TSG-RAN Working Group1 meeting #7
30 August – 3 September 1999
Hanover, Germany

TSGR1#7(99)D87

Agenda item: **AH4 Text proposals**

Source: **QUALCOMM Europe**

Title: **Revised text proposal for radio frame equalization, 1st interleaving and radio frame segmentation for TDD (25.222)**

Document for: **Decision**

Introduction

This is an updated version of the text proposal presented in Tdoc R1-99B05 and endorsed in principle by AH04. The revision is mostly editorial so that the notation is aligned with the one suggested in Tdoc R1-99B23 and endorsed by AH04 and reflects TDD structure.

Text Proposal for inclusion in 25.222

6

6.2 Transport channel coding/multiplexing

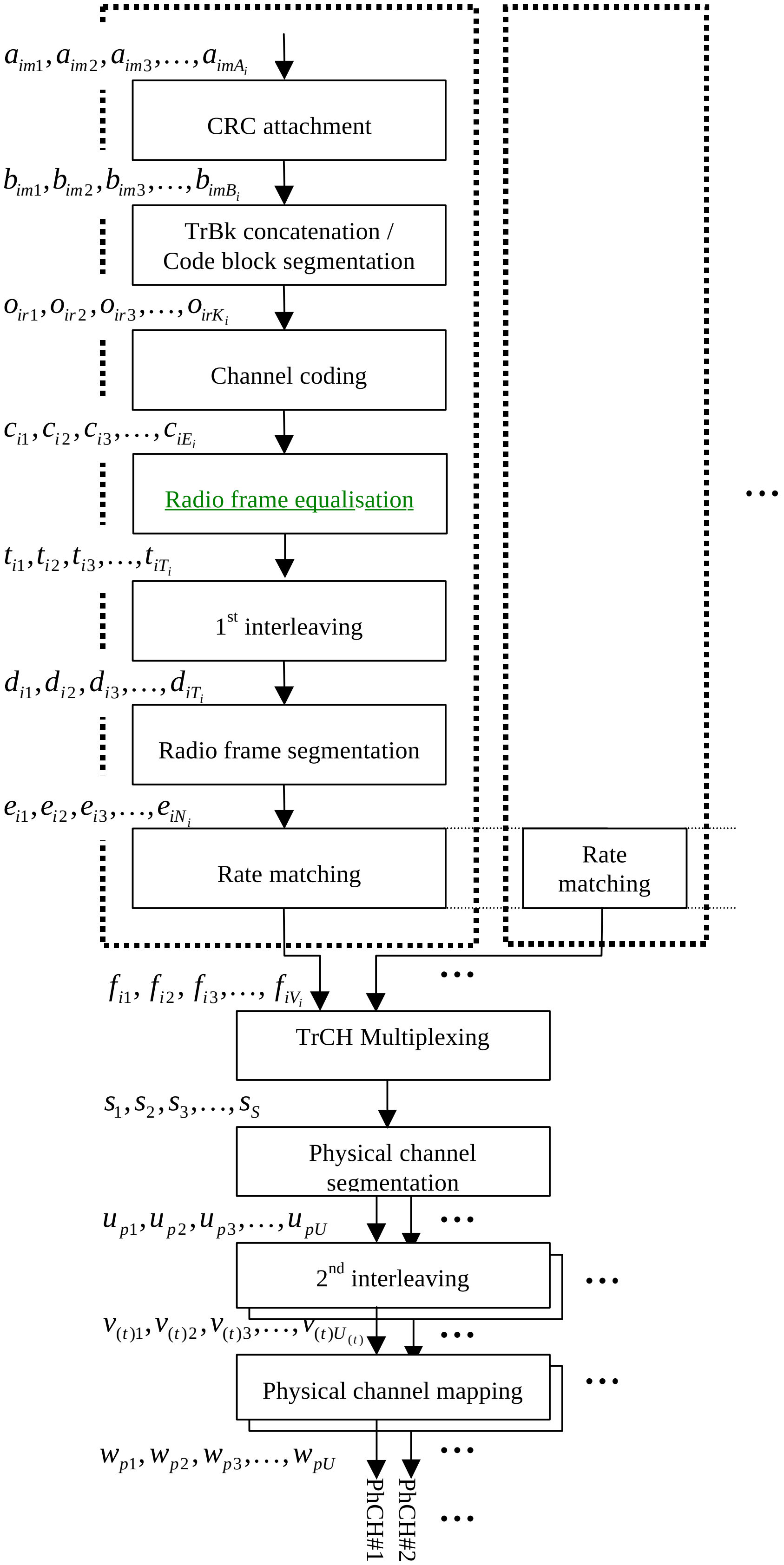
$a_{im1}, a_{im2}, a_{im3}, \ldots, a_{imA_i}$
CRC attachment
$b_{im1}, b_{im2}, b_{im3}, \ldots, b_{imB_i}$
TrBk concatenation / Code block segmentation
$o_{ir1}, o_{ir2}, o_{ir3}, \ldots, o_{irK_i}$
Channel coding
$c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$
Radio frame equalisation
$t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$
1st interleaving
$d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iT_i}$
Radio frame segmentation
$e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$
Rate matching
Rate matching
$f_{i1}, f_{i2}, f_{i3}, \ldots, f_{iV_i}$
TrCH Multiplexing
$s_1, s_2, s_3, \ldots, s_S$
Physical channel segmentation
$u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU}$
2nd interleaving
$v_{(t)1}, v_{(t)2}, v_{(t)3}, \ldots, v_{(t)U_{(t)}}$
Physical channel mapping
$w_{p1}, w_{p2}, w_{p3}, \ldots, w_{pU}$
PhCH#1
PhCH#2

