TSGR1#7 (99)c70

TS 25.212 V2.1.0 (1999-08)

Technical Specification

3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG) Radio Access Network (RAN); Working Group 1 (WG1); Multiplexing and channel coding (FDD)

Reference

<Workitem> (<Shortfilename>.PDF)

Keywords

<keyword[, keyword]>

3GPP

Postal address

Office address

Internet

secretariat@3gpp.org

Contents

Intellectual Property Rights

<IPR notice shall be provided once correct notice is available within 3GPP>

Foreword

This Technical Specification has been produced by the 3GPP.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version 3.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 Indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification.

1 Scope

This specification describes the documents being produced by the 3GPP TSG RAN WG1and first complete versions expected to be available by end of 1999. This specification describes the characteristics of the Layer 1 multiplexing and channel coding in the FDD mode of UTRA.

The 25.2series specifies Um point for the 3G mobile system. This series defines the minimum level of specifications required for basic connections in terms of mutual connectivity and compatibility.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] 3GPP RAN TS 25.201: “Physical layer – General Description”

[2] 3GPP RAN TS 25.211: “Transport channels and physical channels (FDD)”

[3] 3GPP RAN TS 25.213: “Spreading and modulation (FDD)”

[4] 3GPP RAN TS 25.214: “Physical layer procedures (FDD)”

[5] 3GPP RAN TS 25.221: “Transport channels and physical channels (TDD)”

[6] 3GPP RAN TS 25.222: “Multiplexing and channel coding (TDD)”

[7] 3GPP RAN TS 25.223: “Spreading and modulation (TDD)”

[8] 3GPP RAN TS 25.224: “Physical layer procedures (TDD)”

[9] 3GPP RAN TS 25.231: “Measurements”

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

<defined term>: <definition>.

example: text used to clarify abstract rules by applying them literally.

3.2 Symbols

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

<symbol> <Explanation>

3.3 Abbreviations

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

<ACRONYM> <Explanation>

ACS	Add, Compare, Select
ARQ	Automatic Repeat Request
BCH	Broadcast Channel
BER	Bit Error Rate
BLER	Block Error Rate
BS	Base Station
CCPCH	Common Control Physical Channel
CCTrCH	Coded Composite Transport Channel
CRC	Cyclic Redundancy Code
DCH	Dedicated Channel
DL	Downlink (Forward link)
DPCH	Dedicated Physical Channel
DPCCH	Dedicated Physical Control Channel
DPDCH	Dedicated Physical Data Channel
DS-CDMA	Direct-Sequence Code Division Multiple Access
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FER	Frame Error Rate
GF	Galois Field
MAC	Medium Access Control
Mcps	Mega Chip Per Second
MS	Mobile Station
OVSF	Orthogonal Variable Spreading Factor (codes)
PCCC	Parallel Concatenated Convolutional Code
PCH	Paging Channel
PRACH	Physical Random Access Channel
QoS	Quality of Service
RACH	Random Access Channel
RX	Receive
SCCC	Serial Concatenated Convolutional Code
SCH	Synchronisation Channel
SF	Spreading Factor
SFN	System Frame Number
SIR	Signal-to-Interference Ratio
SNR	Signal to Noise Ratio
TFCI	Transport Format Combination Indicator
TPC	Transmit Power Control
TX	Transmit
UL	Uplink (Reverse link)

4 Multiplexing, channel coding and interleaving

4.1 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, rate matching, interleaving and transport channels mapping onto/splitting from physical channels.

4.2 Transport-channel coding/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 4.2.1)
- Channel coding (see Section 4.2.3)
- Rate matching (see Section 4.2.6)
- Insertion of discontinuous transmission (DTX) indication bits (see Section 4.2.7)
- Interleaving (two steps, see Section 4.2.4 and 4.2.10)
- Radio frame segmentation (see Section 4.2.5)
- Multiplexing of transport channels (two steps, see Section 4.2.2 and 4.2.8)
- Physical channel segmentation (see Section 4.2.9)
- Mapping to physical channels (see Section 4.2.11)

The coding/multiplexing steps for uplink and downlink are shown in Figure 1 and Figure 2 respectively.

