TSG-RAN Working Group1 meeting #7 *TSGR1#7(99)d84*
Hannover, Germany, August 30 – September 3, 1999

Agenda Item:
Source: SAMSUNG Electronics Co. and LGIC
Title: Text proposal for Turbo codes and rate matching in TS 25.212, TS 25.222 (rev. of R1-99d56)
Document for: Decision

1. Introduction

Rate matching algorithm for Turbo codes proposed by Samsung and LGIC has been approved as a working assumption in Ad Hoc 5 meeting. This document proposes a revised text proposal of R1-99d56 [1]. The revisions are mainly focused on notational consistency and more clear description for the TS 25.212 and 25.222 specification [2,3], as recommended by the chairman in WG1 plenary meeting.

2. Text proposal for TS 25.212

-------------------- Start text proposal --------------------

4.2.3. Channel coding

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

- Convolutional coding
- Turbo coding
- No channel coding

Table 1: Error Correction Coding Parameters

~~Transport channel type~~	~~Coding scheme~~	~~Coding rate~~
~~BCH~~	~~Convolutional code~~	~~1/2~~
~~PCH~~		
~~FACH~~		
~~RACH~~		
~~DCH~~		~~1/3, 1/2, or no coding~~
~~DCH~~	~~Turbo code~~	

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

NOTE 1: The exact physical layer encoding/decoding capabilities for different code types are FFS.
NOTE 2: In the UE the channel coding capability should be linked to the terminal class.
<Editor's note: Removal of 1/2 Turbo code rate is a working assumption.>

-------------------- End text proposal --------------------

-------------------- Start text proposal --------------------

4.2.3.2.1. Turbo coder

NOTE: 4-state SCCC is not included in Release-99. It needs to be clarified from TSG-SA what are the service specifications with respect to different quality of services. The performance below BER of 10^{-6} need to be studied if there is a requirement for this quality of services of physical layer.

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

The 8-state PCCC and the 4-state SCCC -are described below.

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

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

where,

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

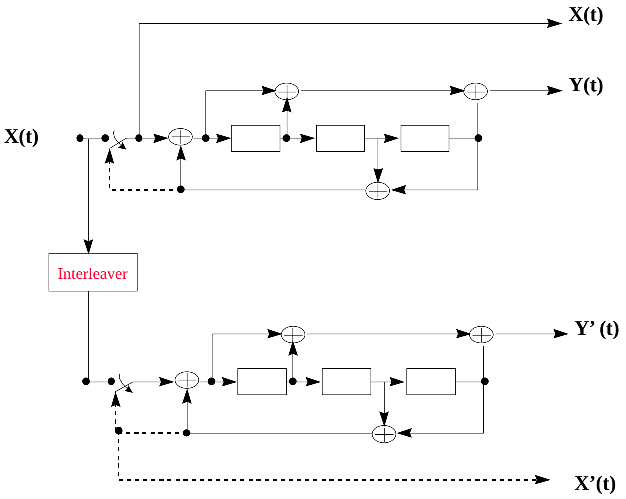

Figure 1: Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)

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

The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate 1/3~~ or 1/2~~. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc. ~~For rate 1/2, the parity bits produced by the constituent encoders are alternately punctured to produce the output sequence X(0), Y(0), X(1), Y'(1), X(2), Y(2), X(3), Y'(3), etc.~~

The SCCC is a rate 1/3 SCCC, The outer code of the SCCC is a rate 2/3 obtained by puncturing a rate ½ code with generating matrix

$$G^{(o)}(Z)=(1,(1+Z^2)/(1+Z+Z^2))$$

-------------------- End text proposal --------------------

-------------------- Start text proposal --------------------

4.2.6. Rate matching

[Editors' note: Rate matching for Turbo codes is a working assumption.]

Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. 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 transport channel can vary between different transmission time intervals. In the downlink the transmission is interrupted if the number of bits is lower than maximum. When the number of bits between different transmission time intervals in uplink is changed, bits are repeated or punctured to ensure that the total bit rate after second multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

Notation used in Section 4.2.6 and subsections:

N_{ij}: For uplink : Number of bits in a radio frame before rate matching on transport channel *i* with transport format combination *j* .
For downlink : An intermediate calculation variable (not a integer but a multiple of 1/8).