Figure 6–1. Transport channel multiplexing structure for uplink and downlink

------------ snip -------------

6.2.4 Radio frame size equalisation

Radio frame size equalisation is padding the input bit sequence in order to ensure that the output can be segmented in F_i data segments of same size as described in the section 6.2.6.

The input bit sequence to the radio frame size equalisation is denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is TrCH number and E_i the number of bits. The output bit sequence is denoted by $t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iT_i}$, where T_i is the number of bits. The output bit sequence is derived as follows:

$t_{ik} = c_{ik}$, for k = 1… E_i and

$t_{ik} = \{0 \mid 1\}$ for k= E_i +1… T_i, if $E_i < T_i$

where

$T_i = F_i * N_i$ and

$N_i = \lfloor (E_i - 1)/F_i \rfloor + 1$ is the number of bits per segment after size equalisation.

6.2.56.2.4 1st interleaving

The 1st interleaving is a block interleaver with inter-column permutations. The input bit sequence to the 1st interleaver is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$, where i is TrCH number and X_i the number of bits (at this stage X_i is assumed and guaranteed to be an integer multiple of TTI). The output bit sequence is derived as follows:

(1) Select the number of columns C_I from Table 6.2.5-1.

(2) Determine the number of rows R_I defined as $R_I = X_i/C_I$

(3) Write the input bit sequence into the $R_I \times C_I$ rectangular matrix row by row starting with bit $x_{i,1}$ in the first column of the first row and ending with bit $x_{i,(R_I C_I)}$ in column C_I of row R_I:

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

(4) Perform the inter-column permutation based on the pattern $\{P_1(j)\}$ (j=0,1, ..., C-1) shown in Table 4-3, where $P_1(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{ik}:

$$\begin{bmatrix} y_{i1} & y_{i,(R_I+1)} & y_{i,(2R_I+1)} & \cdots y_{i,((C_I-1)R_I+1)} \\ y_{i2} & y_{i,(R_I+2)} & y_{i,(2R_I+2)} & \cdots y_{i,((C_I-1)R_I+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{iR_I} & y_{i,(2R_I)} & y_{i,(3R_I)} & \cdots & y_{i,(C_I R_I)} \end{bmatrix}$$

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

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

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

The 1st interleaving is a block interleaver with inter-column permutations. The bits input to the 1st interleaving are denoted by $c_{i1}, c_{i2}, c_{i3}, \ldots, c_{iE_i}$, where i is the TrCH number and E_i the number of bits. The following steps are applied:

(1) Select the number of columns C_1 from Table 6.2.4-1.