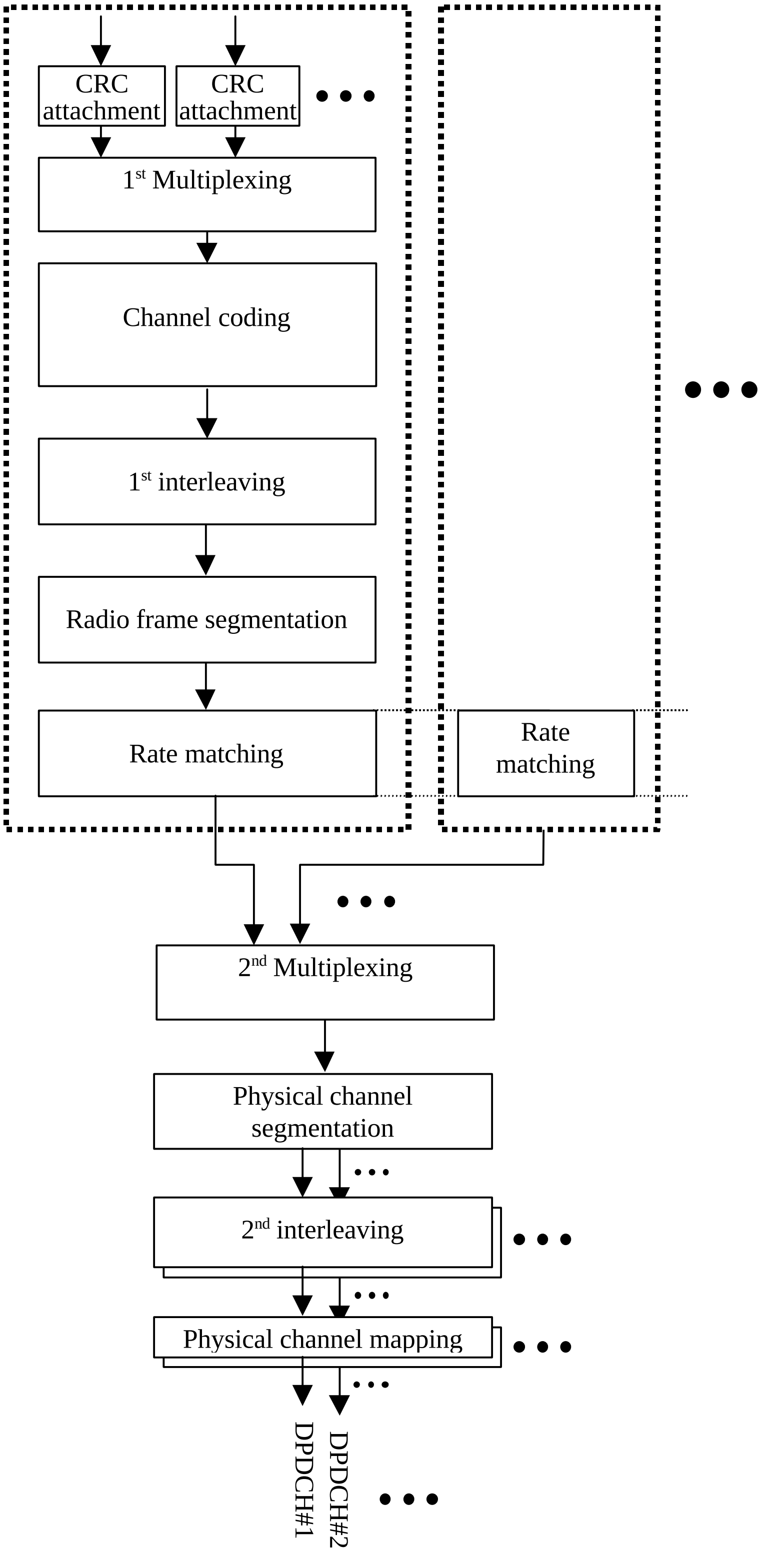

Figure 1: Transport channel multiplexing structure for uplink

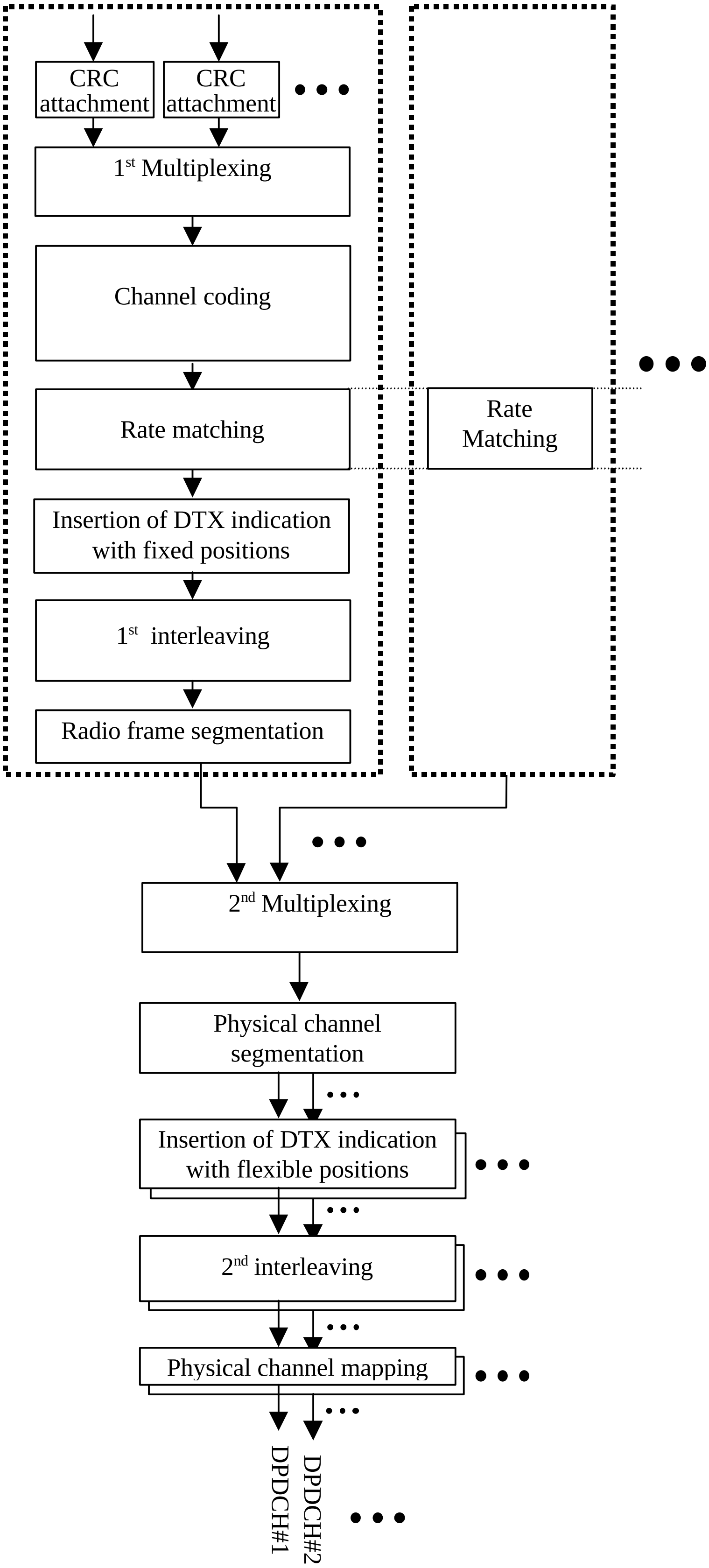

Figure 2: Transport channel multiplexing structure for downlink

<Editor's note: Code multiplexing is not used in uplink as a working assumption in WG1..>

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 code multiplexing, which corresponds to having several parallel multiplexing chains, resulting in several data stream, each mapped to one or several physical channels. This code multiplexing is used only for downlink DSCHs. For the other transport channels including downlink DCHs, the code multiplexing shall not be used.

4.2.1 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.

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:

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

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

Denote the bits in a transport block delivered to layer 1 by $b_1, b_2, b_3, \dots b_N$, and the parity bits by $p_1, p_2, \dots p_L$. 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

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

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

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

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

4.2.1.2 Relation between input and output of the Cyclic Redundancy Check

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

$w_k = b_k$ $\quad$ k = 1, 2, 3, ... N

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

4.2.2 1st Multiplexing

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: $w_{111}, w_{112}, w_{113}, \dots w_{11K}$

Bits from transport block 2 of transport channel 1: $w_{121}, w_{122}, w_{123}, \dots w_{12K}$

…

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: $w_{211}, w_{212}, w_{213}, \ldots w_{21K}$

…

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

...

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

…

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

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

$d_k = w_{11k}$ $\quad k = 1, 2, \ldots, K$

$d_k = w_{12(k-K)}$ $\quad k = K + 1, K + 2, \ldots, 2K$

…

$d_k = w_{1P(k-(P-1)K)}$ $\quad k = (P - 1)K + 1, \ldots, PK$

} TrCH 1

$d_k = w_{21(k-PK)}$ $\quad k = PK + 1, \ldots, (P + 1)K$

…

$d_k = w_{2P(k-(2P-1)K)}$ $\quad k = (2P - 1)K + 1, \ldots, 2PK$

} TrCH 2

…

$d_k = w_{R1(k-(R-1)PK)}$ $\quad k = (R - 1)PK + 1, \ldots, ((R - 1)P + 1)K$

$d_k = w_{RP(k-(RP-1)K)}$ $\quad k = (RP - 1)K + 1, \ldots, RPK$

} 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.>

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	

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.

4.2.3.1 Convolutional coding

4.2.3.1.1 Convolutional coder

- Constraint length K=9. Coding rate 1/3 and ½.
- The configuration of the convolutional coder is presented in Figure 3.
- The output from the convolutional coder shall be done in the order starting from output0, output1 and output2. (When coding rate is 1/2, output is done up to output 1).
- K-1 tail bits (value 0) shall be added to the end of the coding block.
- The initial value of the shift register of the coder shall be "all 0".

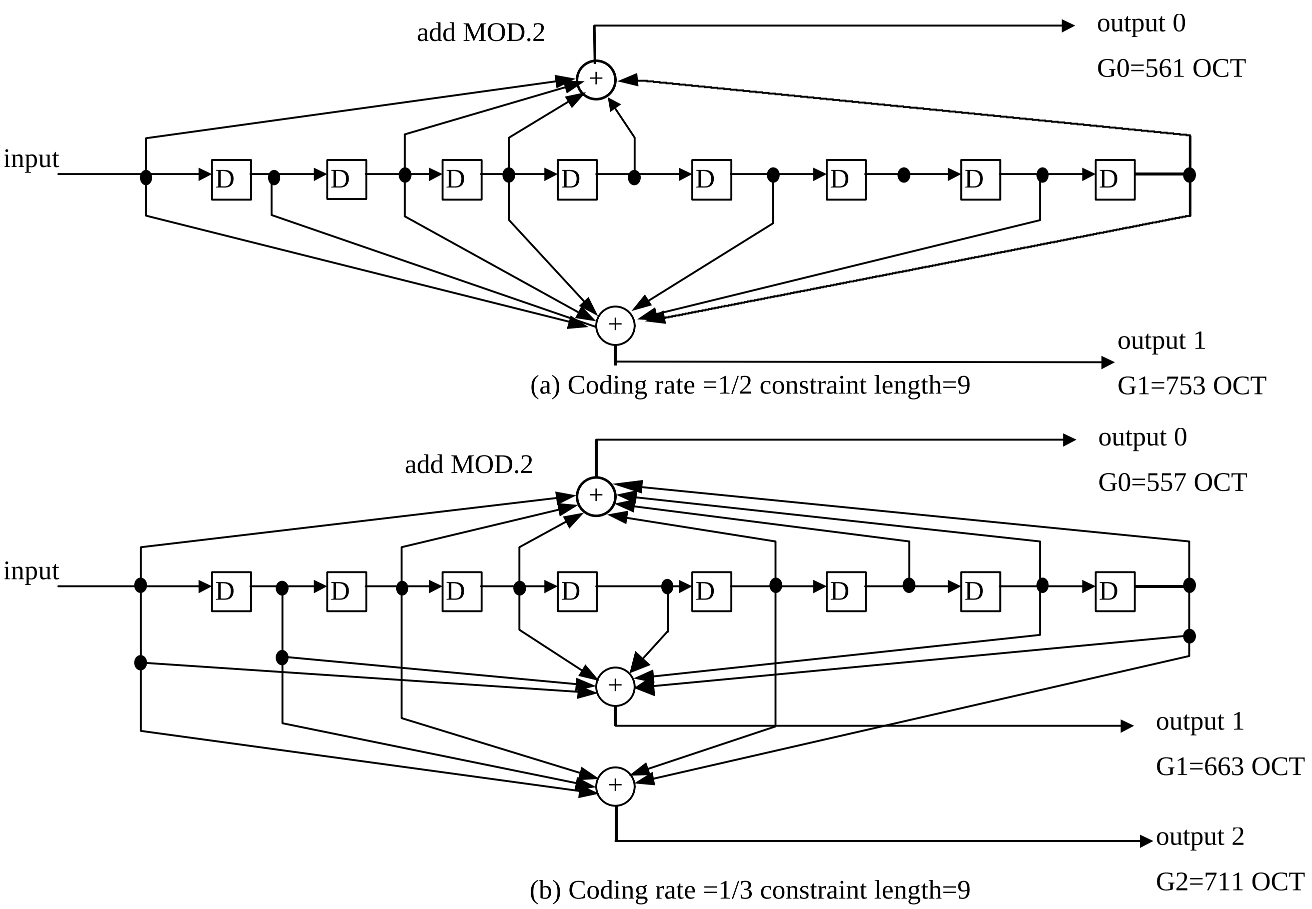

Figure 3: Convolutional Coder

4.2.3.1.2 Segmentation into code blocks for convolutional coding

<Note: It is for further study if the maximum code block size is 504 or shorter.>

If the transport blocks or multiplexed transport blocks are longer than [504] bits (including CRC bits), they are segmented before convolutional encoding. Denote the number of transport blocks before coding by P and the number of bits in each transport block or the sum of the number of bits in the multiplexed blocks by M. Note that if first multiplexing is performed, all transport blocks of a transport channel in the same transmission time interval are multiplexed together, i.e. P=1. The bits before segmentation can then be described as follows:

Bits in transport block 1before segmentation: $d_{1,1}$, $d_{1,2}$, $d_{1,3}$, … $d_{1,M}$

Bits in transport block 2 before segmentation: $d_{2,1}$, $d_{2,2}$, $d_{2,3}$, … $d_{2,M}$

…

Bits in transport block P before segmentation: $d_{P,1}$, $d_{P,2}$, $d_{P,3}$, … $d_{P,M}$

If M ≤ [504], no segmentation is performed. If M > [504] the following parameters are calculated:

Number of code blocks: S = round_up(PM / [504])

Length of coded blocks: C = round_up(PM / S)

Remainder: R = PM – S round_down(PM / S)

Number of filler bits: F = S – R, if R≠0
F = 0, if R=0

round_up(x) means the smallest integer number larger or equal to x.

round_down(x) means the largest integer number smaller or equal to x.

