TSG-RAN Working Group 1 meeting #6
Hanover, Germany
August 30 – September 3, 1999

TSGR1#7(99)d23

Agenda item: Ad Hoc 4

Source: Siemens

Title: Text proposal for new notation in 25.222

Document for: Decision

Introduction

This paper proposes modifications on multiplexing for TDD in order to provide a consistent description throughout the document. All proposed changes reflect the modifications introduced in [3] to guarantee that both specifications on multiplexing, [1] and [2], use the same notation.

Besides, the text proposal in [4], which has already been accepted by Ad Hoc 1, has been adapted to the new notation and incorporated in this text proposal. Furthermore, section 6.2.11 dealing with multicode transmission has been removed, since this topic is described in detail elsewhere [5]. Finally, the picture given in Annex 1 detailing part of the data flow has been removed since it is no longer needed.

References

[1] TSG RAN WG1, TS 25.222 “Multiplexing and channel coding (TDD)”, Source: Editor

[2] TSG RAN WG1, TS 25.212 “Multiplexing and channel coding (FDD)”, Source: Editor

[3] Tdoc (99)B29 “Transport block concatenation and code block segmentation”, Source: Ericsson

[4] Tdoc (99)C09 “Text proposal: Physical Channel Segmentation and 2nd Interleaving for TDD”, Source: Siemens

[5] TSG RAN WG1, TS 25.221 “Physical channels and mapping of transport channels onto physical channels (TDD)”, Source: Editor

5 Definitions, symbols and abbreviations

5.1 Definitions

For the purposes of the present document, the following definitions apply:

<defined term>: <definition>.

5.2 Symbols

For the purposes of the present document, the following symbols apply:

éxù round towards ¥, i.e. integer such that x £ *éxù* < $x+1$
ëxû round towards -¥, i.e. integer such that $x-1$ < *ëxû* £ x
çxç absolute value of x

Unless otherwise is explicitly stated when the symbol is used, the meaning of the following symbols is:

i	TrCH number
j	TFC number
k	Bit number
l	TF number
m	Transport block number
n	Radio frame number
p	PhCH number
r	Code block number
I	Number of TrCHs in a CCTrCH.
C_i	Number of code blocks in one TTI of TrCH i.
F_i	Number of radio frames in one TTI of TrCH i.
M_i	Number of transport blocks in one TTI of TrCH i.
P	Number of PhCHs used for one CCTrCH.
PL	Puncturing Limit for the uplink. Signalled from higher layers
RM_i	Rate Matching attribute for TrCH i. Signalled from higher layers.

5.3 Abbreviations

ARQ	Automatic Repeat on Request
BCH	Broadcast Channel
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BS	Base Station
BSS	Base Station Subsystem
CA	Capacity Allocation
CAA	Capacity Allocation Acknowledgement
CBR	Constant Bit Rate
CCCH	Common Control Channel
CCTrCH	Coded Composite Transport Channel
CD	Capacity Deallocation
CDA	Capacity Deallocation Acknowledgement
CDMA	Code Division Multiple Access
CTDMA	Code Time Division Multiple Access

CRC	Cyclic Redundancy Check
DCA	Dynamic Channel Allocation
DCCH	Dedicated Control Channel
DL	Downlink
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Control
FER	Frame Error Rate
GF	Galois Field
HCS	Hierarchical Cell Structure
JD	Joint Detection
L1	Layer 1
L2	Layer 2
LLC	Logical Link Control
MA	Multiple Access
MAC	Medium Access Control
MAHO	Mobile Assisted Handover
MO	Mobile Originated
MOHO	Mobile Originated Handover
MS	Mobile Station
MT	Mobile Terminated
NRT	Non-Real Time
PC	Power Control
PCCC	Parallel Concatenated Convolutional Code
PCH	Paging Channel
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RACH	Random Access Channel
RF	Radio Frequency
RLC	Radio Link Control
RRC	Radio Resource Control
RRM	Radio Resource Management
RT	Real Time
RU	Resource Unit
SCCC	Serial Concatenated Convolutional Code
SCH	Synchronization Channel
SDCCH	Stand-alone Dedicated Control Channel
SFN	System Frame Number
SNR	Signal to Noise Ratio
SP	Switching Point
TCH	Traffic channel

TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TFCI	Transport Format Combination Indicator
TrBk	Transport Block
TrCH	Transport Channel
UL	Uplink
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate

6 Multiplexing, channel coding and interleaving

6.1 General

Data stream from/to MAC and higher layers (Transport block / Transport block set) is encoded/decoded to offer transport services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting (including rate matching), and interleaving and transport channels mapping onto/splitting from physical channels.

In the UTRA-TDD mode, the total number of basic physical channels (a certain time slot one spreading code on a certain carrier frequency) per frame is given by the maximum number of time slots which is 15 and the maximum number of CDMA codes per time slot.

~~*<Note: There has to be some guidance given on how different services are mapped to resource units (codes, time slots).>*~~