N_{il}^{TTI} : Number of bits in a transmission time interval before rate matching on TrCH *i* with transport format *l*.

ΔN_{ij} : For uplink : If positive - number of bits that should be repeated in each radio frame on TrCH *i* with transport format combination *j*.
If negative - number of bits that should be punctured in each radio frame on TrCH *i* with transport format combination *j*.

For downlink : An intermediate calculation variable (not a integer but a multiple of 1/8).

ΔN_{il}^{TTI} : If positive - number of bits to be repeated in each transmission time interval on TrCH *i* with transport format *l*.
If negative - number of bits to be punctured in each transmission time interval on TrCH *i* with transport format *l*.

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 avoid multicode or to enable the use of a higher spreading factor. Signalled from higher layers.

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

I: Number of TrCHs in the CCTrCH.

Z_{ij}: Intermediate calculation variable.

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

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

q: Average puncturing distance. Used in uplink only.

$I_F(n_i)$: 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). Used in uplink only.

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

$TF_i(j)$: Transport format of TrCH i for the transport format combination j.
TFS(i): The set of transport format indexes *l* for TrCH i.
TFCS: The set of transport format combination indexes *j*.
e_{ini}: Initial value of variable *e* in the rate matching pattern determination algorithm of section 4.2.6.3.
X: Systematic bit in 4.2.3.2.1.
Y: 1st parity bit (from the upper Turbo constituent encoder) in section 4.2.3.2.1.
Y': 2nd parity bit (from the lower Turbo constituent encoder) 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.

The * (star) notation is used to replace an index *x* when the indexed variable X_x does not depend on the index *x*. In the left wing $X_* = Y$” is equivalent to “**for all** *x* **do** $X_x = Y$ ”. In the right wing of an assignment, the meaning is that “$Y = X_*$ ” is equivalent to “**take any** *x* **and do** $Y = X_x$ ”

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

$Z_{0,j} = 0$

$$Z_{ij} = \left\lfloor \frac{\sum_{m=1}^{i} RM_m \cdot N_{mj}}{\sum_{m=1}^{I} RM_m \cdot N_{mj}} \cdot N_{data,j} \right\rfloor$$ for all *i* = 1 .. *I, where ë û means round downwards*

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

4.2.6.1. Determination of rate matching parameters in uplink

In uplink puncturing can be used to avoid multicode or to enable the use of a higher spreading factor when this is needed because the UE does not support SF down to 4. The maximum amount of puncturing that can be applied is signalled setup from higher layers and denoted by *PL*. The number of available bits in the radio frames for all possible spreading factors is given in [2]. Denote these values by N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, and N_4, where the index refers to the spreading factor. The possible values of N_{data} then are *{* N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, $N_{4,}$ $2N_4$, $3N_4$, $4N_4$, $5N_4$, $6N_4$*}*.Depending on the UE capabilities, the supported set of $\mathrm{N}_{\mathrm{data}}$, denoted SET0, can be a subset of *{* N_{256}, N_{128}, N_{64}, N_{32}, N_{16}, N_8, $N_{4,}$ $2N_4$, $3N_4$, $4N_4$, $5N_4$, $6N_4$*}*. $N_{data,\,j}$ for the transport format combination j is determined by executing the following algorithm:

```
SET1 = { $N_{data}$ in SET0 such that $N_{data} - \sum_{x=1}^{I} \frac{RM_{x,}}{\min_{1\le y\le I}\{RM_y\}} \cdot N_{x,j}$ is non negative }
If the smallest element of SET1 requires just one PhCH then
    $N_{data,j}$ = min SET1
else
    SET2 = { $N_{data}$ 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 }
    Sort SET2 in ascending order
    $N_{data}$ = min SET2
    While $N_{data}$ is not the max of SET2 and the follower of $N_{data}$ requires no additional PhCH do
        $N_{data}$ = follower of $N_{data}$ in SET2
    End while
    $N_{data,j}$ = $N_{data}$
End if
```

