Z_{mj}: Intermediate calculation variable.

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

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

Q: Average puncturing distance.

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

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

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

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

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

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

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

~~X~~<u>b</u>: <u>Indicates systematic and parity bits</u>
<u>b=1:</u> Systematic bit<u>. X(t)</u> in section 4.2.3.2.1.
Y: <u>b=2:</u> 1st parity bit (from the upper Turbo constituent encoder)<u>. Y(t)</u> in section 4.2.3.2.1.
Y': <u>b=3:</u> 2nd parity bit (from the lower Turbo constituent encoder)<u>. Y'(t)</u> in section 4.2.3.2.1.

~~NOTE: Time index t in 4.2.3.2.1 is omitted for simplify the rate matching description~~

4.2.7.1 Determination of rate matching parameters

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

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

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

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

Puncturing can be used to minimise the required transmission capacity. The maximum amount of puncturing that can be applied is signalled from higher layers and denoted by PL. The possible values for N_{data} in depend on the number of dedicated physical channels and on their characteristics (spreading factor, length of midamble and TFCI, usage of TPC and multiframe structure), respectively. The supported set of N_{data} , denoted SET0, depends on the UE capabilities.

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

SET1 = { N_{data} in SET0 such that $N_{data} - PL \cdot \sum_{x=1}^{I} \frac{RM_x}{\min_{1 \le y \le I} \{RM_y\}} \cdot N_{x,j}$ is non negative }

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

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

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

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

~~For convolutional codes,~~

4.2.7.1.1 Uncoded and convolutionally encoded TrCHs

a = 2

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

X_iN = $N_{i,j}$

q= $\lfloor X_i$N /(| ΔN_i |)$\rfloor$

If q is even

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

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

else

q' = q

endif

for x = 0 to F_i-1

S(I_F ($\lceil$x*q'$\rceil$ mod F_i)) = ($\lceil$x*q'$\rceil$ div F_i)–

End for

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

$e_{plus} = a \cdot X_i N$

$e_{minus} = a \times |DN_i|$

puncturing for $DN_i<0$, repetitionating otherwise.

4.2.7.1.2 Turbo encoded TrCHs

For turbo codes, if repetition is to be performed, such as $DN_{i,j}>0$, parameters for turbo codes are the same as parameter for convolutional codes. If repetition is to be performed on turbo encoded TrCHs, i.e. $DN_{i,j}>0$, the parameters in section 4.2.7.1.1 are used.

If puncturing is to be performed, the parameters are as followsbelow 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*=2for Y sequence, and

a = 1 when *b*=3for Y' sequence.

$$\Delta N_i = \begin{cases} \lfloor \Delta N_{i,j}/2 \rfloor, & b=2 \\ \lceil \Delta N_{i,j}/2 \rceil, & b=3 \end{cases} \quad \Delta N = \begin{cases} \lfloor DN_{i,j}/2 \rfloor & \text{for Y sequence} \\ \lceil DN_{i,j}/2 \rceil & \text{for Y' sequence} \end{cases}$$

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

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

if($q \le 2$)

for x=0 to F_i-1

S[I_F[(3x+b-1) mod F_i]] = x mod 2;if(Y sequence)

S[I_F[(3x+1) mod F_i]] = x mod 2;

if(Y' sequence)

S[I_F [(3x+2) mod F_i]] = x mod 2;

end for

else

if *q* is even

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

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

else $q' = q$

endif

for x=0 to F_i –1

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

S[I_F[(3r+b-1) mod F_i]] = ⌈ x*q' ⌉ div F_i;if(Y sequence)

S[I_F[(3r+1) mod F_i]] = ⌈ x*q' ⌉ div F_i;