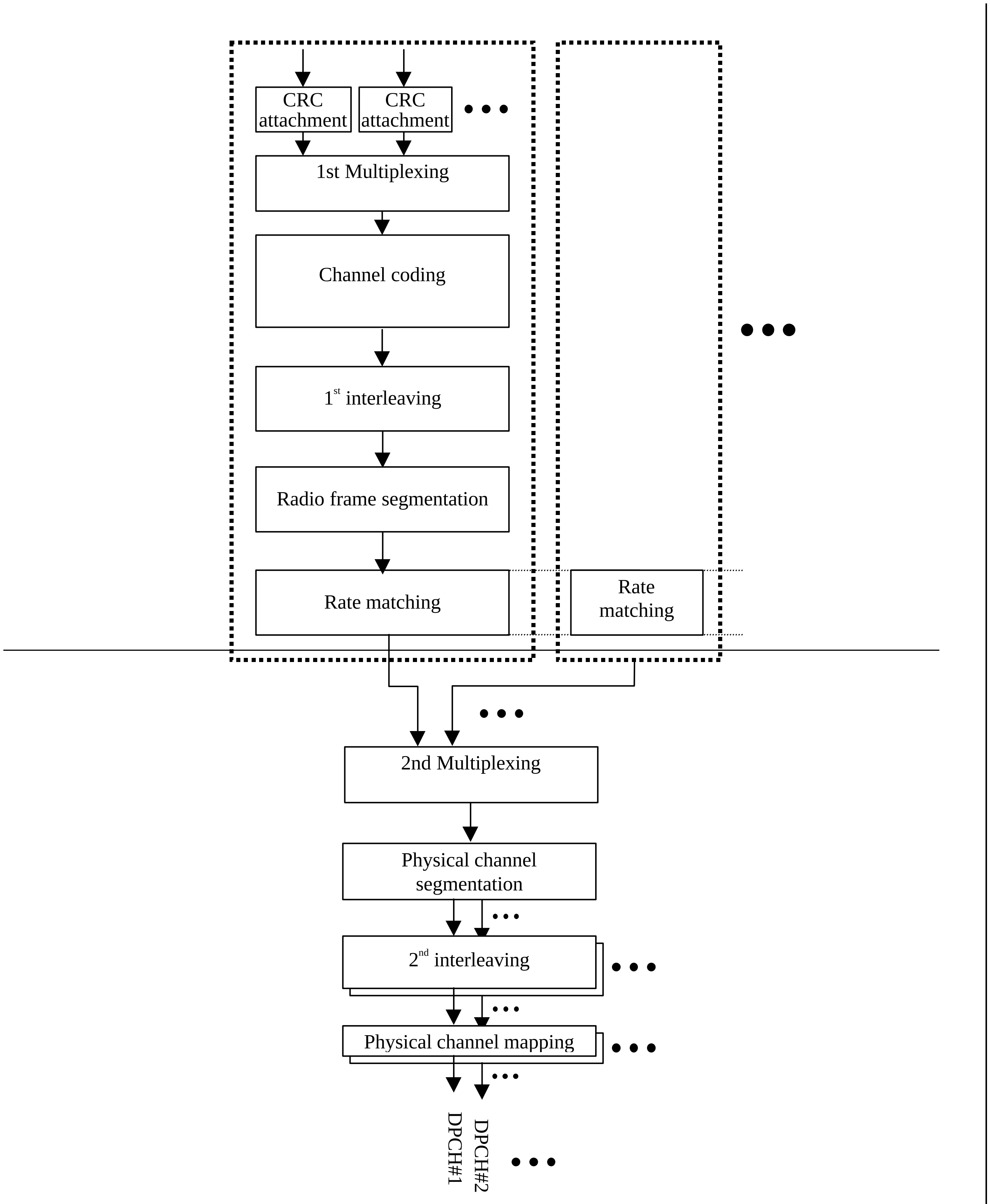
6.2 Transport channel coding/multiplexing

Figure 6-1 illustrates the overall concept of transport-channel coding and multiplexing. Data arrives to the coding/multiplexing unit in form of transport block sets, once every transmission time interval. The transmission time interval is transport-channel specific from the set {10 ms, 20 ms, 40 ms, 80 ms}.

The following coding/multiplexing steps can be identified:

- Add CRC to each transport block (see section 6.2.1)
- TrBk concatenation / Code block segmentation (see section 6.2.2)
- Channel coding (see section 6.2.3)
- Rate matching (see section 6.2.5)
- Interleaving (two steps, see sections 6.2.4 and 6.2.8)
- Radio frame segmentation
- Multiplexing of transport channels (~~two steps,~~ see section~~s 6.2.2 and~~ 6.2.6)
- Physical channel segmentation (see section 6.2.7)
- Mapping to physical channels (see section 6.2.9)

The coding/multiplexing steps for uplink and downlink are shown in Figure 6-1.



CRC attachment
CRC attachment
1st Multiplexing
Channel coding
1st interleaving
Radio frame segmentation
Rate matching
Rate matching
2nd Multiplexing
Physical channel segmentation
2nd interleaving
Physical channel mapping
DPCH#1
DPCH#2