The number of bits to be repeated or punctured, DN_{ij}, within one radio frame for each TrCH *i* is calculated with the relations given in Section 4.2.6 for all possible transport format combinations *j* and selected every radio frame. Additionally for determining e_{ini}, the following parameters are needed:

For convolutonal codes,
a=2 for the rate matching algorithm in section 4.2.6.4.

q= ëN_{ij} */(*ôDN_{ij}ô*)*û *, where ë û means round downwards and ô ô means absolute value.*

```
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
```

```
        $S(I_F$ (éx*q'ù mod $F_i$)) = (éx*q'ù div $F_i$)
end for
```

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 4.2.6.~~3~~4, where:
$DN = DN_{i,j}$
$N = N_{i,j}$, *and*
e_{ini} = (~~2~~a·S(n_i)·|ΔN| + N) mod ~~2~~a·N, if e_{ini} = 0 then e_{ini} = a·N.

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 puncturing is to be performed, parameters are as follows.

a=2 for Y parity sequence, and
a=1 for Y' parity sequence.

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 4.2.6.4, where:

$$DN = \begin{cases} \lfloor DN_{i,j}/2 \rfloor \text{for Y sequence} \\ \lceil DN_{i,j}/2 \rceil \text{for Y' sequence} \end{cases}$$

N = ë$N_{i,j}$/3û ,

q = ëN/|DN| û

```
if($q \le 2$)
  for $x$=0 to $F_i$-1
    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 = é x*q'ù mod $F_i$;
    if(Y sequence)
      $S[I_F$[(3r+1) mod $F_i$]] = é x*q'ù div $F_i$;
    if(Y' sequence)
      $S[I_F$[(3r+2) mod $F_i$ ]] = é x*q'ù div $F_i$;
  endfor
endif
```

e_{ini} = ($a{\times}S(n_i){\times}|DN|$ + N) mod $a{\times}N$, if e_{ini} =0 then e_{ini} = $a{\times}N$.

4.2.6.2. Determination of rate matching parameters in downlink

For downlink $N_{data,j}$ does not depend on the transport format combination j. $N_{data,j}$ is given by the channelization code(s) assigned by higher layers.

4.2.6.2.1. Determination of rate matching parameters for fixed positions of TrCHs

First an intermediate calculation variable $N_{i,*}$ is calculated for all transport channels *i* by the following formula :

$$N_{i,*} = \frac{1}{F_i} \cdot \max_{l \in TFS(i)} N_{i,l}^{TTI}$$

The computation of the $\Delta N_{i,l}^{TTI}$ parameters is then performed in for all TrCH *i* and all TF *l* by the following formula, where $\Delta N_{i,*}$ is derived from $N_{i,*}$ by the formula given at section 4.2.6:

$$\Delta N_{i,*}^{TTI} = F_i \cdot \Delta N_{i,*}$$

Note : the order in which the transport format combinations are checked does not change the final result.

For each transmission time interval of TrCH *i* with TF *l*, the rate-matching pattern is calculated with the algorithm in Section 4.2.6.~~3~~4. The following parameters are used as input:

For convolutional codes,

$$\Delta N = \Delta N_{i,*}^{TTI}$$

$$N = N_{il}^{TTI}$$

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

a=2 for the rate matching algorithm.

For turbo codes, if repetition is to be performed, such as *DN>0*, parameters for turbo codes are the same as parameter for convolutional codes. If puncturing is to be performed, parameters are as follows.

a=2 for Y sequence,
a=1 for Y' sequence.

