TSG-RAN Working Group 1 meeting #9
Dresden, Germany
November 30 – December 3, 1999

**TSGR1#9(99)i56**

**Agenda item:**

**Source:** Ericsson

**Title:** CR 25.212-012: Zero length transport blocks

**Document for:** Decision

---

The coding and multiplexing needs to handle transport blocks of length zero, i.e. the case when no data arrives for transmission in the TTI. Efficient handling of zero length transport blocks is important, but exactly how to handle them is not clearly specified in all parts of TS 25.212.

Another scenario can be a whole CCTrCH that has zero length transport blocks on all TrCHs. Also this case needs to be handled.

Clearly, one do not want to send any bits at all when there is no data to be transmitted, i.e. no CRC shall be attached, no tail bits shall be added, no coding shall be performed, etc.

The impact on the downlink and uplink functions for the cases with zero length transport blocks is listed below.

Downlink:

- **CRC attachment**: If no input bits are available ($A_i = 0$), the function shall be transparent ($B_i = 0$).
- **TrBk concatenation/code block segmentation**: The function can handle the case of no input bits.
- **Channel coding**: If no input bits are available ($Y_i = 0$), the function shall be transparent ($E_i = 0$).
- **Rate matching**: The function can handle the case of TrCHs having zero length transport blocks as long as not all TrCHs within the CCTrCH have zero length. If no input bits are available from all TrCHs within the CCTrCH, the function shall be transparent
- **Insertion of DTX indication with fixed positions**: The function can handle the case of no input bits.
- **1st interleaving**: The function can handle the case of no input bits which is possible for flexible positions.
- **Radio frame segmentation**: The function can handle the case of no input bits which is possible for flexible positions.
- **TrCH multiplexing**: The function can handle the special case for all TrCHs within a CCTrCH having zero bits which is possible for flexible positions.
- **Insertion of DTX indication with flexible positions**: The function can handle the case of no input bits.
- **Physical channel segmentation**: No impact since there will always be at least DTX indication bits.
- **2nd interleaving**: No impact since there will always be at least DTX indication bits.
- **Physical channel mapping**: The function can handle the case of CCTrCHs containing only DTX indication bits.

Uplink:

- **CRC**: If no input bits are available ($A_i = 0$), the function shall be transparent ($B_i = 0$).

- **TrBk concatenation/code block segmentation**: The function can handle the case of no input bits.
- **Channel coding**: If no input bits are available ($Y_i = 0$), the function shall be transparent ($E_i = 0$).
- **Radio frame size equalisation**: The function can handle the special case of no input bits.
- **1st interleaving**: The function can handle the special case of no input bits.
- **Radio frame segmentation**: The function can handle the special case of no input bits.
- **Rate matching**: The function can handle the case of TrCHs having zero length transport blocks as long as not all TrCHs within the CCTrCH have zero length. If no input bits are available from all TrCHs within the CCTrCH, the function shall be transparent and no uplink physical channel shall be selected.
- **TrCH multiplexing**: The function can handle the special case of all TrCHs within a CCTrCH having zero length.
- **Physical channel segmentation**: The function can handle the special case of all TrCHs within a CCTrCH having zero length.
- **2nd interleaving**: The function can handle the special case of all TrCHs within a CCTrCH having zero length.
- **Physical channel mapping**: The function can handle the special case of all TrCHs within a CCTrCH having zero length, i.e. no uplink physical channel is selected in this case.

From the above list, it can be noted that some additional information is needed in 25.212 to handle the special cases with zero length transport blocks.

It is proposed to include the case of zero length transport blocks into the descriptions of CRC attachment (4.2.1.1), channel coding (4.2.3) and rate matching (4.2.7).

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

***Document ???99???***
*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.*

**25.212 CR 012** 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 | b | | *(for SMG use only)* |
| *list expected approval meeting # here ↑* | For information | | non-strategic | | |

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

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

**Source:** Ericsson **Date:** 1999-11-24

**Subject:** Zero length transport blocks

**Work item:**

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

**Reason for change:** The handling of zero length transport blocks is not described in all parts of the current version of TS 25.212.

**Clauses affected:** 4.2.1.1, 4.2.3, 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:**

help.doc

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

## 4.2.1 Error detection

Error detection is provided on transport blocks through a Cyclic Redundancy Check. The CRC is 24, 16, 12, 8 or 0 bits and it is signalled from higher layers what CRC length that should be used for each TrCH.