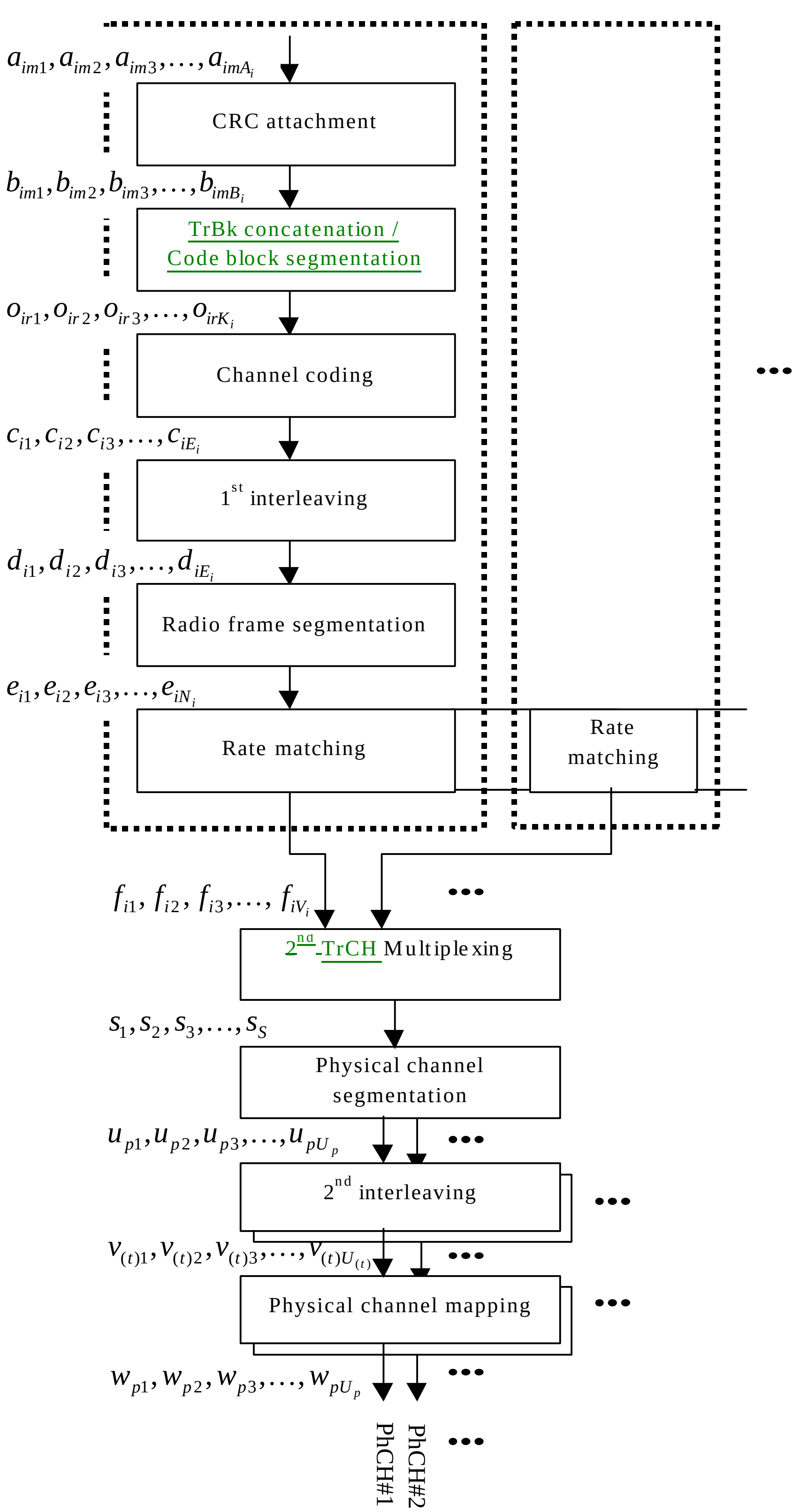


Figure 6–1. Transport channel multiplexing structure for uplink and downlink

Primarily, transport channels are multiplexed as described above, i.e. into one data stream mapped on one or several physical channels. However, an alternative way of multiplexing services is to use multiple CCTrCHs (Coded Composite Transport Channels), which corresponds to having several parallel multiplexing chains as in Figure 6-1, resulting in several data streams, each mapped to one or several physical channels.

6.2.1 Error detection

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

6.2.1.1 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:

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

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

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 ofa transport block of TrCH *i*, *m* is the transport block number, and L_i is 16, 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

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

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

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

yields a remainder equal to 0 when divided by $g_{CRC8}(D)$.
~~Denote the bits in a transport block delivered to layer 1 by $b_1, b_2, b_3, \ldots b_N$, and the parity bits by $p_1, p_2, \ldots p_L$. N is the length of the transport block and L is 16, 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~~

~~$b_1D^{N+15} + b_2D^{N+14} + \ldots + b_ND^{16} + p_1D^{15} + p_2D^{14} + \ldots + p_{15}D^1 + p_{16}$~~

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

~~$b_1D^{N+7} + b_2D^{N+6} + \ldots + b_ND^8 + p_1D^7 + p_2D^6 + \ldots + p_7D^1 + p_8$~~

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

6.2.1.2 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:

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

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

~~Bits delivered to layer 1 are denoted $b_1, b_2, b_3, \ldots b_N$, where N is the length of the transport block. The bits after CRC attachment are denoted by $w_1, w_2, w_3, \ldots w_{N+L}$, where L is 16, 8, or 0. The relation between b and w is:~~

~~$w_k = b_k$ $k = 1, 2, 3, \ldots N$~~

~~$w_k = p_{(L+1-(k-N))}$ $k = N+1, N+2, N+3, \ldots N+L$~~

6.2.2 ~~1st multiplexing~~ 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*, then code block segmentation is performed after the concatenation of the transport blocks. The maximum size of the code blocks depend on if convolutional or turbo coding is used for the TrCH.

~~1.1.1.1~~6.2.2.1 Concatenation of transport blocks

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

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

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

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

$\ldots$

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

~~1.1.1.2~~6.2.2.2 Code block segmentation

< Note: It is assumed that filler bits are set to 0 >

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: $Z = 512 - K_{tail}$
turbo coding: $Z = 5120 - K_{tail}$

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: $Y_i = C_iK_i - X_i$
If $X_i \le Z$, then $o_{i1k} = x_{ik}$, and $K_i = X_i$.
If $X_i \ge Z$, then

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

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

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

$\ldots$

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

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

~~Fix rate transport channels that are characterised by the same transport format attributes (as defined in 25.302) can be multiplexed before coding. When this multiplexing step is present, the transport blocks from different transport channels are serially concatenated. Denote the number of transport channels (TrCHs) by R, the number of transport blocks on each TrCH by P, and the number of bits in each transport block, including CRC bits, by K. The bits before multiplexing can then be described as follows:~~

~~Bits from transport block 1 of transport channel 1: w_{111}, w_{112}, w_{113}, … w_{11K}~~

~~Bits from transport block 2 of transport channel 1: w_{121}, w_{122}, w_{123}, … w_{12K}~~

~~…~~

~~Bits from transport block P of transport channel 1: $w_{1P1}, w_{1P2}, w_{1P3}, \dots w_{1PK}$~~

~~Bits from transport block 1 of transport channel 2: $w_{211}, w_{212}, w_{213}, \dots w_{21K}$~~

~~…~~

~~Bits from transport block P of transport channel 2: $w_{2P1}, w_{2P2}, w_{2P3}, \dots w_{2PK}$~~

~~…~~

~~Bits from transport block 1 of transport channel R: $w_{R11}, w_{R12}, w_{R13}, \dots w_{R1K}$~~

~~…~~

~~Bits from transport block P of transport channel R: $w_{RP1}, w_{RP2}, w_{RP3}, \dots w_{RPK}$~~

~~The bits after first multiplexing are denoted by $d_1, d_2, d_3, \dots d_M$, and defined by the following relations:~~

~~$d_k = w_{11k}$ k = 1, 2, …, K~~

~~$d_k = w_{12(k-K)}$ k = K + 1, K + 2, …, 2K~~

~~…~~

~~$d_k = w_{1P(k-(P-1)K)}$ k = (P - 1)K + 1, …, PK~~

TrCH 1

~~$d_k = w_{21(k-PK)}$ k = PK + 1, …, (P + 1)K~~

~~…~~

~~$d_k = w_{2P(k-(2P-1)K)}$ k = (2P - 1)K + 1, …, 2PK~~

TrCH 2

~~…~~

~~$d_k = w_{R1(k-(R-1)PK)}$ k = (R - 1)PK + 1, …, ((R - 1)P + 1)K~~

—

~~$d_k = w_{RP(k-(RP-1)K)}$ k = (RP - 1)K + 1, …, RPK~~

TrCH R

~~*<Note: Above it is assumed that all transport blocks have the same size. There are cases when the total number of bits that are sent during a transmission time interval is not a multiple of the number of transport blocks. A few padding bits are then needed but the exact insertion point (in the multiplexing chain) of these bits is for further study.>*~~

6.2.3 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 $x_{ir1}, x_{ir2}, x_{ir3}, \dots, x_{irX_i}$. 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 $c_{i1}, c_{i2}, c_{i3}, \dots, c_{iE_i}$, where *i* is the TrCH number and $E_i = C_iX_i$. The output bits are defined by the following relations:

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

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

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

…

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

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

~~The maximum encoding segment length for turbo coding is 5120 bits. In Real Time (RT) services a FEC coding is used, instead Non Real Time (NRT) services could be well managed with a proper combination of FEC and ARQ.~~

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]~~ ~~*<Editor's note: the values in square brackets have not yet been approved.>*~~
DCH		1/2, 1/3 or no coding
DCH	Turbo code	

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.