The F filler bits are appended to the end of the last code block before tail insertion and channel encoding. They are denoted f_1, f_2, f_3, ... f_F. The bits after segmentation are denoted by $u_{1,1}$, $u_{1,2}$, $u_{1,3}$, … $u_{1,C}$, $u_{2,1}$, $u_{2,2}$, $u_{2,3}$, ... ,$u_{2,C}$, ... $u_{S,1}$, $u_{S,2}$, $u_{S,3}$, ... ,$u_{S,C}$, and defined by the following relations:

$u_{1,k} = d_{1,k}$ k = 1, 2, 3, ... C

$u_{2,(k-C)} = d_{1,k}$ k = C + 1, C + 2, C + 3, ... 2C

...

$u_{j,(k-(j-1)C)} = d_{1,k}$ k = (j-1)C + 1, (j-1)C + 2, (j-1)C + 3, ... M

$u_{j,(k-(j-1)C)} = d_{2,(k-M)}$ k = M + 1, M + 2, M + 3, ... jC

$u_{j+1,(k-jC)} = d_{2,(k-M)}$ k = jC + 1, jC + 2, jC + 3, ... (j+1)C

...

$u_{S,(k-(S-1)C)} = d_{P,(M-C+F+k-(S-1)C)}$ k = (S - 1)C + 1, (S - 1)C + 2, (S - 1)C + 3, ... SC - F

$u_{S,(k-(S-1)C)} = f_{k-SC+F}$ k = SC – F + 1, SC - F+ 2, SC – F +3, ... SC

<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.>

4.2.3.2 Turbo coding

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,

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

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

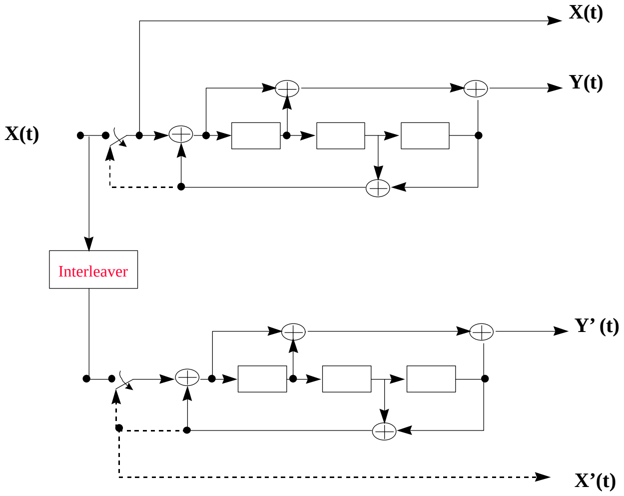

Figure 4: Structure of the 8 state PCCC encoder (dotted lines effective for trellis termination only)

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

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

The rate 2/3 is obtained by puncturing every other parity-check bit.
The inner code is a rate ½ systematic recursive convolutional code with the same previous generating matrix

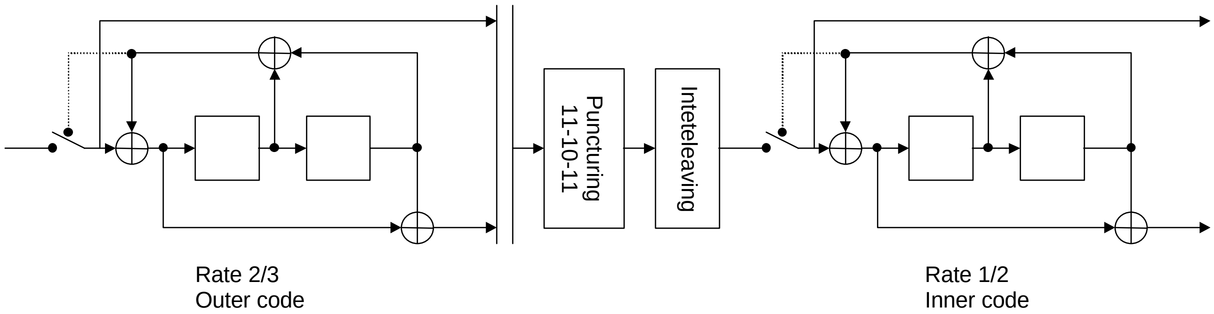

Figure 5: Structure of the 4 state SCCC encoder (dotted lines effective for trellis termination only)

4.2.3.2.2 Trellis termination for Turbo coding

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are added after the encoding of information bits.

Trellis termination for PCCC

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of Figure 4 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of Figure 4 in lower position) while the first constituent encoder is disabled.

The transmitted bits for trellis termination shall then be

X(t) Y(t) X(t+1) Y(t+1) X(t+2) Y(t+2) X’(t) Y’(t) X’(t+1) Y’(t+1) X’(t+2) Y’(t+2).

Trellis termination for SCCC

The conventional method of trellis termination is used also for SCCC in which the tail bits are taken from the shift register feedback after all bits are encoded. The tailing bits of the outer encoder are included in the interleaver. The outer code is terminated first with two additional input bits taken from the shift register feedback (dotted line of Figure 5), the outer code thus, after puncturing, outputs three additional bit that are feeded into the interleaver. After that all bits have been encoded from the inner encoder (included the interleaved tail bit of the outer encoder), two additional input bits are taken from the shift register feedback of the inner encoder producing four tail bits (dotted line of Figure 5). Thus the total overhead due to the tailing bits is 3*2+4 =10 bits

4.2.3.2.3 Turbo code internal interleaver

Figure 6 depicts the overall 8 state PCCC Turbo coding scheme including Turbo code internal interleaver. The Turbo code internal interleaver consists of mother interleaver generation and pruning. For arbitrary given block length K, one mother interleaver is selected from the 207 mother interleavers set. The generation scheme of mother interleaver is described in section 4.2.3.2.3.1. After the mother interleaver generation, *l*-bits are pruned in order to adjust the mother interleaver to the block length K. The definition of *l* is shown in section 4.2.3.2.3.2.

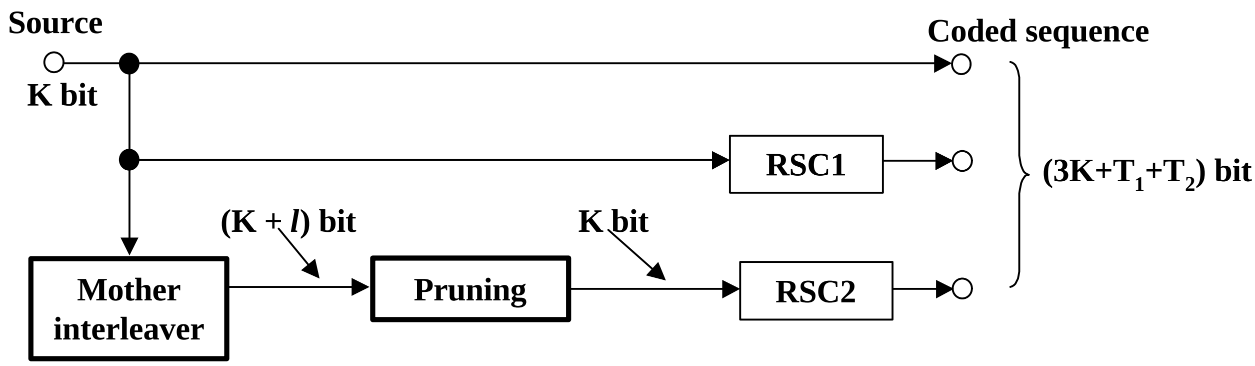

Figure 6: Overall 8 State PCCC Turbo Coding

4.2.3.2.3.1 Mother interleaver generation

The interleaving consists of three stages. In first stage, the input sequence is written into the rectangular matrix row by row. The second stage is intra-row permutation. The third stage is inter-row permutation. The three-stage permutations are described as follows, the input block length is assumed to be K (320 to 5120 bits).

<u>First Stage</u>:

(1) Determine a row number R such that

R=10 (K = 481 to 530 bits; Case-1)

R=20 (K = any other block length except 481 to 530 bits; Case-2)

(2) Determine a column number C such that

Case-1; C = $p = 53$

Csae-2;

(i) find minimum prime p such that,

0 =< (p+1)-K/R,

(ii) if (0 =< p-K/R) then go to (iii),

else C = p+1.

(iii) if (0 =< p-1-K/R) then C=p-1,

else C = p.

(3) The input sequence of the interleaver is written into the RxC rectangular matrix row by row.

<u>Second Stage:</u>

<u>A. If C = p</u>

(A-1) Select a primitive root g_0 from Table 2.

(A-2) Construct the base sequence $c(i)$ for intra-row permutation as:

$$c(i) = [g_0 \times c(i-1)] \bmod p, \; i = 1,2,\ldots,(p-2)., \; c(0) = 1.$$

(A-3) Select the minimum prime integer set {q_j } (j=1,2,…R-1) such that

g.c.d{q_j, p-1} =1

$q_j > 6$

$q_j > q_{(j-1)}$

where g.c.d. is greatest common divider. And $q_0 = 1$.

(A-4) The set {q_j} is permuted to make a new set {p_j} such that

$p_{P(j)} = q_j$, j = 0, 1, …. R-1,

where P(j) is the inter-row permutation pattern defined in the third stage.

(A-5) Perform the j-th (j = 0, 1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$, i =0, 1,2,…, (p-2)., and $c_j(p\text{-}1) = 0$,

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

<u>B. If C = *p*+1</u>

(B-1) Same as case A-1.

(B-2) Same as case A-2.

(B-3) Same as case A-3.

(B-4) Same as case A-4.

(B-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$, i =0,1,2,…, (p-2)., $c_j(p\text{-}1) = 0$, and $c_j(p) = p$,

(B-6) If (K = C x R) then exchange $c_{R-1}(p)$ with $c_{R-1}(0)$.

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

<u>C. If C = *p*-1</u>

(C-1) Same as case A-1.

(C-2) Same as case A-2.

(C-3) Same as case A-3.

(C-4) Same as case A-4.

(C-5) Perform the j-th (j = 0,1, 2, …, R-1) intra-row permutation as:

$c_j(i) = c([i \times p_j] \bmod (p-1))$ -1, i =0,1,2,…, (p-2),

where $c_j(i)$ is the input bit position of i-th output after the permutation of j-th row.