Systematic bit should not be punctured.

$$DN = \begin{cases} \left\lfloor DN_{i,*}^{TTI} / 2 \right\rfloor \text{for Y sequence} \\ \left\lceil DN_{i,*}^{TTI} / 2 \right\rceil \text{for Y' sequence} \end{cases}$$

$$N = \left\lfloor N_{il}^{TTL} / 3 \right\rfloor$$

$$\max_{l \in TFS(i)} \left\lfloor N_{il}^{TTL} / 3 \right\rfloor$$

$$e_{ini} = \max_{l \in TFS(i)} \left\lfloor N_{il}^{TTL} / 3 \right\rfloor \bmod a \times \max_{l \in TFS(i)} \left\lfloor N_{il}^{TTL} / 3 \right\rfloor.$$

4.2.6.2.2. Determination of rate matching parameters for flexible positions of TrCHs

First an intermediate calculation variable N_{ij} is calculated for all transport channels *i* and all transport format combinations *j* by the following formula :

$$N_{i,j} = \frac{1}{F_i} \cdot N_{i,TF_i(j)}^{TTI}$$

Then rate matching ratios RF_i are calculated for each the transport channel i in order to minimise the number of DTX bits when the bit rate of the CCTrCH is maximum. The RF_i ratios are defined by the following formula :

$$RF_i = \frac{N_{data,*}}{\max\limits_{j \in TFCS} \sum_{i=1}^{i=I} \left(RM_i \cdot N_{i,j} \right)} \cdot RM_i$$

The computation of $\Delta N_{i,l}^{TTI}$ parameters is then performed in two phases. In a first phase, tentative temporary values of $\Delta N_{i,l}^{TTI}$ are computed, and in the second phase they are checked and corrected. The first phase, by use of the RF_i ratios, ensures that the number of DTX indication bits inserted is minimum when the CCTrCH bit rate is maximum, but it does not ensure that the maximum CCTrCH bit rate is not greater than $N_{data,*}$. per 10ms. The latter condition is ensured through the checking and possible corrections carried out in the second phase.

At the end of the second phase, the latest value of $\Delta N_{i,l}^{TTI}$ is the definitive value.

The first phase defines the tentative temporary $\Delta N_{i,l}^{TTI}$ for all transport channel i and any of its transport format l by use of the following formula :

$$\Delta N_{i,l}^{TTI} = F_i \cdot \left\lceil \frac{RF_i \cdot N_{i,l}^{TTI}}{F_i} \right\rceil - N_{i,l}^{TTI}$$

The second phase is defined by the following algorithm :

```
for all $j$ in TFCS do                                                          -- for all TFC
    $D = \sum_{i=1}^{i=I} \frac{N_{i,TF_i(j)}^{TTI} + \Delta N_{i,TF_i(j)}^{TTI}}{F_i}$          -- CCTrCH bit rate (bits per 10ms) for TFC $l$
    if $D > N_{data,*}$ then
        for $i$ = 1 to $I$ do                                                   -- for all TrCH
            $\Delta N = F_i \cdot \Delta N_{i,j}$                                  -- $\Delta N_{i,j}$ is derived from $N_{i,j}$ by the formula given at section 4.2.6
            if $\Delta N_{i,TF_i(j)}^{TTI} > \Delta N$ then
                $\Delta N_{i,TF_i(j)}^{TTI} = \Delta N$
            end-if
        end-for
    end-if
end-for
```

Note : the order in which the transport format combinations are checked does not change the final result.

For each transmission time interval of TrCH i with TF l, the rate-matching pattern is calculated with the algorithm in Section 0. The following parameters are used as input:

For convolutional codes,

$DN = DN_{il}^{TTI}$

$N = N_{il}^{TTI}$

$e_{ini} = N_{il}^{TTI}$

a=2 for the rate matching algorithm.

For turbo codes, if repetition is to be performed, such as *DN>0*, parameters for turbo codes are the same as parameter for convolutional codes. If puncturing is to be performed, parameters are as follows.

a=2 for Y sequence,
a=1 for Y’ sequence.