6.2.3.1 Convolutional Coding

- Constraint length K=9. Coding rates 1/2~~,~~ and 1/3~~ and [2/3, 7/8]~~.
- The configuration of the convolutional coder is presented in Fig. 6-2.
- The output from the convolutional coder shall be done in the order ~~starting from~~ output0, output1,~~ and~~ output2, output0, output1, ... , output2. (When coding rate is 1/2, output is done up to output 1).
- The initial value of the shift register of the coder shall be "all 0".
- K-1 tail bits (value 0) shall be added to the end of the code~~ing~~ block before coding.

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

6.2.4 1st interleaving

~~The 1st interleaving of channel interleaving consists of two stage operations. In first stage, the input sequence is written into rectangular matrix row by row. The second stage is inter-column permutation. The two-stage operations are described as follows, the input block length is assumed to be K_1.~~

~~**First Stage:**~~

~~(1) Select a column number C_1 from Table 6.2.4-1.~~

~~(2) Determine a row number R_1 by finding minimum integer R_1 such that,~~

~~$K_1 <= R_1 \times C_1$.~~

~~(3) The input sequence of the 1st interleaving is written into the $R_1 \times C_1$ rectangular matrix row by row.~~

~~**Second Stage:**~~

~~(1) Perform the inter-column permutation based on the pattern $\{P_1(j)\}$ (j=0,1, ..., C-1) that is shown in Table 4-xx, where $P_1(j)$ is the original column position of the j-th permuted column.~~

~~(2) The output of the 1st interleaving is the sequence read out column by column from the inter-column permuted $R_1 \times C_1$ matrix and the output is pruned by deleting the non-existence bits in the input sequence, where the deleting bits number l_1 is defined as:~~

~~$l_1 = R_1 \times C_1 - K_1$.~~

~~Table 6.2.4-1~~

~~Interleaving span~~	~~Column number 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}~~

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,

$$E_i \le R_1 \times C_1.$$

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

$$\begin{bmatrix} c_{i1} & c_{i2} & c_{i3} & \cdots & c_{iC_1} \\ c_{i,(C_1+1)} & c_{i,(C_1+2)} & c_{i,(C_1+3)} & \cdots & c_{i,(2C_1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c_{i,((R_1-1)C_1+1)} & c_{i,((R_1-1)C_1+2)} & c_{i,((R_1-1)C_1+3)} & \cdots & c_{i,(R_1C_1)} \end{bmatrix}$$

(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}.

$$-\begin{bmatrix} y_{i1} & y_{i,(R_1+1)} & y_{i,(2R_1+1)} & \cdots & y_{i,((C_1-1)R_1+1)} \\ y_{i2} & y_{i,(R_1+2)} & y_{i,(2R_1+2)} & \cdots & y_{i,((C_1-1)R_1+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{iR_1} & y_{i,(2R_1)} & y_{i,(3R_1)} & \cdots & y_{i,(C_1R_1)} \end{bmatrix}$$

(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 6.2.4-1

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}

6.2.5 Radio frame segmentation

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 $N_i = (E_i + Z_i) / F_i$:

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

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

…

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

$$n_i = F_i - Z_i + 1 \begin{cases} e_{ik} = d_{i(k+(F_i-Z_i)N_i)} & k = 1, 2, \ldots, N_i - 1 \\ e_{iN_i} = 0 & \end{cases}$$

…

$$n_i = F_i \begin{cases} e_{ik} = d_{i(k+(F_i-1)N_i)} & k = 1, 2, \ldots, N_i - 1 \\ e_{iN_i} = 0 & \end{cases}$$

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

~~Each transport channel with transmission time interval 10, 20, 40, or 80 msec is segmented into 10 msec equi-sized data blocks. Those segmented 1, 2, 4, or 8 blocks, depending on transmission time interval, are output to rate matching in block-wise order at every 10 msec.~~

~~Figure A-1 illustrates data flow from 1^{st} interleaver down to 2^{nd} interleaver in channel coding and multiplexing chain. In the figures, it is assumed that there are N different channel coding and multiplexing chains. The following subsections describe input-output relationship of radio frame segmentation in bit-wise manner, referring to the notation in Figure A-1, where the notations in each data block, for examples L_1, R_1, K_1, P/M, etc., indicate number of bits of the data block.~~

~~Define some notations:L_i = Size of i^{th} transport channel data in bits to radio frame segmentation~~

~~T_i = Transmission Time Interval of i^{th} channel coding and multiplexing chain (msec) / 10 (msec)~~

~~So, T_i Î $\{1, 2, 4, 8\}$ for $i = 0, 1, 2, \ldots, N$~~

~~6.2.5.1 Radio frame size equalization~~

~~i^{th} transport channel data of size L_i is segmented into radio frames of size L_i/T_i. Since the size of radio frame, L_i/T_i is not necessarily an integer, some of T_i the radio frames will contain one bit less than others. For systematic process of the proceeding functional blocks, the radio frame sizes are equalized to be one finite size by considering the number of proper filler bits. Note that maximum possible filler bits are 7 for transmission time interval of 80 msec. These filler bits are evenly distributed over the one-bit short radio frames. Following is the algorithm of radio frame size equalization.~~

~~*t = radio frame index (1, 2, 3, …, T_i) for a given i^{th} channel coding and multiplexing chain*~~

~~$r_i = T_i - (L_i \bmod T_i)$ Î $\{0, 1, 2, \ldots, T_i-1\}$ *// number of filler bits*~~

~~$(L_i + r_i)/ T_i = R_i$ *// Target radio frame size for uplink*~~

~~*If* r_i ¹ 0 *then*~~

~~*For each t (*³ $T_i - r_i + 1$*)*~~

~~*Add one filler bit to the end of t^{th} radio frame*~~

~~*End*~~
~~*End If*~~

~~6.2.5.2 Radio frame segmentation rule~~

~~Parameter r_i for segmentation are determined in radio frame size equalization.~~

~~The bits before radio frame segmentation for i^{th} channel coding and multiplexing chain are denoted by:~~

~~$b_{i1}, b_{i2}, \ldots b_{iL_i}$~~

~~Bits after radio frame segmentation block are 10 msec-based and denoted by:~~

~~$c_{i1}, \ldots c_{i,(L_i+r_i)/T_i}$~~

~~and related to the input bits to radio frame segmentation as follows.~~

~~Bits after radio frame segmentation in the first 10 msec time interval: $(t=1)$~~