<u>**Third Stage:**</u>

(1) Perform the inter-row permutation based on the following P(j) (j=0,1, ..., R-1) patterns, where P(j) is the original row position of the j-th permuted row.

P_A: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11} for R=20

P_B: {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 16, 13, 17, 15, 3, 1, 6, 11, 8, 10} for R=20

P_C: {9, 8, 7, 6, 5, 4, 3, 2, 1, 0} for R=10

The usage of these patterns is as follows:

Block length K: P(j)

320 to 480-bit: P_A

481 to 530-bit: P_C

531 to 2280-bit: P_A

2281 to 2480-bit: P_B

2481 to 3160-bit: P_A

3161 to 3210-bit: P_B

3211 to 5120-bit: P_A

(2) The output of the mother interleaver is the sequence read out column by column from the permuted R $\times$ C matrix.

Table 2: Table of prime p and associated primitive root

p	g_o	*p*	g_o	*p*	g_o	*p*	g_o	*p*	g_o	*p*	g_o	*p*	g_o	*p*	g_o
17	3	59	2	103	5	157	5	211	2	269	2	331	3	389	2
19	2	61	2	107	2	163	2	223	3	271	6	337	10	397	5
23	5	67	2	109	6	167	5	227	2	277	5	347	2	401	3
29	2	71	7	113	3	173	2	229	6	281	3	349	2	409	21
31	3	73	5	127	3	179	2	233	3	283	3	353	3		
37	2	79	3	131	2	181	2	239	7	293	2	359	7		
41	6	83	2	137	3	191	19	241	7	307	5	367	6		
43	3	89	3	139	2	193	5	251	6	311	17	373	2		
47	5	97	5	149	2	197	2	257	3	313	10	379	2		
53	2	101	2	151	6	199	3	263	5	317	2	383	5		

4.2.3.2.3.2 Definition of number of pruning bits

The output of the mother interleaver is pruned by deleting the *l*-bits in order to adjust the mother interleaver to the block length K, where the deleted bits are non-existent bits in the input sequence. The pruning bits number *l* is defined as:

$l = R \times C - K$,

where R is the row number and C is the column number defined in section 4.2.3.2.3.1.

4.2.3.2.4 Encoding blocks for Turbo code

Input data blocks for a turbo encoder consist of the user data and possible extra data being appended to the user data before turbo encoding. The encoding segments for a turbo encoder are defined in terms of systematic bits. The segment includes the user data, a possible error detection field (CRC), possible filler bits, and the termination. The maximum encoding segment length for turbo coding is 5120 bits. The Algorithm for combining and segmentation is as follows:

Inputs:

N_{DATA} size of input data block to turbo encoder

N_{TAIL} number of tail bits to be appended to the encoding segments (termination)

Outputs:

N_S number of segments

N_{TB} number of bits in the turbo encoder input segments

N_{FILL} number of filler (zero) bits in the last turbo encoder input segment

Do:

1. Let N_S = round_up(N_{DATA} / (5120 – N_{TAIL}))
2. Let N_{TB} = round_up (N_{DATA} / N_S) + N_{TAIL};
3. Let N_{REM} = remainder of N_{DATA} / N_S;
4. If N_{REM} not equal to 0 then insert N_{FILL} = (N_S – N_{REM}) zero bits to the end of the input data else N_{FILL} = 0.
5. End.

Here *round_up(x)* stands for an smallest interger number being larger or equal to *x*.

All turbo encoder input segments are of equal size and therefore the same turbo interleaver can be used for all turbo segments. A number of systematic bits over an entire channel interleaving block at output of the encoder is

$$N_S * (\text{round\_up}(N_{DATA} / N_S) + N_{TAIL}).$$

The N_{FILL} filler bits are padded to the end of the last encoding segment in order to make the last segment equal size to the precedent ones. The filler bits are encoded.

4.2.4 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 3.
(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 3, 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 3

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}

4.2.5 Radio frame segmentation

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 for uplink and 2nd multiplexing for downlink in block-wise order at every 10 msec.

Figure B-1 and B-2 illustrate data flow from 1st interleaver down to 2nd interleaver in both uplink and downlink channel coding and multiplexing chains. 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 notations in Figure B-1 and B-2, 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 $i = 0, 1, 2, \ldots, N$

4.2.5.1 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 $(1, 2, 3, \ldots, T_i)$ *for a given* i^{th} *channel coding and multiplexing chain*

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

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

$(L_i + r_i)/ T_i = K_i$ *// Target radio frame size for downlink*

If ri $\neq 0$ *then*
  For each t $(\geq T_i\text{-}r_i+1)$
  Add one filler bit to the end of t^{th} *radio frame*
  End
End If

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

Bits after radio frame segmentation in the $(T_i - r_i+1)^{th}$ 10 msec time interval: $(t=T_i - r_i+1)$
$c_{ij} = b_{i,(j+(T_i-r_i)(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i\text{-}1$
$c_{ij} = filler\_bit(0/1)$ $j= (L_i+r_i)/T_i$ *(filler bit)*
…

Bits after radio frame segmentation in the T_i^{th} 10 msec time interval: $(t=T_i)$
$c_{ij} = b_{i,(j+(T_i-1)(L_i+r_i)/T_i)}$ $j=1,2, \ldots, (L_i+r_i)/T_i\text{-}1$
$c_{ij} = filler\_bit(0/1)$ $j=(L_i+r_i)/T_i$ *(filler bit)*

4.2.6 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.

Notation used in Section 4.2.6 and subsections:

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

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

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

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

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*.

T: Number of transport channels in the CCTrCH.

Z_{mj}: Intermediate calculation variable.

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

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

q: Average puncturing distance.

$I_F(k)$: 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): The shift of the puncturing pattern for radio frame *k*.

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

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 .. *T*, *where ë û means round downwards*

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

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. The maximum amount of puncturing that can be applied is signalled at connection 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 N_{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 $N_{data} - \sum_{i=1}^{T} \frac{RM_i}{\min_l \{RM_l\}} \cdot N_{ij}$ is non negative }

If the smallest element of SET1 requires just one DPDCH then

$N_{data,j}$ = min SET1

else

SET2 = { 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 }

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 DPDCH 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 transport channel *i* is calculated with the relations given in Section 4.2.6 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 4.2.6.3, where $DN = DN_{ij}$ and $N = N_{ij}$.

Additionally, the following parameters are needed:

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 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

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.

NOTE: The rule to convert the rate matching attributes in downlink to the parameters input to rate matching pattern algorithm are working assumption. So, it remains to be verified that they hold for all possible transport format combinations. It has been identified that the case when the transport format combination with highest rate include a transport format with zero bits need special treatment.

Radio frame segmentation is performed after 1st interleaving and N_{ij} is therefore calculated as:

$l = TF_i(j)$ and $N_{ij} = \left\lceil \frac{N_{i,l}^{TTI}}{F_i} \right\rceil$

The number of bits repeated or punctured, DN_{iL}, within one radio frame for each transport channel is calculated for the transport format combination L with highest bitrate with the relations given in Section 4.2.6.

If fix positions of the transport channels in the radio frame are used then the same DN_{ij} is used for all transport format combinations and the last part of the rate-matching pattern omitted. That is to say for all transport format combinations j we have:

$$\Delta N_{ij} = \Delta N_{iL}$$

When flexible positions of the transport channels are used, the number of bits DN_{ij} repeated or punctured for all transport format combinations j other than L is calculated as:

$$\Delta N_{ij} = \left\lfloor \frac{\Delta N_{iL}}{N_{iL}} \cdot N_{ij} \right\rfloor$$

For each transmission time interval, the rate-matching pattern is calculated with the algorithm in Section 4.2.6.3. The following parameters are used as input:

$l = TF_i(j)$ and $\Delta N = \Delta N_{il}^{TTI} = F_i \Delta N_{ij}$

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

S=0.

4.2.6.3 Rate matching algorithm

Denote the bits before rate matching by:

$c_1, c_2, c_3, \ldots c_N$

The rate matching rule is as follows:

if puncturing is to be performed

$y = -DN$

```
        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 = DN
        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.

4.2.7 Insertion of discontinuous transmission (DTX) indication bits

In the downlink, DTX is used to fill up the frame. The insertion point of DTX indication bits depends on whether fixed or flexible positions of the transport channels are used in the radio frame. It is up to the UTRAN to decide whether fixed or flexible positions are used during the connection for each transport channel. DTX indication bits only indicate when the transmission should be turned off, they are not transmitted.

4.2.7.1 Insertion of DTX indication bits with fixed positions

This step of inserting DTX indication bits is used only for those transport channels which use fixed position scheme. With fixed position scheme a fixed number of bits is reserved for transport channel in the radio frame.

Denote the bits from rate matching block by r_1, r_2, r_3, …, r_N, where N is the number of these bits per L*10 ms, which is the transmission time interval. r_1 is the first input bit to this block and r_N is the last input bit into this block. Denote the number of bits reserved from one radio frame for this transport channel (or fix rate TrCHs with the same transport format attributes) by M. After inserting the DTX indication bits, there are three valued symbols s_k. They can be described as follows:

$s_k = r_k$ k=1,2,3,…,N

$s_k = x$ k=N+1, N+2, N+3, …, LM

where DTX indication bits are denoted by x. Here $r_k \in \{0,1\}$and $x \notin \{0,1\}$. s_1 is the first output symbol from this block and s_{LM} is the last output symbol from this block.

4.2.7.2 Insertion of DTX indication bits with flexible positions

This step of inserting DTX indication bits is used only if transport channels use flexible position scheme. In flexible position scheme transport channels have been concatenated one after another in the 2nd multiplexing step. The DTX indication bits shall be placed at the end of the frame, after all the encoded data bits.

Denote the bits from physical channel segmentation into one physical channel by p_1, p_2, p_3, …, p_N, where N is the number of these bits per one radio frame. p_1 is the first input bit to this block and p_N is the last input bit to this block. Denote the number of bits that can be fitted to DPDCH field of one radio frame by M. After insertion of the DTX indication bits, there are three valued symbols s_k. They can be described as follows:

$s_k = p_k$ k=1,2,3,…,N

$s_k = x$ k=N+1, N+2, N+3, …, M

where DTX indication bits are denoted by x. Here $p_k \in \{0,1\}$and $x \notin \{0,1\}$. s_1 is the first output symbol from this block and s_M is the last output symbol from this block.

4.2.8 2nd Multiplexing

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 B-1 and B-2 illustrate data flow from 1st interleaver down to 2nd interleaver in both uplink and downlink channel coding and multiplexing chains. In the figures, 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 B-1 and B-2, 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.

4.2.8.1 Second multiplexing in uplink

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

Bits from rate matching 1: $c_{11}, c_{12}, \dots c_{1K_1}$
Bits from rate matching 2: $c_{21}, c_{22}, \dots c_{2K_2}$
Bits from rate matching 3: $c_{31}, c_{32}, \dots c_{3K_3}$
...
Bits from rate matching *N*: $c_{N1}, c_{N2}, \dots c_{NK_N}$

The bits after second multiplexing are denoted by $d_1, d_2, \dots, d_P$
and defined by the following relationships:

For $j=1,2,3 \dots,P$ *where* $P=K_1+K_2+ \dots +K_N$

$d_j = c_{1j}$ $\quad j=1,2, \dots K_1$
$d_j = c_{2,(j-K_1)}$ $\quad j= K_1+1, K_1+2, \dots, K_1+K_2$
$d_j = c_{3,(j-(K_1+K_2))}$ $\quad j=(K_1+K_2)+1, (K_1+K_2)+2, \dots, (K_1+K_2)+ K_3$
…
$d_j = c_{N,(j-(K_1+K_2+\dots+K_{N-1}))}$ $\quad j=(K_1+K_2+\dots+ K_{N-1})+1, (K_1+K_2+\dots+ K_{N-1})+2, \dots, (K_1+K_2+\dots+ K_{N-1})+K_N$

4.2.8.2 Second multiplexing in downlink

The bits before second multiplexing in downlink are described as follows:

Bits from radio frame segmentation 1: $c_{11}, c_{12}, \ldots c_{1K_1}$
Bits from radio frame segmentation 2: $c_{21}, c_{22}, \ldots c_{2K_2}$
Bits from radio frame segmentation 3: $c_{31}, c_{32}, \ldots c_{3K_3}$
…
Bits from radio frame segmentation 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 relationship:

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

$d_j = c_{1j} \quad j=1,2, \ldots K_1$
$d_j = c_{2,(j-K_1)} \quad j= K_1+1, K_1+2, \ldots, K_1+K_2$
$d_j = c_{3,(j-(K_1+K_2))} \quad j=(K_1+K_2)+1, (K_1+K_2)+2, \ldots, (K_1+K_2)+ K_3$
…
$d_j = c_{N,(j-(K_1+K_2+\ldots+K_{N-1}))} \quad j=(K_1+K_2+\ldots+ K_{N-1})+1, (K_1+K_2+\ldots+ K_{N-1})+2, \ldots, (K_1+K_2+\ldots+ K_{N-1})+ K_N$

4.2.9 Physical channel segmentation

<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 10ms interval.

Figure B-1 and B-2 illustrate data flow from 1st interleaver down to 2nd interleaver in both uplink and downlink channel coding and multiplexing chains. 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 notations in Figure B-1 and B-2, 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:
$e_{1j} = d_j \quad j=1,2,\ldots,P/M$

The second physical channel bits after physical channel segmentation:
$e_{2j} = d_{(j+P/M)} \quad j=1,2, \ldots, P/M$
…

The M^{th} physical channel bits after physical channel segmentation:
$e_{Mj} = d_{(j+(M-1)P/M)} \quad j=1,2, \ldots, P/M$

4.2.10 2nd interleaving

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,

$$K_2 \le R_2 \times C_2.$$

(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, 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:

$$l_2 = R_2 \times C_2 - K_2.$$

Table 4

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}

<Editor's note: Above Table is a working assumption.>

4.2.11 Physical channel mapping

4.2.11.1 Uplink

On the uplink, transport data after 2nd interleaving is mapped onto one DPDCH. Continuous transmission is applied for uplink DPDCH at all times.

4.2.11.2 Downlink

On the downlink, transport data after 2nd interleaving is mapped onto data fields in one DPDCH, which is defined in TS 25.211. If the total bit rate after transport channel multiplexing is not identical to the total channel bit rate of the allocated dedicated physical channels, discontinuous transmission is used.

- If transport data is less than the number of DPDCH bits in a radio frame, the DPDCH transmission can be turn off for data absent.
- The transmission of the DPDCH 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 UTRAN decides whether the positions of the transport channels should be fixed or flexible.
- Pilot and TPC symbols are always transmitted regardless of the data existence.

4.2.12 DSCH transmission when associated with DCH

- The data stream on DSCH shall be transmitted continuously over the 10 ms allocation period with no DTX on slot period.
- The spreading factor is indicated with the TFCI or with higher layer signaling on DCH.

- Rate matching is implemented as in uplink, when there is data to transmit the 10 ms frame is fully filled with no DTX. The rates for the data as well as rate matching parameters are pre-negotiated at higher layers and are all part of the TFCI indication for particular data rate with particular spreading code.

4.2.13 Transport format detection

Transport format detection can be performed both with and without Tansport Format Combination Indicator (TFCI). If a TFCI is transmitted, the receiver detects the transport format combination from the TFCI. When no TFCI is transmitted, so called blind transport format detection is used, i.e. the receiver side detects the transport format combination using some information, e.g. received power ratio of DPDCH to DPCCH, CRC check results.

For uplink, the blind transport format detection is an operator option. For downlink, the blind transport format detection can be applied with convolutional coding, the maximum number of different transport formats and maximum data rates allowed shall be specified.

4.2.13.1 Blind transport format detection

Examples of blind transport format detection methods are given in Annex A.

4.2.13.2 Explicit transport format detection based on TFCI

4.2.13.2.1 Transport format combination indicator

Transport Format Combination Indicator (TFCI) informs the receiver of the number of bits in each frame of each of the services currently in use. As soon as a certain bit-rate is known, the number of code channels, the spreading factor and the puncturing/repetition rate is immediately known from the rules described in section 4.2.6.

This document therefore only explains the mapping from TFCI bits to TFCI service rate combinations.

A connection may in general include the variable-rate services S_1, S_2, …, S_K. Each service S_i has a set of possible transport format combination indicators $TF_{i,1}$,$TF_{i,2}$,…$TF_{i,Li}$:

S_1: $TF_{1,1}$, … , $TF_{1,L1}$

S_2: $TF_{2,1}$, … , $TF_{2,L2}$

…

S_K: $TF_{K,1}$, … , $TF_{K,LK}$

This gives L=L1xL2x…xLK service rate combinations, and thus it is required that L is less than or equal to 64 with the default TFCI word or 1024 with the extended TFCI word.

These service rate combinations shall be mapped to a certain service rate combination number, *m*, in the following way:

```
For j=K:-1:1,
  SRC[j]= m MOD L[j];
  m = m DIV L[j];
End;
```

From this pseudo-code, given a service rate combination number, i.e. a certain combination of TFCI bits, *m*, SRC contains the rates of each of the K services. The integer values used for *m* shall be consecutive, starting from 0. Note that this code gives the mapping rule from *m* to SRC, i.e. the rule used in the receiving side. The mapping rule from SRC to *m*, i.e. the transmitting side rule, is [TBD].

4.2.14 Coding procedure

4.2.14.1 SFN(System Frame Number)

- SFN indicates super frame synchronisation. It is broadcasted in BCH. (See TS 25.211)

<Editor's note: Length of SFN is FFS. It will be determined according to requirement from WG2and WG3>

- SFN is multiplexed with a BCH transport block (see Figure 7).
- SFN is applied CRC calculation and FEC with BCH transport block.

Figure 7: SFN multiplexing

4.3 Coding for layer 1 control

4.3.1 Coding of Transport-format-combination indicator (TFCI)

The number of TFCI bits is variable and is set at the beginning of the call via higher layer signalling. Encoding of the TFCI bits depends on the number of them. If there are at most 6 bits of TFCI, the channel encoding is done as described in section 4.3.1.1. Correspondingly, if the TFCI word is extended to 7-10 bits the channel encoding is done as explained in the section 4.3.1.2. For improved TFCI detection reliability, in downlink, repetition is used by increasing the number of TFCI bits within a slot.

4.3.1.1 Coding of default TFCI word

If the number of TFCI bits is up to 6, the TFCI bits are encoded using punctured biorthogonal (30, 6) block code. The coding procedure is as shown in Figure 8.

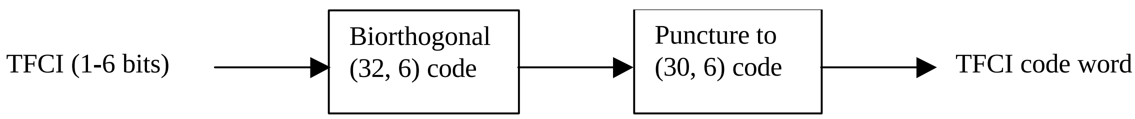

Figure 8: Channel coding of TFCI bits

If the TFCI consist of less than 6 bits, it is padded with zeros to 6 bits, by setting the most significant bits to zero. The receiver can use the information that not all 6 bits are used for the TFCI, thereby reducing the error rate in the TFCI decoder. The length of the TFCI code word is 30 bits. Thus there are 2 bits of (encoded) TFCI in every slot of the radio frame.

The TFCI bits are first encoded using biorthogonal (32, 6) code. The code words of the biorthogonal block code are from the level 32 of the code three of OVSF codes defined in document TS 25.213. The code words, $C_{32,I}$, I = 1,…,32, form an orthogonal set, $S_{C_{32}} = \{C_{32,1}, C_{32,2}, ..., C_{32,32}\}$, of 32 code words of length 32 bits. By taking the binary complements of the code words of $S_{C_{32}}$, another set, $\bar{S}_{C_{32}} = \{\bar{C}_{32,1}, \bar{C}_{32,2}, ..., \bar{C}_{32,32}\}$ is formed. These two sets are mutually biorthogonal yielding total of 64 different code words.

Mapping of the TFCI bits to the biorthogonal code words is done as shown in the Figure 9.

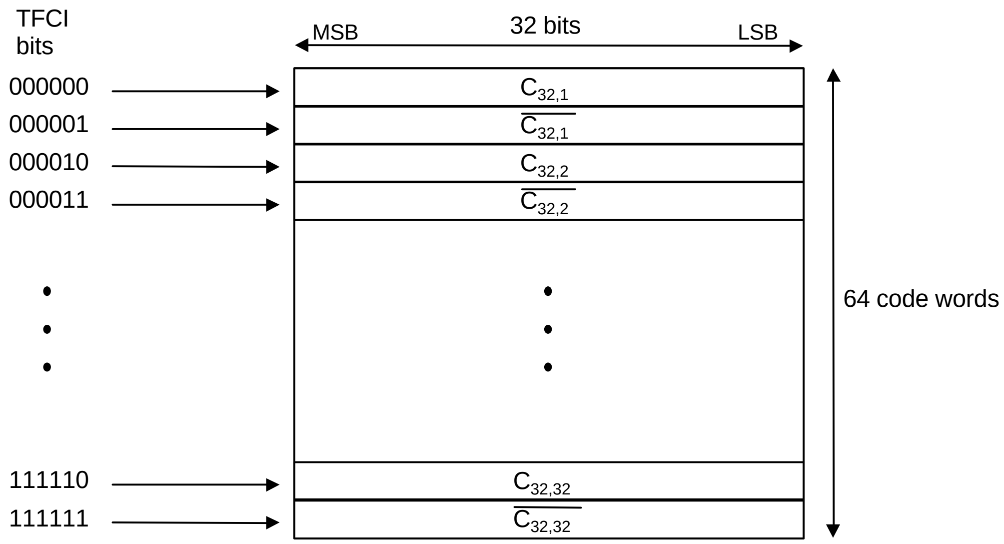

Figure 9: Mapping of TFCI bits to biorthogonal code words

Biorthogonal code words, $C_{32,i}$ and $\overline{C}_{32,i}$, are encoded into TFCI code words of length 30 bits by puncturing the two least significant bits (i.e. the two last bits on right in the Figure 9).

4.3.1.2 Coding of extended TFCI word

If the number of TFCI bits is 7-10 the TFCI information field is split into two words of length 5 bits as shown in the following formula: .

$n := \lfloor \sqrt{TFCI} \rfloor$; n is the largest integer being smaller than or equal to the square root of the transmitted TFCI value.

if $TFCI < n^2 + n$

then $Word1 := n$; $Word2 := TFCI - n^2$

else $Word2 := n$; $Word1 := n^2 + 2n - TFCI$

Both of the words are first encoded using biorthogonal (16, 5) block code. The code words of the biorthogonal (16, 5) code are from two mutually biorthogonal sets, $S_{C_{16}} = \{C_{16,1}, C_{16,2}, ..., C_{16,16}\}$ and its binary complement, $\overline{S}_{C_{16}} = \{\overline{C}_{16,1}, \overline{C}_{16,2}, ..., \overline{C}_{16,16}\}$. Words of set $S_{C_{16}}$ are from the level 16 of the code three of OVSF codes defined in document TS 25.213. The mapping of information bits to code words is shown in the Table 5.

Table 5: Mapping of information bits to code words for biorthogonal (16, 5) code

Information bits	Code word
00000	$C_{16,1}$
00001	$\overline{C}_{16,1}$
00010	$C_{16,2}$
...	...
11101	$\overline{C}_{16,15}$
11110	$C_{16,16}$
11111	$\overline{C}_{16,16}$

Biorthogonal code words, $C_{16,i}$ and $\overline{C}_{16,i}$, are then encoded into TFCI code words of length 15 bits by puncturing the least significant bit (i.e. the rightmost bit).

4.3.2 Operation of Transport-format-combination indicator (TFCI) in soft handover

In the case of DCH in soft handover situation, each Node B shall transmit the identical (30,6) code word for the UE.

In the case of extended TFCI coding, the Node B shall operate with one of the following modes:

- Both words are identical from all links
- If one of the links is associated with a DSCH, the TFCI code word may be split in such a way that the code word relevant for TFCI activity indication is not transmitted from every Node B. The use of such a functionality shall be indicated by higher layer signalling.

4.3.3 Interleaving of TFCI words

4.3.3.1 Interleaving of default TFCI word

As only one code word for TFCI of maximum length of 6 bits is needed no channel interleaving for the encoded bits are done. Instead, the bits of the code word are directly mapped to the slots of the radio frame as depicted in the Figure 10. Within a slot the more significant bit is transmitted before the less significant bit.

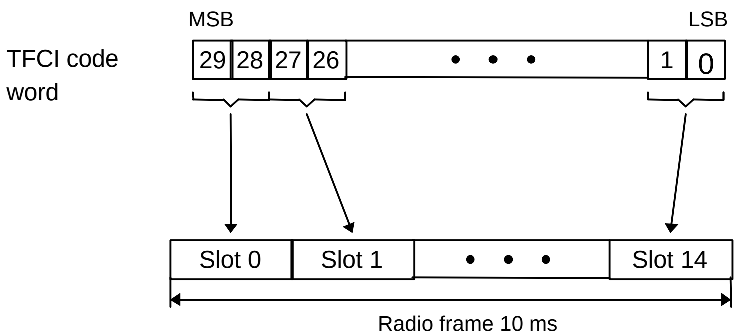

Figure 10: Time multiplexing of code words of (30, 6) code to the slots of the radio frame

4.3.3.2 Interleaving of extended TFCI word

After channel encoding of the two 5 bit TFCI words there are two code words of length 15 bits. They are interleaved and mapped to DPCCH as shown in the Figure 11. Note that $b_{1,i}$ and $b_{2,i}$ denote the bit *i* of code word 1 and code word 2, respectively.

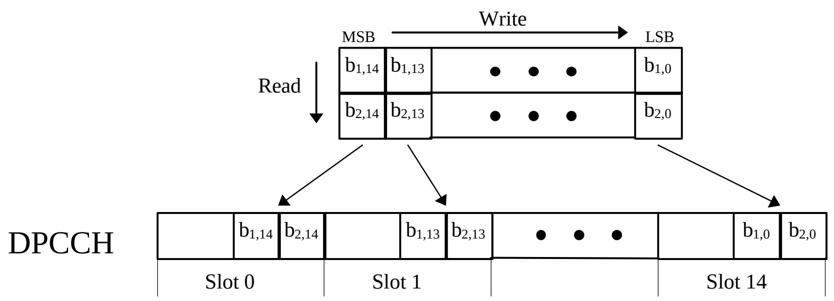

Figure 11: Interleaving of extended TFCI code words