Systematic bit should not be punctured.

$$DN = \begin{cases} \left\lfloor DN_{il}^{TTI} / 2 \right\rfloor \text{for Y sequence} \\ \left\lceil DN_{il}^{TTI} / 2 \right\rceil \text{for Y' sequence} \end{cases}$$

$$N = \left\lfloor N_{il}^{TTI} / 3 \right\rfloor,$$

$$e_{ini} = N_{il}^{TTI} \bmod a \times N_{il}^{TTI},$$

4.2.6.3. Bit separation for rate matching

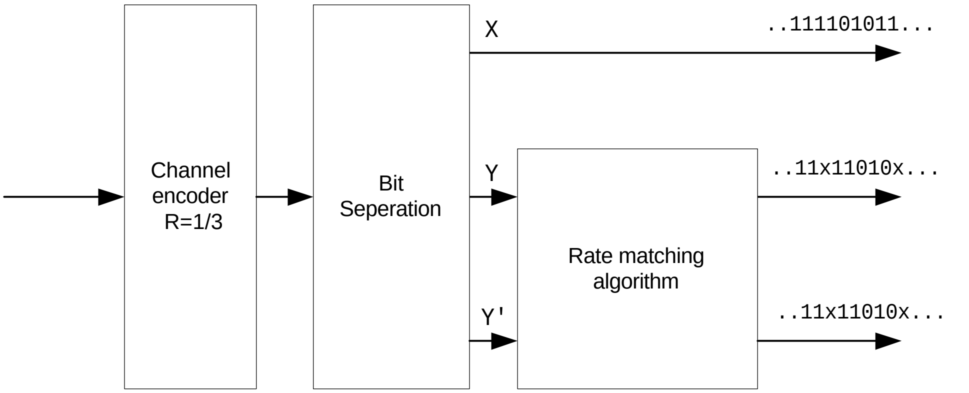

Figure 4-1. Overall rate matching block diagram before first interleaving where x denotes punctured bit.

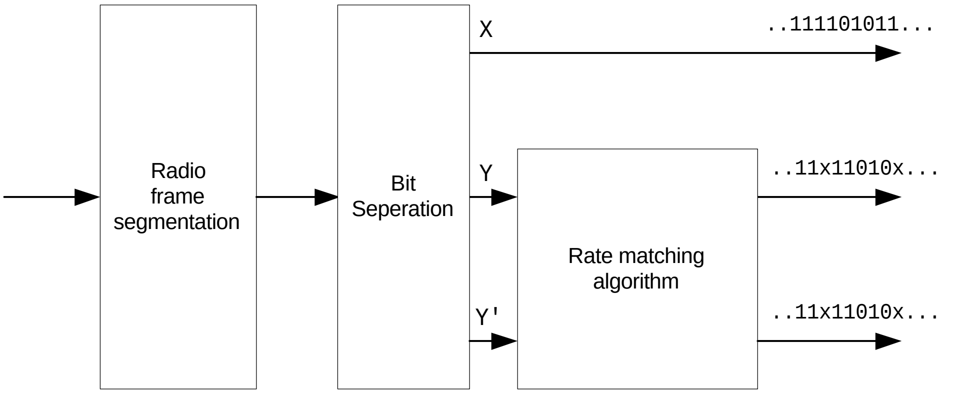

Figure 4-2. Overall rate matching block diagram after first interleaving where x denotes punctured bit.

Rate matching puncturing for Turbo codes in uplink 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.

For uplink, there are two different alternation patterns in bit stream from Radio frame segmentation according to the TTI of a TrCH as shown in Table 1.

Table 1 Alternation patterns of bits from radio frame segmentation in uplink

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 2 shows the initial parity type of each radio frame of a TrCH with TTI = {10, 20, 40, 80} msec.

Table 2 Initial parity type of radio frames of TrCH in uplink

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*

Table 1 and 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 Table 1 and 2 according to the TTI and n_i of a TrCH.

Rate matching puncturing for Turbo codes in downlink is applied separately to Y and Y’s sequences. No puncturing is applied to X sequence. Therefore, it is necessary to separate X, Y, and Y’ sequences before rate matching is applied.

For downlink, output bit sequence pattern from Turbo encoder is always X, Y, Y’, X, Y, Y’, …. Therefore, bit separation is achieved with the alternative selection of bits from Turbo encoder.