~~$c_{ij} = b_{ij}$ $j=1,2,\ldots, (L_i+r_i)/T_i$~~

~~$((L_i+ r_i)/T_i$ equals to R_i and K_i for uplink and downlink, respectively.)~~

~~Bits after radio frame segmentation in the second 10 msec time interval: $(t=2)$~~

~~$c_{ij} = b_{i,(j+(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i$~~

~~…~~

~~Bits after radio frame segmentation in the $(T_i^- - r_i)^{th}$ 10 msec time interval: $(t=T_i^- - r_i)$~~

~~$c_{ij} = b_{i,(j+(T_i-r_i-1)\cdot(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i$~~

~~Bits after radio frame segmentation in the $(T_i^- - r_i+1)^{th}$ 10 msec time interval: $(t=T_i^- - r_i+1)$~~

~~$c_{ij} = b_{i,(j+(T_i-r_i)(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i-1$~~

~~$c_{ij} = filler\_bit(0/1)$ $j= (L_i+r_i)/T_i$ (filler bit)~~

~~…~~

~~Bits after radio frame segmentation in the T_i^{th} 10 msec time interval: $(t=T_i)$~~

~~$c_{ij} = b_{i,(j+(T_i-1)(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i-1$~~

~~$c_{ij} = filler\_bit(0/1)$ $j=(L_i+r_i)/T_i$ (filler bit)~~

6.2.6 Rate matching

Rate matching means that bits on a ~~transport channel~~TrCH are repeated or punctured. Higher layers assign a rate-matching attribute for each ~~transport channel~~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 ~~transport channel~~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 multiplexing is identical to the total channel bit rate of the allocated dedicated physical channels.

Notation used in Section 6.2.~~5~~6 and subsections:

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

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

RM_i: Semi-static rate matching attribute for ~~transport channel~~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.

~~T~~I: Number of ~~transport channel~~TrCHs in a CCTrCH.

Z_{mj}: Intermediate calculation variable.

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

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

q: Average puncturing distance.

$I_F($~~k~~$n_i)$: 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).

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

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

6.2.6.1 Determination of rate matching parameters

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

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

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

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

Puncturing can be used to minimise the number of required transmission capacity. The maximum amount of puncturing that can be applied is signalled ~~at connection setup~~ from higher layers and denoted by *PL*. The possible values for N_{data} ~~are always multiples of~~in uplink and downlink depend on the dedicated physical channels which are assigned to the link, respectively.~~ with the smallest capacity, reduced by the amount of bits which carry the TFCI.~~ 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 ~~transport channel~~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 ~~4~~6.2.6.~~3~~2, where $DN = DN_{ij}$ and $N = N_{ij}$.

Additionally, the following parameters are needed:

q= ëN_{ij}*/(ô*DN_{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 *l* = 0 to F_i-1
    S(I_F *(él*q'ù* mod F_i*)) = (él*q'ù* div F_i*)*~~*-- where é ù means round upwards.*~~

end for

6.2.6.2 Rate matching algorithm

~~Denote the bits before rate matching by:~~

The bits input to the rate matching are denoted by $e_{i1}, e_{i2}, e_{13}, \ldots, e_{iN_i}$, where *i* is the TrCH with $N = N_{ij} = N_i$.

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 $V_i = N + DN = N_{ij} + DN_{ij}$.

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

y = -DN

```
        e = (2*S($n_i$) * y +  N) mod 2N  -- initial error between current and desired puncturing ratio
        m = 1           -- index of current bit
        do while m <= N
            e = e – 2 * y               -- update error
            if e <= 0 then              -- check if bit number m should be punctured
                puncture bit $x_m$
                e = e + 2*N             -- update error
            end if
            m = m + 1                   --  next bit
        end do
    else
        y = DN

        e = (2*S($n_i$) * y +  N) mod 2N  -- initial error between current and desired puncturing ratio
        m = 1               -- index of current bit
        do while m <= N
            e = e – 2 * y               -- update error
            do while e <= 0             -- check if bit number m should be repeated
                repeat bit $x_m$
                e = e +  2*N -- update error
            enddo
            m = m + 1                   --  next bit
        end do
    end if
```

A repeated bit is placed directly after the original one.

~~c_1, c_2, c_3, … c_N~~

~~The rate matching rule is as follows:~~

```
~~if puncturing is to be performed~~

~~y = -ΔN~~
~~e = (2*S(k) * y +  N) mod 2N    -- initial error between current and desired puncturing ratio~~
~~m = 1           -- index of current bit~~
~~do while m <= N~~
~~e = e – 2 * y          -- update error~~
~~if e <= 0 then         -- check if bit number m should be punctured~~
~~puncture bit $c_m$~~
~~e = e + 2*N     -- update error~~
~~end if~~
~~m = m + 1                --  next bit~~
~~end do~~
~~else~~
~~y = ΔN~~

~~e = (2*S(k) * y +  N) mod 2N    -- initial error between current and desired puncturing ratio~~
~~m = 1               -- index of current bit~~
~~do while m <= N~~
~~e = e – 2 * y          -- update error~~
~~do while e <= 0        -- check if bit number m should be repeated~~
~~repeat bit $c_m$~~
~~e = e +  2*N    -- update error~~
~~enddo~~
~~m = m + 1                --  next bit~~
~~end do~~
~~end if~~
```

~~A repeated bit is placed directly after the original one.~~

6.2.7 ~~Second~~TrCH multiplexing

Every 10 ms, one radio frame from each TrCH is delivered to the TrCH multiplexing. These radio frames are serially

multiplexed into a coded composite transport channel (CCTrCH).
The bits input to the TrCH multiplexing are 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 in the radio frame of TrCH i. The number of TrCHs is denoted by I. The bits output from TrCH multiplexing are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits, i.e. $S = \sum_i V_i$. The TrCH multiplexing is defined by the following relations:

$s_k = f_{1k}$ $k = 1, 2, \ldots, V_1$

$s_k = f_{2,(k-V_1)}$ $k = V_1+1, V_1+2, \ldots, V_1+V_2$

$s_k = f_{3,(k-(V_1+V_2))}$ $k = (V_1+V_2)+1, (V_1+V_2)+2, \ldots, (V_1+V_2)+V_3$