4.4 Coding of compressed mode

In compressed mode, slots N_{first} to N_{last} are not used for transmission of data. As illustrated in Figure 12, which shows the example of fixed idle length position with single frame method (see section 4.4.3), the instantaneous transmit power is increased in the compressed frame in order to keep the quality (BER, FER, etc.) unaffected by the reduced processing gain. The amount of power increase depends on the transmission time reduction method (see section 4.4.2). What frames are compressed, are decided by the network. When in compressed mode, compressed frames can occur periodically, as illustrated in Figure 12, or requested on demand. The rate and type of compressed frames is variable and depends on the environment and the measurement requirements.

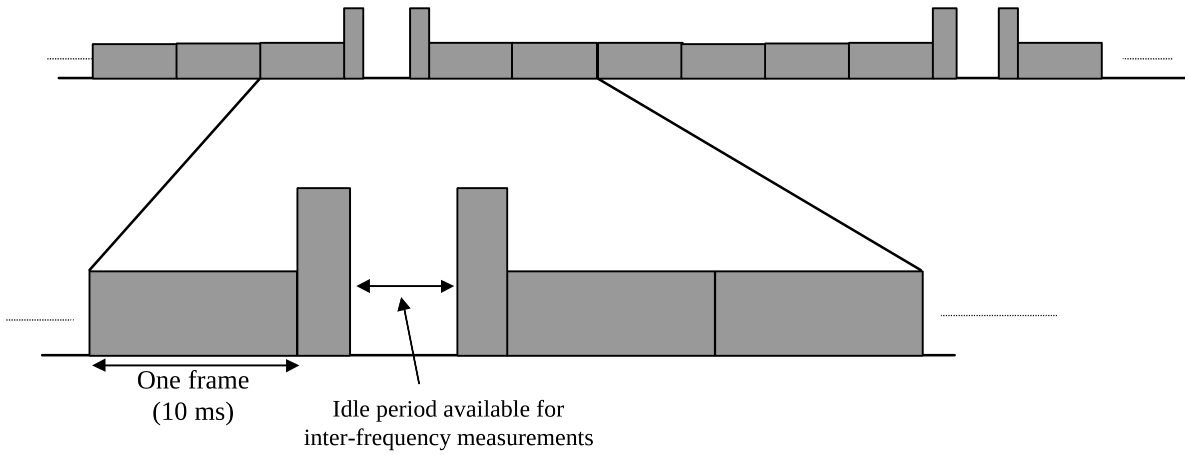

Figure 12: Compressed mode transmission

4.4.1 Frame structure types in downlink

There are two different types of frame structures defined for downlink compressed transmission. Type A is the basic case, which maximises the transmission gap length. Type B, which is more optimised for power control, can be used if the requirement of the transmission gap length allows that. Slot structure for uplink compressed mode is for further study.

- With frame structure of type A, BTS transmission is off from the beginning of TFCI field in slot N_{first}, until the end of Data2 field in slot N_{last} (Figure 13(a)).
- With frame structure of type B, BTS transmission is off from the beginning of Data2 field in slot N_{first}, until the end of Data2 field in slot N_{last} (Figure 13(b)) Dummy bits are transmitted in the TFCI and Data1 fields of slot N_{first}, and BTS and MS do not use the dummy bits. Thus BTS and MS utilize only the TPC field of N_{first}.

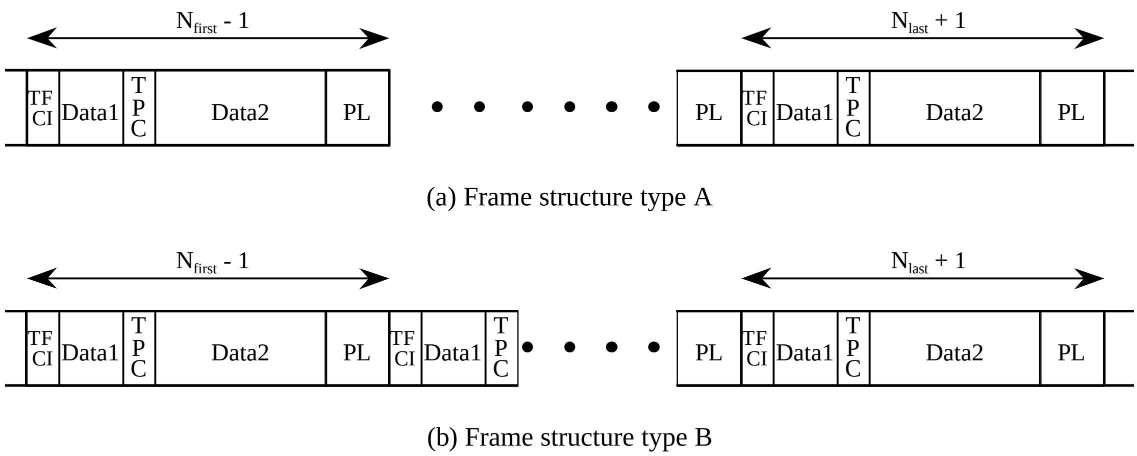

Figure 13: Frame structure types in downlink compressed transmission

4.4.2 Transmission time reduction method

When in compressed mode, the information normally transmitted during a 10 ms frame is compressed in time. The mechanism provided for achieving this is either changing the code rate, which means puncturing in practice, or the reduction of the spreading factor by a factor of two.-The maximum idle length is defined to be 5 ms per one 10 ms frame.

4.4.2.1 Method A1: By puncturing, basic case

During compressed mode, rate matching (puncturing) is applied for making short transmission gap length in one frame. Algorithm of rate matching (puncturing) described in Section 4.2.6 is used. The maximum transmission gap length allowed to be achieved with this method is the case where the code rate is increased from 1/3 to 1/2 by puncturing, which corresponds to 2 - 5 time slots per 10 ms frame, depending on the rate matching conditions that would be used in the non-compressed frame case. The explanation of the rate matching conditions are given below:

Example 1: If rate matching conditions in the non-compressed frame case would be such that maximum puncturing =0.2 would be used, then during compressed mode further puncturing of 1-(2/(3*(1-0.2))) =0.17 is allowed which corresponds to 0.17*16=2.7 => 2 time slots.

Example 2: If rate matching conditions in the non-compressed frame case would be such that no puncturing would be used, then during compressed mode puncturing of 1-(2/3)=0.33 is allowed which corresponds to 0.33*16=5.3 => 5 time slots.

4.4.2.2 Method A2: By puncturing, for services that allow larger delay

Other methods of supporting compressed mode may be considered as options. For example, with services that allows for a larger delay, e.g. data services with interleaving over several frames, multiple frames might be compressed together in order to create a short measurement slot. As an example, for a 2 Mbps service, with interleaving of 5 frames (50 ms), a 5 ms idle slot can be created by puncturing only 10% of 5 frames, as illustrated in Figure 14.

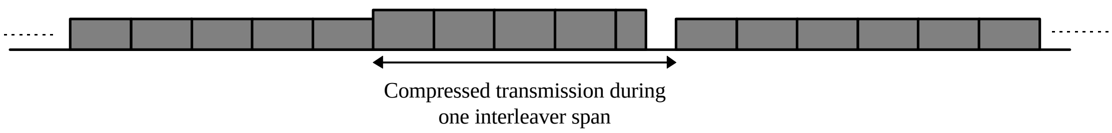

Figure 14: Multi-frame compressed mode for long-delay services

4.4.2.3 Method B: By reducing the spreading factor by 2

During compressed mode, the spreading factor (SF) can be reduced by 2 to enable the transmission of the information bits in the remaining time slots of a compressed frame. This can accommodate up to 50% idle slots per frame which is the maximum compression factor required. Additional rate matching is required if there are less than 50% idle slots. Reducing the spreading factor will normally be used if rate matching alone is not sufficient to transmit all information bits in compressed mode. Decrease of the spreading factor could involve change of the scrambling code, but when such an option could be used is for further study. Use of this method for uplink compressed mode is for further study.

4.4.3 Transmission gap position

Transmission gaps can be placed at both fixed position and adjustable position for each purpose such as interfrequency power measurement, acquisition of control channel of other system/carrier, and actual handover operation.

4.4.3.1 Fixed transmission gap position

Transmission gap lengths can be placed on fixed positions. The fixed transmission gap positions are located on the center of a frame or on the center of two connected frames as shown in Figure 15. Table 6 shows the parameters for the fixed transmission gap position case.

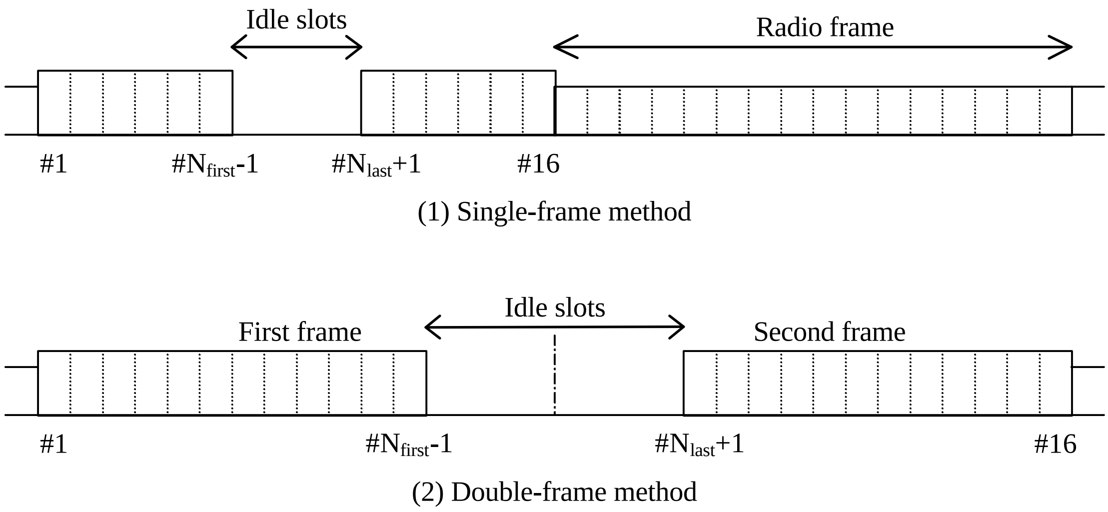

Figure 15: Fixedtransmission gap lengths position

Table 6: Parameters for fixed transmission gap position

	Single-frame method		Double-frame method	
Transmission gap length (slot)	**N_{first}**	**N_{last}**	**N_{first}**	**N_{last}**
3	8	10	16 in first frame	2 in second frame
4	7	10	15 in first frame	2 in second frame
5	7	11	15 in first frame	3 in second frame
6	6	11	14 in first frame	3 in second frame
8	5	12	13 in first frame	4 in second frame
10	N.A.	N.A.	12 in first frame	5 in second frame
16	N.A.	N.A.	9 in first frame	8 in second frame

4.4.3.2 Adjustable transmission gap position