~~4.2.6.3.~~4.2.6.4. Rate matching pattern determination

Denote the bits before rate matching by:
x_{i1}, x_{i2}, x_{i3}, ... x_{iN} , where i is the TrCH number and N is the ~~number of bits before rate matching~~ parameter given in section 4.2.6.2.
The rate matching rule is as follows:

```
if puncturing is to be performed

        $y = -DN$
        $e = e_{ini}$                    -- initial error between current and desired puncturing ratio
        $m = 1$                          -- index of current bit
        do while $m <= N$
                $e = e - ~~2~~a * y$             -- update error
                if $e <= 0$ then                  -- check if bit number m should be punctured
                        puncture bit $x_{i,m}$
                        $e = e + ~~2~~a*N$        -- update error
                end if
                $m = m + 1$                       -- next bit
        end do
else
        $y = DN$
        $e = e_{ini}$                    -- initial error between current and desired puncturing ratio
        $m = 1$                          -- index of current bit
        do while m <= N
                e = e –~~2~~a* y                        -- update error
                do while e <= 0                 -- check if bit number m should be repeated
                        repeat bit $x_{i,m}$
                        e = e +  ~~2~~a*N                  -- update error
                enddo
                m = m + 1                                       --  next bit
        end do
end if
```

A repeated bit is placed directly after the original one.

3. Text proposal for TS 25.222

-------------------- Start text proposal --------------------

6.2.3. Channel coding

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

$$c_{ik} = x_{i1k} \qquad k = 1, 2, \ldots, X_i$$

$$c_{ik} = x_{i,2,(k-X_i)} \qquad k = X_i + 1, X_i + 2, \ldots, 2X_i$$

$$c_{ik} = x_{i,3,(k-2X_i)} \qquad k = 2X_i + 1, 2X_i + 2, \ldots, 3X_i$$

$\ldots$

$$c_{ik} = x_{i,C_i,(k-(C_i-1)X_i)} \qquad k = (C_i - 1)X_i + 1, (C_i - 1)X_i + 2, \ldots, C_iX_i$$

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

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

- Convolutional coding
- Turbo coding
- No channel coding

Table 6.2.3-1 Error Correction Coding Parameters

~~Transport channel type~~	~~Coding scheme~~	~~Coding rate~~
~~BCH~~	~~Convolutional code~~	~~1/2~~
~~PCH~~		
~~FACH~~		
~~RACH~~		~~1/2, [2/3, 7/8]~~ *~~the values in square brackets have not yet been approved.>~~*
~~DCH~~		~~1/2, 1/3 or no coding~~
~~DCH~~	~~Turbo code~~	

Transport channel type	**Coding scheme**	**Coding rate**
BCH	Convolutional code	1/2
PCH		
FACH		
RACH		
DCH		1/3, 1/2, or no coding
DCH	Turbo code	1/3, or no coding

Note 1: The exact physical layer encoding/decoding capabilities for different code types are FFS.

Note 2: In the UE the channel coding capability should be linked to the terminal class.

<Editor's note: Removal of 1/2 Turbo code rate is a working assumption.>

-------------------- End text proposal --------------------

-------------------- Start text proposal --------------------

6.2.3.2.1. Turbo coder

<Note: It needs to be clarified from TSG SA what are the service specifications with respect to different qualities of service. The performance below BER of 10^{-6} needs to be studied if there is a requirement for this quality of services over the physical layer.>

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

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

where,

$$\mathrm{d(D)}=1+D^2+D^3$$
$$\mathrm{n(D)}=1+D+D^3.$$

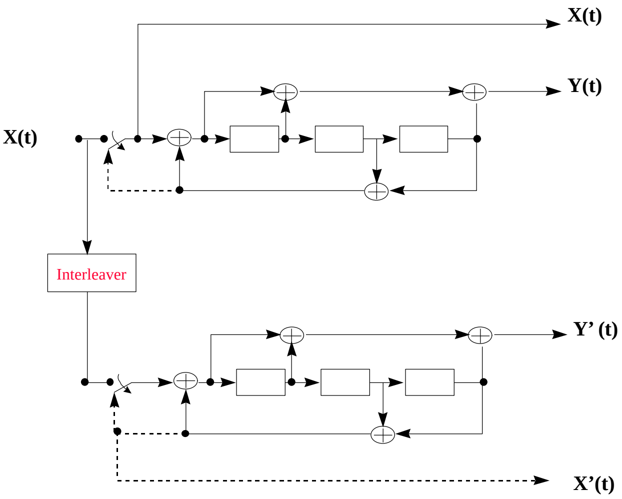

Figure 6-3. Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)