$\ldots$

$s_k = f_{I,(k-(V_1+V_2+\ldots+V_{I-1}))}$ $k = (V_1+V_2+\ldots+V_{I-1})+1, (V_1+V_2+\ldots+V_{I-1})+2, \ldots, (V_1+V_2+\ldots+V_{I-1})+V_I$

~~For both uplink and downlink, radio frames in each channel coding and multiplexing chains are serially multiplexed into a 10 msec coded composite transport channel.~~

~~Figure A-1 illustrates data flow from 1st interleaver down to 2nd interleaver in channel coding and multiplexing chain. In the figure, it is assumed that there are N different channel coding and multiplexing chains. Following subsection describes the input-output relationship of 2nd multiplexing in bit-wise manner, referring to the notations in Figure A-1, where the notation in each data block, for examples L_1, R_1, K_1, P/M, etc., indicate number of bits of the data block.~~

~~The bits before second multiplexing in uplink are described as follows:~~

~~Bits from rate matching 1: $c_{11}, c_{12}, \ldots c_{1K_1}$~~

~~Bits from rate matching 2: $c_{21}, c_{22}, \ldots c_{2K_2}$~~

~~Bits from rate matching 3: $c_{31}, c_{32}, \ldots c_{3K_3}$~~

~~…~~

~~Bits from rate matching N: $c_{N1}, c_{N2}, \ldots c_{NK_N}$~~

~~The bits after second multiplexing are denoted by $d_1, d_2, \ldots, d_P$~~

~~and defined by the following relationships:~~

~~For $j=1,2,3 \ldots,P$ where $P=K_1+K_2+ \ldots +K_N$~~

~~$d_j = c_{1j}$ $j=1,2,\ldots K_1$~~

~~$d_j = c_{2,(j-K_1)}$ $j= K_1+1, K_1+2, \ldots, K_1+K_2$~~

~~$d_j = c_{3,(j-(K_1+K_2))}$ $j=(K_1+K_2)+1, (K_1+K_2)+2, \ldots, (K_1+K_2)+ K_3$~~

~~$\ldots$~~

~~$d_j = c_{N,(j-(K_1+K_2+\ldots+K_{N-1}))}$ $j=(K_1+K_2+\ldots+ K_{N-1})+1, (K_1+K_2+\ldots+ K_{N-1})+2, \ldots, (K_1+K_2+\ldots+ K_{N-1})+K_N$~~

6.2.8 Physical channel segmentation

When more than one PhCH is used, physical channel segmentation divides the bits among the different PhCHs. The bits input to the physical channel segmentation are denoted by $s_1, s_2, s_3, \ldots, s_S$, where S is the number of bits input to the physical channel segmentation block. The number of PhCHs is denoted by P.
The bits after physical channel segmentation are denoted $u_{p1}, u_{p2}, u_{p3}, \ldots, u_{pU_p}$, where p is PhCH number and U_p is the in general variable number of bits in the respective radio frame for each PhCH. The relation between s_k and u_{pk} is given below.

Bits on first PhCH after physical channel segmentation:

$u_{1k} = s_k$ $k = 1, 2, \ldots, U_1$

Bits on second PhCH after physical channel segmentation:

$u_{2k} = s_{(k+U_1)}$ $k = 1, 2, \ldots, U_2$

…

Bits on the P^{th} PhCH after physical channel segmentation:

$u_{Pk} = s_{(k+U_1+\ldots+U_{P-1})}$ $k = 1, 2, \ldots, U_P$

~~*<Editor's note: Physical channel segmentation will depend upon the QoS parameters of the different transport channels in each CCTrCH. It will be necessary to specify what are the parameters expected from L2 for it.>*~~

~~*<Editor's note: for physical channel segmentation, it is assumed that the segmented physical channels use the same SF>*~~

~~Data after multiplexing of transport channels with different QoS can get segmented into multiple physical channels which are transmitted in parallel during a 10ms interval.~~

~~Figure A-1 illustrates data flow from 1st interleaver down to 2nd interleaver in channel coding and multiplexing chain. In the figures, it is assumed that there are *N* different channel coding and multiplexing chains, and *M* physical channels. The following subsection describes input-output relationship of physical channel segmentation in bit-wise manner, referring to the notation in Figure A-1, where the notation in each data block, for examples L_1, R_1, K_1, P/M, etc., indicate number of bits of the data block.~~

~~The bits before physical channel segmentation are described as follows:~~

~~Bits from second multiplexing: $d_1, d_2, \ldots, d_P$~~

~~*M* is the number of physical channel~~

~~The bits after physical channel segmentation are defined by the following relationship:~~

~~The first physical channel bits after physical channel segmentation:~~

~~$e_{1j} = d_j$ $j=1,2,\ldots,P/M$~~

~~The second physical channel bits after physical channel segmentation:~~

~~$e_{2j} = d_{(j+P/M)}$ $j=1,2,\ldots,P/M$~~

~~…~~

~~The M^{th} physical channel bits after physical channel segmentation:~~

~~$e_{Mj} = d_{(j+(M-1)P/M)}$ $j=1,2,\ldots,P/M$~~

6.2.9 2nd interleaving

The 2nd interleaving is a block interleaver with inter-column permutations. It can be applied jointly to all data bits transmitted during one frame (frame related), or separately within each timeslot, on which the CCTrCH is mapped (timeslot related). The selection of the 2nd interleaving scheme is controlled by higher layer.

6.2.9.1 Frame related 2nd interleaving

In case of frame related interleaving, the bits input to the 2nd interleaver are denoted $x_1, x_2, x_3, \ldots, x_U$, where U is the total number of bits after TrCH multiplexing transmitted during the respective radio frame.

The relation between x_k and the bits u_{pk} in the respective physical channels is given below:

$x_k = u_{1k}$ $k = 1, 2, \ldots, U_1$

$x_{(k+U_1)} = u_{2k}$ $k = 1, 2, \ldots, U_2$

…

$x_{(k+U_1+\ldots+U_{P-1})} = u_{Pk}$ $k = 1, 2, \ldots, U_P$

The following steps have to be performed once for each CCTrCH:

(1) Set the number of columns $C_2 = 30$. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that
U £ R_2C_2.