~~if(Y' sequence)~~

~~S[I_F[(3r+2) mod F_i]] = ⌈ x*q' ⌉ div F_i;~~

endfor

endif

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

~~N~~X_i is as above,

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

$e_{plus} = a \cdot X_i$

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

~~puncturing for DN<0, repeating otherwise.~~

4.2.7.2 Bit separation and collection for rate matching

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

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

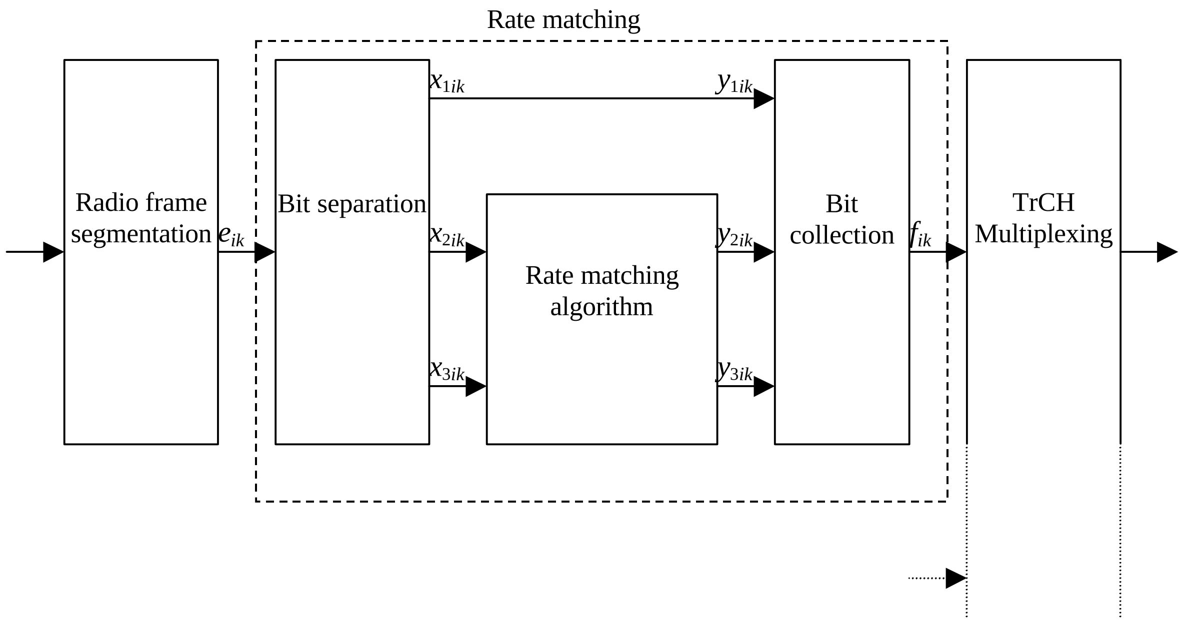


Figure 4-5: Puncturing of turbo encoded TrCHs

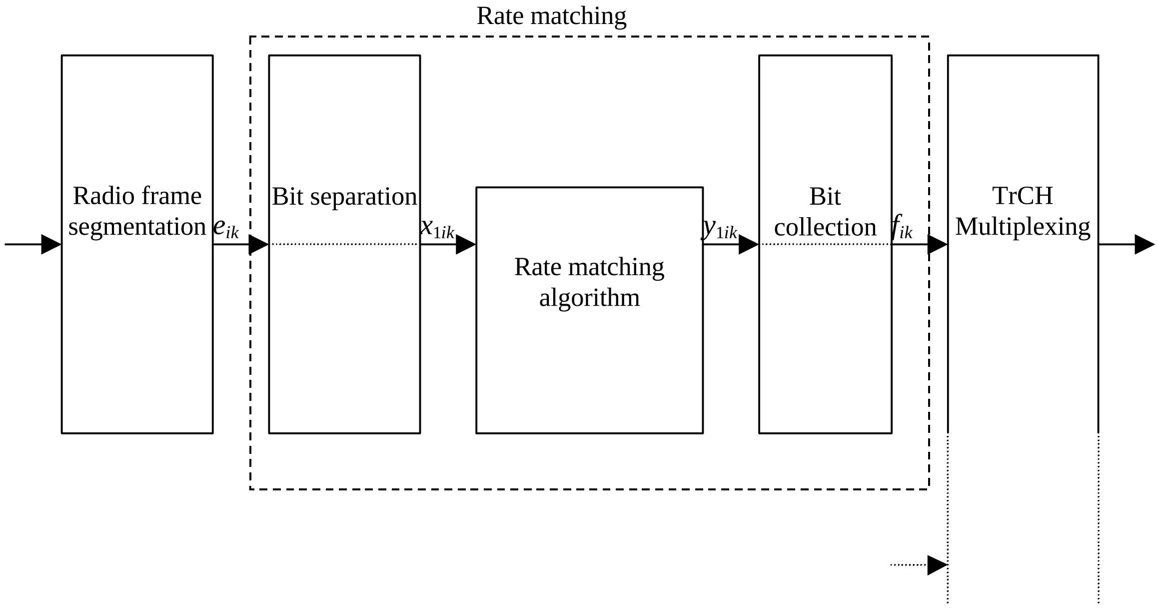


Figure 4-6: Rate matching for uncoded TrCHs, convolutionally encoded TrCHs, and for turbo encoded TrCHs with repetition.

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

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

TTI (ms)	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 4.2.7-2: 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 $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$, where i is the TrCH number and N_i is the number of bits input to the rate matching block. Note that the transport format combination number j for simplicity has been left out in the bit numbering, i.e. $N_i=N_{ij}$. The bits after separation are denoted by $x_{bi1}, x_{bi2}, x_{bi3}, \ldots, x_{biX_i}$. For turbo encoded TrCHs with puncturing, b indicates systematic, first parity, or second parity bit. For all other cases b is defined to be 1. X_i is the number of bits in each separated bit sequence. The relation between e_{ik} and x_{bik} is given below.

For turbo encoded TrCHs with puncturing:

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

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

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

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

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

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

4.2.7.2.2 Bit collection

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

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

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

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

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

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

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

After the bit collection, bits $z_{i,k}$ with value d, where $d \in \{0, 1\}$, 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.