The initial value of the shift registers of the PCCC encoder shall be all zeros.
The output of the PCCC encoder is punctured to produce coded bits corresponding to the desired code rate 1/3~~ or 1/2~~. For rate 1/3, none of the systematic or parity bits are punctured, and the output sequence is X(0), Y(0), Y'(0), X(1), Y(1), Y'(1), etc. ~~For rate 1/2, the parity bits produced by the constituent encoders are alternately punctured to produce the output sequence X(0), Y(0), X(1), Y'(1), X(2), Y(2), X(3), Y'(3), etc.~~

--------------------- End proposal ---------------------

--------------------- Start text proposal ---------------------

6.2.7. Rate matching

[Editors' note: Rate matching for Turbo codes is a working assumption.]

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. In the downlink the transmission is interrupted if the number of bits is lower than maximum. When the number of bits between different transmission time intervals in uplink is changed, bits are repeated or punctured to ensure that the total bit rate after second multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

Notation used in Section 6.2.7 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 that should be repeated in each radio frame on TrCH *i* with transport format combination *j*.

If negative - number of bits that should 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 avoid multicode or to enable the use of a higher spreading factor. Signalled from higher layers.

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

I: Number of TrCHs in the CCTrCH.

Z_{ij}: Intermediate calculation variable.

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

n_i: Radio frame number in the transmission time interval of TrCH *i* (*0 £ 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 6.2.7.3.
X: Systematic bit in 6.2.3.2.1.
Y: 1st parity bit (from the upper Turbo constituent encoder) in section 6.2.3.2.1.
Y': 2nd parity bit (from the lower Turbo constituent encoder) in section 6.2.3.2.1.
Note: Time index t in 6.2.3.2.1 is omitted for simplify the rate matching description.

6.2.7.1. Determination of rate matching parameters

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

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

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

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

Puncturing can be used to minimise the number of 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 uplink and downlink depend on the dedicated physical channels which are assigned to the link, 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 $N_{data} - PL \cdot \sum_{i=1}^{T} \frac{RM_i}{\min_l \{RM_l\}} \cdot N_{ij}$ is non negative }

$N_{data,j}$ = min SET1

The number of bits to be repeated or punctured, *DN_{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. For each radio frame, the rate-matching pattern is calculated with the algorithm in Section 6.2.7.2, where *DN* = *DN_{ij}* and *N* = *N_{ij}*. Additionally for determining e_{ini}, the following parameters are needed:

For convolutonal codes,
a=2 for the rate matching algorithm in section 6.2.7.3.

q= ëN_{ij} /(ôDN_{ij}ô)û , where ë û means round downwards and ô ômeans absolute value.

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
        $S(I_F$ (éx*q'ù mod $F_i$)) = (éx*q'ù div $F_i$)
end for
```

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 6.2.7.3, where:

$DN = DN_{i,j}$

$N = N_{i,j}$, *and*

$e_{ini} = (a·S(n_i)·|DN| + N) \bmod a·N$, *if* $e_{ini} = 0$ *then* $e_{ini} = a·N$.

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 puncturing is to be performed, parameters are as follows.

a = 2 for Y parity sequence, and
a = 1 for Y' parity sequence.

For each radio frame, the rate-matching pattern is calculated with the algorithm in section 6.2.7.3, where:

$$DN = \begin{cases} \lfloor DN_{i,j}/2 \rfloor & \text{for Y sequence} \\ \lceil DN_{i,j}/2 \rceil & \text{for Y' sequence} \end{cases}$$

N = ë$N_{i,j}$/3û ,

q = ëN/|DN| û