(2)Determine the number of rows R_1 by finding minimum integer R_1 such that,

$$E_i \leq R_1 \times C_1.$$

(3)The bits input to the 1st interleaving are written into the $R_1 \times C_1$ rectangular matrix row by row.

$$\begin{bmatrix} c_{i1} & c_{i2} & c_{i3} & \cdots & c_{iC_1} \\ c_{i,(C_1+1)} & c_{i,(C_1+2)} & c_{i,(C_1+3)} & \cdots & c_{i,(2C_1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c_{i,((R_1-1)C_1+1)} & c_{i,((R_1-1)C_1+2)} & c_{i,((R_1-1)C_1+3)} & \cdots & c_{i,(R_1 C_1)} \end{bmatrix}$$

(4)Perform the inter-column permutation based on the pattern $\{P_1(j)\}$ (j=0, 1, ..., C-1) that is shown in Table 6.2.4-1, where $P_1(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{ik}.

$$\begin{bmatrix} y_{i1} & y_{i,(R_1+1)} & y_{i,(2R_1+1)} & \cdots y_{i,((C_1-1)R_1+1)} \\ y_{i2} & y_{i,(R_1+2)} & y_{i,(2R_1+2)} & \cdots y_{i,((C_1-1)R_1+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{iR_1} & y_{i,(2R_1)} & y_{i,(3R_1)} & \cdots & y_{i,(C_1 R_1)} \end{bmatrix}$$

(5)The output of the 1st interleaving is the bit sequence read out column by column from the inter-column permuted $R_1 \times C_1$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_{ik} that corresponds to bits c_{ik} with $k>E_i$ are removed from the output. The bits after 1st interleaving are denoted $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iE_i}$, where d_{i1} corresponds to the bit y_{ik} with smallest index k after pruning, d_{i2} corresponds to the bit y_{ik} with second smallest index k after pruning, and so on.

Table 6.2.45-1

TTI	Number of columns C_1	Inter-column permutation patterns
10 ms	1	{0}

20 ms	2	{0,1}
40 ms	4	{0,2,1,3}
80 ms	8	{0,4,2,6,1,5,3,7}

6.2.56.2.6 Radio frame segmentation

When the transmission time interval is longer than 10 ms, the input bit sequence is segmented and mapped onto consecutive radio frames. Following radio frame size equalisation the input bit sequence length is guaranteed to be an integer multiple of F_i.

The input bit sequence is denoted by $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iX_i}$ where i is the TrCH number and X_i is the number bits. The 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.
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 $N_i = Y_i$.

If the transmission time interval is longer than 10 ms, the bits in the TTI are segmented into several radio frames. The radio frame segmentation is done so that the number of bits in each radio frame is the same. If the number of bits in the TTI is not a multiple of the number of radio frames in the TTI, then filler bits are added to the radio frames which contain one bit less than the first radio frame.
< Note: It is assumed that filler bits are set to 0 >
The number of radio frames in the transmission time interval of TrCH i is denoted by F_i and the number of bits in the TTI by E_i. The number of filler bits Z_i for TrCH i is calculated as:

$Z_i = (F_i - (E_i \bmod F_i)) \bmod F_i$ $(Z_i \in \{0, 1, 2, \ldots, F_i-1\})$

The radio frames are numbered $1 £ n_i £ F_i$. The bits input to physical channel segmentation are denoted by $d_{i1}, d_{i2}, d_{i3}, \ldots, d_{iE_i}$, and the output by $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{iN_i}$. The radio frame segmentation is defined by the following relations, where $N_i = (E_i + Z_i) / F_i$:

$n_i = 1$ $e_{ik} = d_{ik}$ $k = 1, 2, \ldots, N_i$

$n_i = 2$ $e_{ik} = d_{i(k+N_i)}$ $k = 1, 2, \ldots, N_i$

….

$n_i = F_i - Z_i$ $e_{ik} = d_{i(k+(F_i-Z_i-1)N_i)}$ $k = 1, 2, \ldots, N_i$

$n_i = F_i - Z_i + 1$

$$e_{ik} = d_{i(k+(F_i - Z_i)N_i)} \quad k = 1, 2, \ldots, N_i - 1$$
$$e_{iN_i} = 0$$

…

$n_i = F_i$

$$e_{ik} = d_{i(k+(F_i - 1)N_i)} \quad k = 1, 2, \ldots, N_i - 1$$
$$e_{iN_i} = 0$$

The bits from radio frame segmentation are output radio frame by radio frame in ascending order with respect to n_i.