(3) The bits input to the 2nd interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_1 & x_2 & x_3 & \ldots & x_{30} \\ x_{31} & x_{32} & x_{33} & \ldots & x_{60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{(R_2-1)30+1} & x_{(R_2-1)30+2} & x_{(R_2-1)30+3} & \ldots & x_{R_2 \cdot 30} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern $\{P_2(j)\}$ $(j = 0, 1, ..., C_2\text{-}1)$ that is shown in Table 6.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_k.

$$\begin{bmatrix} y_1 & y_{R_2+1} & y_{2R_2+1} & \ldots y_{29R_2+1} \\ y_2 & y_{R_2+2} & y_{2R_2+2} & \ldots y_{29R_2+2} \\ \vdots & \vdots & \vdots & \ldots \quad \vdots \\ y_{R_2} & y_{2R_2} & y_{3R_2} & \ldots \; y_{30R_2} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_k that corresponds to bits x_k with $k>U$ are removed from the output. The bits after 2nd interleaving are denoted by $v_1, v_2, \ldots, v_U$, where v_1 corresponds to the bit y_k with smallest index k after pruning, v_2 to the bit y_k with second smallest index k after pruning, and so on.

6.2.9.2 Timeslot related 2nd interleaving

In case of timeslot related 2nd interleaving, the bits input to the 2nd interleaver are denoted $x_{t1}, x_{t2}, x_{t3}, \ldots, x_{tU_t}$, where t refers to a certain timeslot, and U_t is the number of bits transmitted in this timeslot during the respective radio frame.

In each timeslot t the relation between x_{tk} and u_{pk} is given below with P_t refering to the number of physical channels within the respective timeslot:

$x_{tk} = u_{1k}$ $k = 1, 2 , \ldots, U_1$

$x_{t(k+U_1)} = u_{2k}$ $k = 1, 2 , \ldots, U_2$

…

$x_{t(k+U_1+\ldots+U_{P_t-1})} = u_{P_t k}$ $k = 1, 2 , \ldots, U_{P_t}$

The following steps have to be performed for each timeslot t, on which the respective CCTrCH is mapped:

(1) Set the number of columns $C_2 = 30$. The columns are numbered 0, 1, 2, …, C_2-1 from left to right.

(2) Determine the number of rows R_2 by finding minimum integer R_2 such that
U_t £ R_2C_2.

(3) The bits input to the 2nd interleaving are written into the $R_2 \times C_2$ rectangular matrix row by row.

$$\begin{bmatrix} x_{t1} & x_{t2} & x_{t3} & \ldots & x_{t30} \\ x_{t31} & x_{t32} & x_{t33} & \ldots & x_{t60} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_{t,((R_2-1)30+1)} & x_{t,((R_2-1)30+2)} & x_{t,((R_2-1)30+3)} & \ldots & x_{t,(R_2 30)} \end{bmatrix}$$

(4) Perform the inter-column permutation based on the pattern {$P_2(j)$} (j = 0, 1, ..., C_2-1) that is shown in Table 6.2.9-1, where $P_2(j)$ is the original column position of the j-th permuted column. After permutation of the columns, the bits are denoted by y_{tk}.

$$\begin{bmatrix} y_{t1} & y_{t,(R_2+1)} & y_{t,(2R_2+1)} & \cdots & y_{t,(29R_2+1)} \\ y_{t2} & y_{t,(R_2+2)} & y_{t,(2R_2+2)} & \cdots & y_{t,(29R_2+2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_{tR_2} & y_{t,(2R_2)} & y_{t,(3R_2)} & \cdots & y_{t,(30R_2)} \end{bmatrix}$$

(5) The output of the 2nd interleaving is the bit sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix. The output is pruned by deleting bits that were not present in the input bit sequence, i.e. bits y_{tk} that corresponds to bits x_{tk} with $k>U_t$ are removed from the output. The bits after 2nd interleaving are denoted by $v_{t1}, v_{t2}, \ldots, v_{tU_t}$, where v_{t1} corresponds to the bit y_{tk} with smallest index k after pruning, v_{t2} to the bit y_{tk} with second smallest index k after pruning, and so on.

~~The 2nd interleaving of channel interleaving consists of two stage operations. In first stage, the input sequence is written into rectangular matrix row by row. The second stage is inter-column permutation. The two-stage operations are described as follows, the input block length is assumed to be K_2.~~

~~First Stage:~~

~~(1) Set a column number C_2 = 30.~~

~~(2) Determine a row number R_2 by finding minimum integer R_2 such that,~~

~~$K_2 <= R_2 \times C_2$.~~

~~(3) The input sequence of the 2nd interleaving is written into the $R_2 \times C_2$ rectangular matrix row by row.~~

~~Second Stage:~~

~~(1) Perform the inter-column permutation based on the pattern {$P_2(j)$} (j=0,1, ..., C-1) that is shown in Table 4-yy, where $P_2(j)$ is the original column position of the j-th permuted column.~~

~~(2) The output of the 2nd interleaving is the sequence read out column by column from the inter-column permuted $R_2 \times C_2$ matrix and the output is pruned by deleting the non-existence bits in the input sequence, where the deleting bits number l_2 is defined as:~~
~~$l_2 = R_2 \times C_2 - K_2$.~~
~~*<Note: Inter-column permutation pattern in table 6.2.8-1 is a working assumption of WG1.>*~~

Table 6.2.~~8~~9-1

Column number C_2	Inter-column permutation pattern
30	{0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17}

6.2.10 Physical channel mapping

The PhCH for both uplink and downlink is defined in [6]. The bits after physical channel mapping are denoted by $w_{p1}, w_{p2}, \ldots, w_{pU_p}$, where p is the PhCH number and U_p is the number of bits in one radio frame for the respective PhCH. The bits w_{pk} are mapped to the PhCHs so that the bits for each PhCH are transmitted over the air in ascending order with respect to k. The mapping scheme depends on the applied 2nd interleaving scheme.

6.2.10.1 Mapping scheme after frame related 2nd interleaving

The following mapping rule is applied:

Bits on first PhCH after physical channel mapping:

$w_{1k} = v_k$ $k = 1, 2, \ldots, U_1$

Bits on second PhCH after physical channel mapping:

$w_{2k} = v_{(k+U_1)}$ $k = 1, 2, \ldots, U_2$

…

Bits on the P^{th} PhCH after physical channel mapping:

$w_{Pk} = v_{(k+U_1+\ldots+U_{P-1})}$ $k = 1, 2, \ldots, U_P$

6.2.10.2 Mapping scheme after timeslot related 2nd interleaving

For each timeslot only those physical channels with $p = 1, 2, \ldots, P_t$ are considered respectively, which are transmitted in that timeslot, and the following mapping scheme is applied:

Bits on first PhCH in timeslot *t* after physical channel mapping:

$w_{1k} = v_{tk}$ $k = 1, 2, \ldots, U_1$

Bits on second PhCH in timeslot *t* after physical channel mapping:

$w_{2k} = v_{t(k+U_1)}$ $k = 1, 2, \ldots, U_2$

…

Bits on the PhCH P_t in timeslot *t* after physical channel mapping:

$w_{P_t k} = v_{(k+U_1+\ldots+U_{P_t-1})}$ $k = 1, 2, \ldots, U_{P_t}$

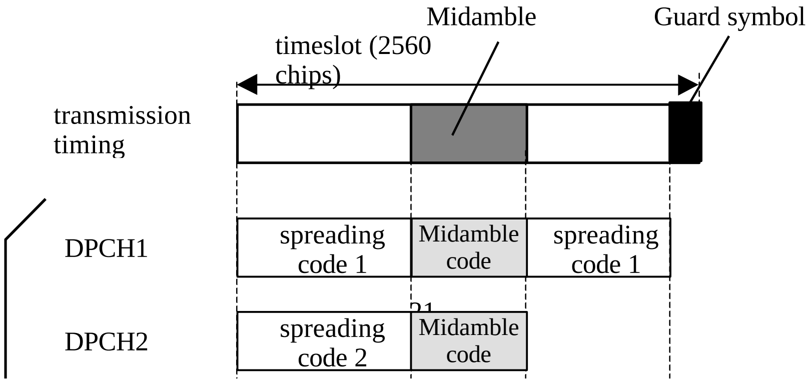
- ~~If transport data is less than the number of DPCH bits in a radio frame, dynamic rate matching is applied to ensure that all resource units which belong to one user are either completely filled with data or empty.~~
- ~~The transmission of the DPCH symbols shall be ON, only if there is data to transmit. If there is no data, the transmission shall be OFF.~~
- ~~For transport channels not relying on TFCI for transport format detection (blind transport format detection), the positions of the transport channels within the frame should be fixed.~~
- ~~For transport channels relying on TFCI for transport format detection, the positions of the transport channels should be non-fixed.~~

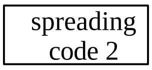
~~6.2.11Multicode transmission~~

- ~~For multi-code transmission several codes within one or several timeslots can be allocated for each user independently for uplink and downlink. Those resource units are negotiated by means of higher layers at the beginning of a transmission dependent on the individual services.~~

- ~~For one user within a certain timeslot there is only one midamble code.~~

- ~~Dependent on the actual amount of data to be transmitted not all resource units which are allocated to a certain link are used continuously. If no data is transmitted within a time slot during a frame, then also the midamble will be omitted.~~





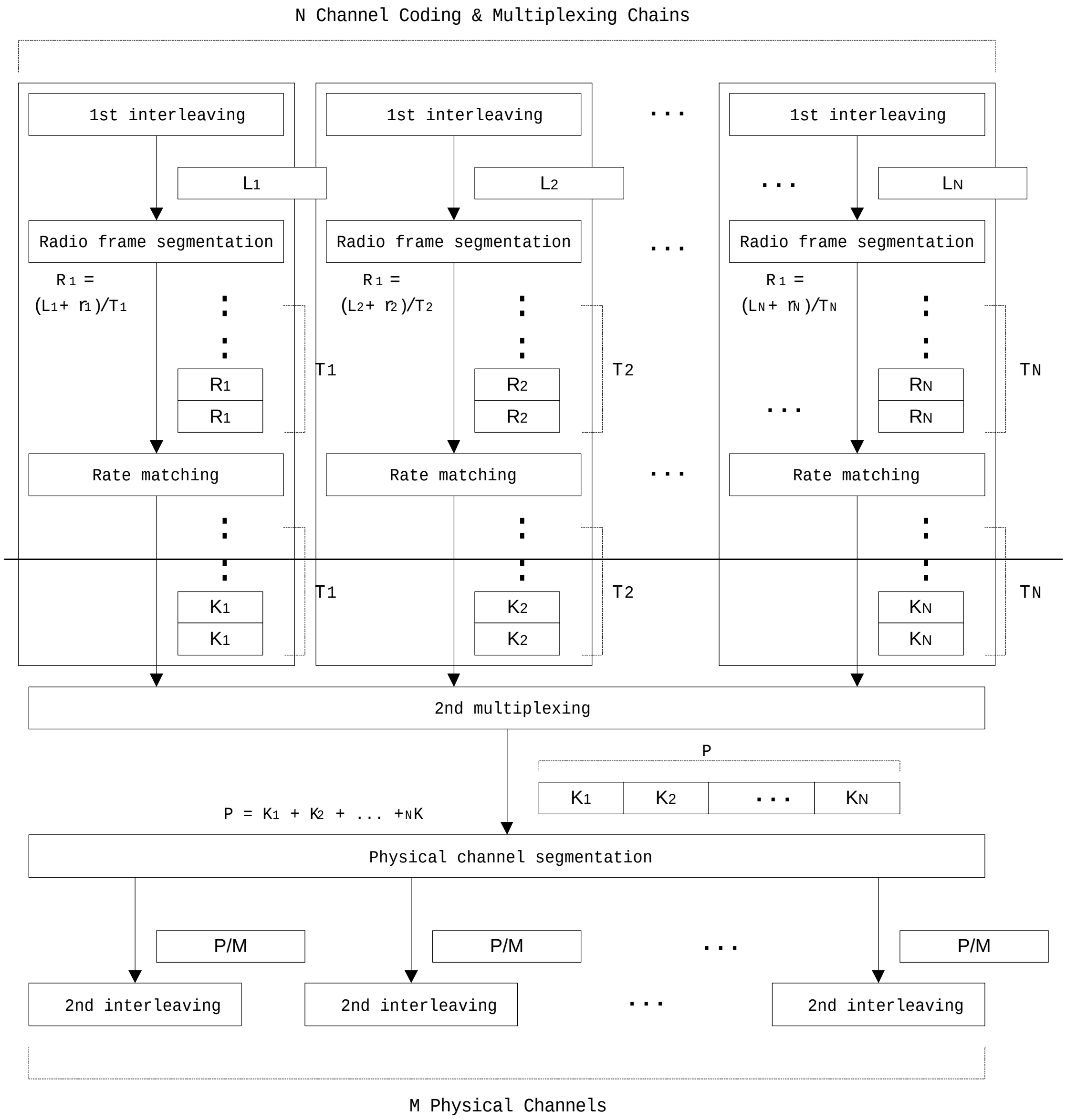
~~**Fig. 6-7 Spreading code and midamble in multi-code transmission**~~

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

---

7 Annex

~~A.1 Data Flow from Radio Frame Segmentation to Physical Channel Segmentation~~



~~*Figure A-1 Part of channel coding and multiplexing chains*~~

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