```
if($q \le 2$)
   for $x$=0 to $F_i$-1
     if(Y sequence)
        $S[I_F[(3x+1) \bmod F_i]] = x \bmod 2$;
     if(Y' sequence)
           $S[I_F[(3x+2) \bmod F_i]] = x \bmod 2$;
   end for
else
   if $q$ is even
     then  $q' = q - gcd(q, F_i)/F_i$  -- where $gcd(q, F_i)$ means greatest common divisor of $q$ and $F_i$
                                   -- note that $q'$ is not an integer, but a multiple of 1/8
   else        $q' = q$
   endif
   for $x$=0 to $F_i$ -1
     $r$ = é x*q'ù mod $F_i$;
     if(Y sequence)
        $S[I_F[(3r+1) \bmod F_i]]$ = é x*q'ù div $F_i$;
     if(Y' sequence)
        $S[I_F[(3r+2) \bmod F_i]]$ = é x*q'ù div $F_i$;
   endfor
endif
```

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

6.2.7.2. Bit separation for rate matching

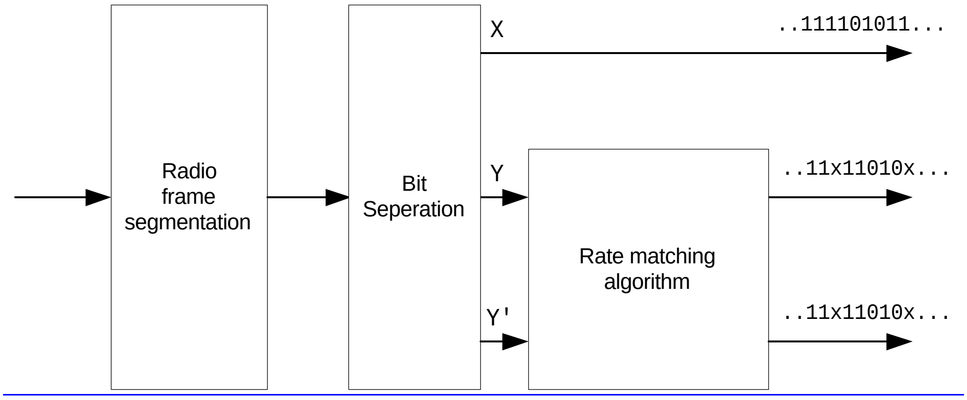

Figure 6-1. Overall rate matching block diagram after first interleaving where x denotes punctured bit.

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 1.

Table 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 2 shows the initial parity type of each radio frame of a TrCH with TTI = {10, 20, 40, 80} msec.

Table 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*

Table 1 and 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 Table 1 and 2 according to the TTI and n_i of a TrCH.

~~6.2.6.2.~~6.2.7.3. Rate matching pattern determination

Denote the bits before rate matching by:
x_{i1}, x_{i2}, x_{i3}, ... x_{iN} , where i is the TrCH number and N is the ~~number of bits before rate matching~~ parameter given in section 6.2.7.1.
The rate matching rule is as follows:

```
if puncturing is to be performed

        y = -DN
        e = e_ini                    -- initial error between current and desired puncturing ratio
        m = 1                        -- index of current bit
        do while m <= N
                e = e – ~~2~~ a * y              -- update error
                if e <= 0 then                  -- check if bit number m should be punctured
                        puncture bit x_i,m
                        e = e + ~~2~~ a *N       -- update error
                end if
                m = m + 1                       --  next bit
        end do
else
        y = DN
        e = e_ini                    -- initial error between current and desired puncturing ratio
        m = 1                        -- index of current bit
        do while m <= N
                e = e – ~~2~~ a * y              -- update error
                do while e <= 0                 -- check if bit number m should be repeated
                        repeat bit x_i,m
                        e = e + ~~2~~ a *N       -- update error
                enddo
                m = m + 1                       --  next bit
        end do
end if
```

A repeated bit is placed directly after the original one.

4. References

[1] “Text proposal for rate matching algorithm for turbo codes in TS25.212, TS25.222”, Samsung Electronics Co. and LGIC, WG1#7, R1-99b56.

[2] TS 25.212 (V2.0.1): “Multiplexing and channel coding (FDD)”

[3] TS 25.222 (V2.1.1): “Multiplexing and channel coding (TDD)”