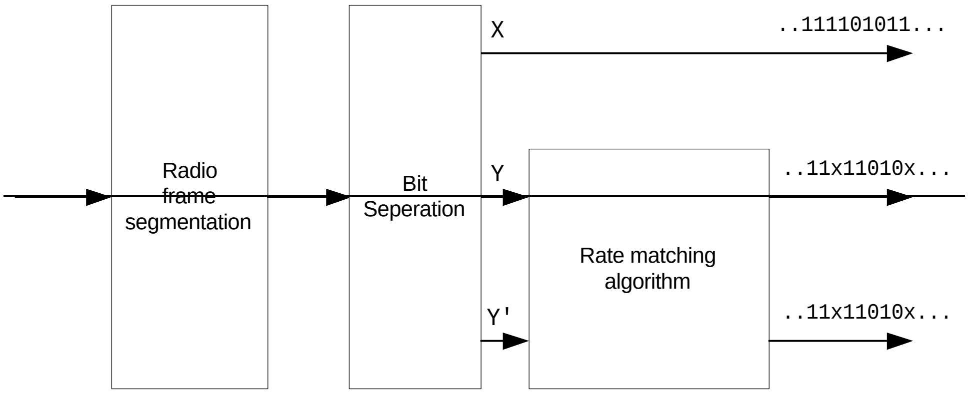
For uncoded TrCHs, convolutionally encoded TrCHs, and turbo encoded TrCHs with repetition:

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

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 $d \in \{0, 1\}$, 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.

~~4.2.7.2 Bit separation for rate matching~~



~~Figure 4-5: Overall rate matching block diagram after first interleaving where x denotes punctured bit~~

~~Rate matching puncturing for Turbo codes is applied separately to *Y* and *Y'* sequences. No puncturing is applied to *X* sequence. Therefore, it is necessary to separate *X*, *Y*, and *Y'* sequences before rate matching is applied.~~

~~There are two different alternation patterns in bit stream from Radio frame segmentation according to the TTI of a TrCH as shown in table 4.2.7-1.~~

~~Table 4.2.7-1: Alternation patterns of bits from radio frame segmentation~~

~~TTI (msec)~~	~~Alternation patterns~~
~~10, 40~~	~~… X,Y,Y',…~~
~~20, 80~~	~~… X,Y',Y,…~~

~~In addition, each radio frame of a TrCH starts with different initial parity type. Table 4.2.7-2 shows the initial parity type of each radio frame of a TrCH with TTI = {10, 20, 40, 80} msec.~~

~~Table 4.2.7-2: Initial parity type of radio frames of TrCH~~

~~TTI (msec)~~	~~Radio frame indexes (n_i)~~							
	~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~	~~7~~
~~10~~	~~X~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~
~~20~~	~~X~~	~~Y~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~
~~40~~	~~X~~	~~Y'~~	~~Y~~	~~X~~	~~NA~~	~~NA~~	~~NA~~	~~NA~~
~~80~~	~~X~~	~~Y~~	~~Y'~~	~~X~~	~~Y~~	~~Y'~~	~~X~~	~~Y~~

~~Tables 4.2.7-1 and 4.2.7-2 defines a complete output bit pattern from Radio frame segmentation.~~

~~Ex. 1. TTI = 40 msec, n_i = 2~~
~~Radio frame pattern: *Y, Y', X, Y, Y', X, Y, Y', X, …*~~

~~Ex. 2 TTI = 40 msec, n_i = 3~~
~~Radio frame pattern: *X, Y, Y', X, Y, Y', X, Y, Y', X, …*~~

~~Therefore, bit separation is achieved with the alternative selection of bits with the initial parity type and alternation pattern specified in tables 4.2.7-1 and 4.2.7-2 according to the TTI and n_i of a TrCH.~~

4.2.7.3 Rate matching pattern determination

The bits input to the rate matching are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ ~~$e_{i1}, e_{i2}, e_{13}, \ldots, e_{iN_i}$~~, where *i* is the TrCH ~~with $X_i = N_{ij} = N_i$. Here N~~ and X_i is the parameter given in sections 4.2.7.1.1 and 4.2.7.1.2. The bits output from the rate matching are denoted by $f_{i1}, f_{i2}, f_{13}, \ldots, f_{iV_i}$, where *i* is the TrCH number and $V_i=N+DN$.

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

The rate matching rule is as follows:

```
if puncturing is to be performed
    e = e_ini       -- initial error between current and desired puncturing ratio
    m = 1           -- index of current bit
    do while m <= X_i N
        e = e – 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}puncture bit e_i,m
            e = e + e_plus      -- update error
        end if
        m = m + 1               -- next bit
    end do
else
    e = e_ini       -- initial error between current and desired puncturing ratio
    m = 1           -- index of current bit
    do while m <= X_i N
        e = e – e_minus         -- update error
        do while e <= 0         -- check if bit number m should be repeated
            repeat bit x_i,m e_i,m
            e = e + e_plus  -- update error
        end do
        m = m + 1               -- next bit
    end do
end if
```

A repeated bit is placed directly after the original one.