### 4.2.1.1 CRC Calculation

The entire transport block is used to calculate the CRC parity bits for each transport block. The parity bits are generated by one of the following cyclic generator polynomials:

$$g_{CRC24}(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$$

$$g_{CRC16}(D) = D^{16} + D^{12} + D^5 + 1$$

$$g_{CRC12}(D) = D^{12} + D^{11} + D^3 + D^2 + D + 1$$

$$g_{CRC8}(D) = D^8 + D^7 + D^4 + D^3 + D + 1$$

Denote the bits in a transport block delivered to layer 1 by $a_{im1}, a_{im2}, a_{im3}, \ldots, a_{imA_i}$, and the parity bits by $p_{im1}, p_{im2}, p_{im3}, \ldots, p_{imL_i}$. $A_i$ is the length of a transport block of TrCH *i*, *m* is the transport block number, and $L_i$ is 24, 16, 12, 8, or 0 depending on what is signalled from higher layers.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial

$$a_{im1}D^{A_i+23} + a_{im2}D^{A_i+22} + \ldots + a_{imA_i}D^{24} + p_{im1}D^{23} + p_{im2}D^{22} + \ldots + p_{im23}D^1 + p_{im24}$$

yields a remainder equal to 0 when divided by $g_{CRC24}(D)$, polynomial

$$a_{im1}D^{A_i+15} + a_{im2}D^{A_i+14} + \ldots + a_{imA_i}D^{16} + p_{im1}D^{15} + p_{im2}D^{14} + \ldots + p_{im15}D^1 + p_{im16}$$

yields a remainder equal to 0 when divided by $g_{CRC16}(D)$, polynomial

$$a_{im1}D^{A_i+11} + a_{im2}D^{A_i+10} + \ldots + a_{imA_i}D^{12} + p_{im1}D^{11} + p_{im2}D^{10} + \ldots + p_{im11}D^1 + p_{im12}$$

yields a remainder equal to 0 when divided by $g_{CRC12}(D)$ and polynomial

$$a_{im1}D^{A_i+7} + a_{im2}D^{A_i+6} + \ldots + a_{imA_i}D^{8} + p_{im1}D^{7} + p_{im2}D^{6} + \ldots + p_{im7}D^1 + p_{im8}$$

yields a remainder equal to 0 when divided by $g_{CRC8}(D)$.

If no bits are input to the CRC calculation ($A_i = 0$), no CRC attachment shall be done, i.e. $B_i = 0$.

#### 4.2.1.1.1 Relation between input and output of the Cyclic Redundancy Check

The bits after CRC attachment are denoted by $b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$, where $B_i=A_i+L_i$. The relation between $a_{imk}$ and $b_{imk}$ is:

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

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

blocks on TrCH *i* is denoted by $M_i$. The bits after concatenation are denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where *i* is the TrCH number and $X_i = M_i B_i$. They are defined by the following relations:

$x_{ik} = b_{i1k}$ $k = 1, 2, \ldots, B_i$

$x_{ik} = b_{i,2,(k-B_i)}$ $k = B_i + 1, B_i + 2, \ldots, 2B_i$

$x_{ik} = b_{i,3,(k-2B_i)}$ $k = 2B_i + 1, 2B_i + 2, \ldots, 3B_i$

…

$x_{ik} = b_{i,M_i,(k-(M_i-1)B_i)}$ $k = (M_i - 1)B_i + 1, (M_i - 1)B_i + 2, \ldots, M_i B_i$

#### 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 last block. The filler bits are transmitted and they are always set to 0. The maximum code block sizes are:

convolutional coding: $Z = 504$

turbo coding: $Z = 5114$

no channel coding: $Z$ = *unlimited*

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

Number of code blocks: $C_i = éX_i / Zù$

Number of bits in each code block: $K_i = éX_i / C_iù$

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

If $X_i \leq Z$, then $o_{i1k} = x_{ik}$, and $K_i = X_i$.

If $X_i \geq Z$, then

$o_{i1k} = x_{ik}$ $k = 1, 2, \ldots, K_i$

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

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

…

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

$o_{iC_ik} = 0$ $k = (K_i - Y_i) + 1, (K_i - Y_i) + 2, \ldots, K_I$

### 4.2.3 Channel coding

Code blocks are delivered to the channel coding block. They are denoted by $o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$, where *i* is the TrCH number, *r* is the code block number, and $K_i$ is the number of bits in each code block. The number of code blocks on TrCH *i* is denoted by $C_i$. After encoding the bits are denoted by $y_{ir1}, y_{ir2}, y_{ir3}, \ldots, y_{irY_i}$. 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 $E_i = C_i Y_i$. The output bits are defined by the following relations:

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

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

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

$$\ldots$$

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

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 channel coding

The values of $Y_i$ in connection with each coding scheme:

- Convolutional coding, ½ rate: $Y_i = 2*K_i + 16$; *1/3 rate:* $Y_i = 3*K_i + 24$
- Turbo coding, 1/3 rate: $Y_i = 3*K_i + 12$
- No channel coding, $Y_i = K_i$

**Table 1: Error Correction Coding Parameters**

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

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

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where $i$ is the TrCH number and $X_i$ is the number bits. The $Fi$ output bit sequences per TTI are denoted by $y_{i,n_i1}, y_{i,n_i2}, y_{i,n_i3}, \ldots, y_{i,n_iY_i}$ where $n_i$ is the radio frame number in current TTI and $Y_i$ is the number of bits per radio frame for TrCH $i$. The output sequences are defined as follows:

$y_{i,n_ik} = x_{i,((n_i-1)Y_i)+k}$ , $n_i = 1\ldots F_i$, $j = 1\ldots Y_i$

where

$Y_i = (X_i / F_i)$ is the number of bits per segment,

$x_{ik}$ is the k^th bit of the input bit sequence and

$y_{i,n_ik}$ is the k^th bit of the output bit sequence corresponding to the n^th radio frame

The $n_i$ -th segment is mapped to the $n_i$ -th radio frame of the transmission time interval.

### 4.2.6.1 Relation between input and output of the radio frame segmentation block in uplink

The input bit sequence to the radio frame segmentation is denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$ , where $i$ is the TrCH number and $T_i$ the number of bits. Hence, $x_{ik} = d_{ik}$ and $X_i = T_i$.

The output bit sequence corresponding  radio frame $n_i$ is denoted by $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$ , where $i$ is the TrCH number and $N_i$ is the number of bits. Hence, $e_{i,k} = y_{i,n_ik}$ and $N_i = Y_i$.

### 4.2.6.2 Relation between input and output of the radio frame segmentation block in downlink

The bits input to the radio frame segmentation are denoted by $q_{i1}, q_{i2}, q_{i3}, \ldots, q_{iQ_i}$ , where $i$ is the TrCH number and $Q_i$ the number of bits. Hence, $x_{ik} = q_{ik}$ and $X_i = Q_i$.

The output bit sequence corresponding to radio frame $n_i$ is denoted by $f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}$ , where $i$ is the TrCH number and $V_i$ is the number of bits. Hence, $f_{i,k} = y_{i,n_ik}$ and $V_i = Y_i$.

## 4.2.7 Rate matching

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.

If no bits are input to the rate matching for all TrCHs within a CCTrCH, the rate matching shall output no bits for all TrCHs within the CCTrCH and no uplink physical channel will be selected in the case of uplink rate matching.

**Notation used in section 4.2.7 and subsections:**

$N_{ij}$: For uplink: Number of bits in a radio frame before rate matching on TrCH $i$ 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*. Used in downlink only.

$\Delta N_{ij}$ : 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 integer but a multiple of 1/8).

$\Delta N_{il}^{TTI}$ : If positive - number of bits to be repeated in each transmission time interval on TrCH *i* with transport format *j*.

If negative - number of bits to be punctured in each transmission time interval on TrCH *i* with transport format *j*.

Used in downlink only.

$RM_i$: Semi-static rate matching attribute for transport channel *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 4.2.7.4.

$e_{plus}$ Increment of variable *e* in the rate matching pattern determination algorithm of section 4.2.7.4.

$e_{minus}$ Decrement of variable *e* in the rate matching pattern determination algorithm of section 4.2.7.4.

*X:* Systematic bit in section 4.1.3.2.1.

*Y:* 1st parity bit (from the upper Turbo constituent encoder) in section 4.1.3.2.1.

*Y':* 2nd parity bit (from the lower Turbo constituent encoder) in section 4.1.3.2.1.

NOTE: Time index t in section 4.1.3.2.1 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 of an assignment the meaning is that "$X_* = Y$" is equivalent to "**for all** $\underline{x}$ **do** $X_x = Y$ ". In the right wing of an assignment, the meaning is that "$Y = X_*$ " is equivalent to "**take any** $\underline{x}$ **and do** $Y = X_x$ "

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

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

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

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