Position of transmission gaps can be adjustable/relocatable for some purpose e.g. data acquisition on certain position as shown in Figure 16. Parameters of the adjustable transmission gap lengths positions are calculated as follows:

N_{idle} is the number of consecutive idle slots during compressed mode, as shown in Table 6,

N_{idle} = 3,4,5,6,8,10,16.

N_{first} specifies the starting slot of the consecutive idle slots,

N_{first} = 1,2,3,…,16.

N_{last} shows the number of the final idle slot and is calculated as follows;

If $N_{first} + N_{idle} <= 17$, then $N_{last} = N_{first} + N_{idle} - 1$ (in the same frame),

If $N_{first} + N_{idle} > 17$, then $N_{last} = N_{first} + N_{idle} - 17$ (in the next frame).

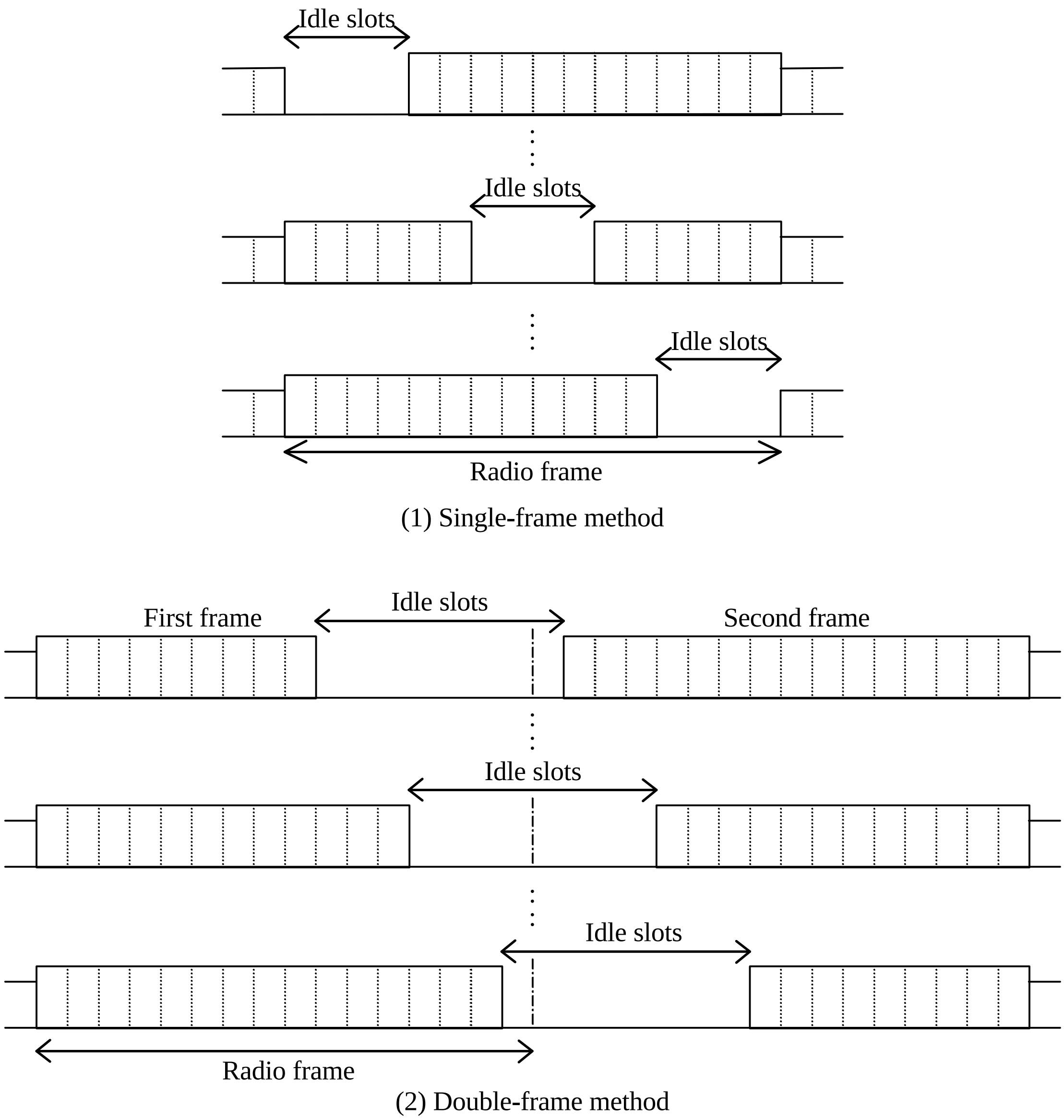

Figure 16: Concept of adjustable transmission gap lengths position

<Editors note: Adjustment needed>

4.4.3.3 Parameters for compressed mode

< Editor's note: WG1 suggestion is that there is need for further clarifications in Table 7 (e.g. rationales between change of coding rate/puncturing/change of spreading factor and idle time size, spreading factor range for different modes, etc.).>

Table 7 shows the detailed parameters for each number of idle slots. This is an example for the 10ms interleaving depth. Application of compressed mode for interleaving depths other than 10ms are for further study. Each number of idle slots are classified for three cases:

Case 1 - Power measurement : Number of idle slots = 3, 4, 5, 6.

Case 2 - Acquisition of control channels : Number of idle slots = 3, 4, 5, 6, 8, 10.

Case 3 - Actual handover operation : Number of idle slots = 10, 16.

Table 7: Parameters for compressed mode

<Editors note: Smallest spreading factor used in FDD is 4, thus modification needed for the table below>

Number of idle slots	Mode	Spreading Factor	Idle length [ms]	Transmission time reduction method	Idle frame combining
3	A	512 - 256	1.63 - 1.63	Puncturing	(S)/(D)
	B	128 - 1	1.63 - 1.75		
4	A	512 - 256	2.25 - 2.25	Puncturing (D) Coding rate reduction:R=1/3->1/2 (S)	
	B	128 - 1	2.25 - 2.37		
5	A	512 - 256	2.87 - 2.87		
	B	128 - 1	2.87 – 2.99		
6	A	512 - 256	3.50 - 3.50	Puncturing (D)Spreading factor reduction by 2 (S)	
	B	128 - 2/1	3.50 - 3.62		
8	A	512 - 256	4.75 - 4.75	R=1/3->1/2(D) Spreading factor reduction by 2 (S)	
	B	128 - 2/1	4.75 - 4.87		
10	A	512 - 256	6.00 - 6.00	Coding rate reduction: R=1/3->1/2	(D)
	B	128 - 1	6.00 - 6.12		
16	A	512 - 256	9.75 - 9.75	Spreading factor reduction by 2	
	B	128 - 2	9.75 - 9.87		

(S): Single-frame method as shown in Figure 15 (1).

(D): Double-frame method as shown in Figure 15 (2).

SF="2/1": "2" is for (S) and "1" is for (D).

Annex A (informative): Blind transport format detection

A.1 Blind transport format detection using received power ratio

- This method is used for dual transport format case (the possible data rates, 0 and full rate, and only transmitting CRC for full rate).
- The rate detection is done using average received power ratio of DPDCH to DPCCH.
  Pc: Received Power per bit of DPCCH calculated from all pilot and TPC bits per slot over 10ms frame.
  Pd: Received Power per bit of DPDCH calculated from *X* bits per slot over 10ms frame.
  X: the number of DPDCH bits per slot when transport format corresponds to full rate.
  T: Threshold of average received power ratio of DPDCH to DPCCH for rate detection.

If *Pd/Pc >T* then

"TX_ON"

else

"TX_OFF"

A.2 Blind transport format detection using CRC

- This method is used for multiple transport format case (the possible data rates: 0, …, (full rate)/r, …, full rate, and always transmitting CRC for all transport formats).
- At the transmitter, the variable-rate DCH data to be transmitted is block-encoded using a cyclic redundancy check (CRC) and then convolutionally encoded. It is necessary that the CRC parity bits are mapped on the head position (or certain position) in a frame as shown in Figure A-1.
- The receiver knows only the possible transport formats (or the possible end bit position $\{n_{end}\}$ by Layer-3 negotiation (See Figure A-1). The receiver performs Viterbi-decoding on the soft decision sample sequence. The correct trellis path of the Viterbi-decoder ends at the zero state at the correct end bit position.
- Blind rate detection method by using CRC traces back the surviving trellis path ending at the zero state (hypothetical trellis path) at each possible end bit position to recover the data sequence. Each recovered data sequence is then error-detected by CRC and if there is no error, the recovered sequence is declared to be correct.
- The following variable is defined:

$$s(n_{end}) = - 10 \log ((a_0(n_{end}) - a_{min}(n_{end})) / (a_{max}(n_{end})-a_{min}(n_{end}))) \text{ [dB]} \qquad \text{(Eq. 1)}$$

  where $a_{max}(n_{end})$ and $a_{min}(n_{end})$ are, respectively, the maximum and minimum path-metric valuse among all survivors at end bit position n_{end}, and $a_0(n_{end})$ is the path-metric value at zero state.

- In order to reduce the probability of false detection (this happens if the selected path is wrong but the CRC misses the error detection), a path selection threshold D is introduced. D determines whether the hypothetical trellis path connected to the zero state should be traced back or not at each end bit position n_{end}. If the hypothetical trellis path connected to the zero state that satisfies

$$s(n_{end}) =< D \qquad \text{(Eq. 2)}$$

  is found, the path is traced back to recover the frame data, where D is the path selection threshold and a design parameter.

- If more than one end bit positions satisfying Eq. 2 are found, the end bit position which has minimum value of $s(n_{end})$ is declared to be correct.
- If no path satisfying Eq. 2 is found even after all possible end bit positions have been exhausted, the received frame data is declared to be in error.

Figure A-2 shows the procedure of blind transport format detection using CRC.

<Note: CRC moved to the end of the data block.>

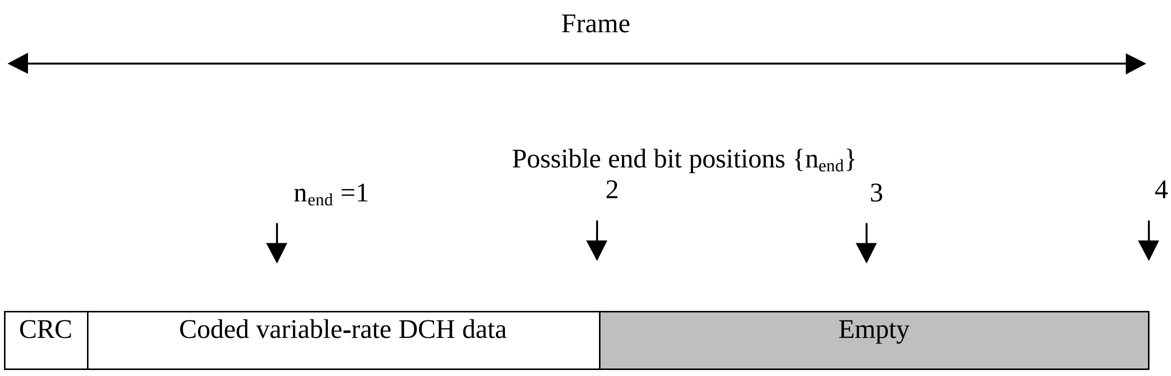

Figure A-1: An example of variable rate data format

(Number of possible transport formats = 4, transmitted end bit position n_{end} = 2)

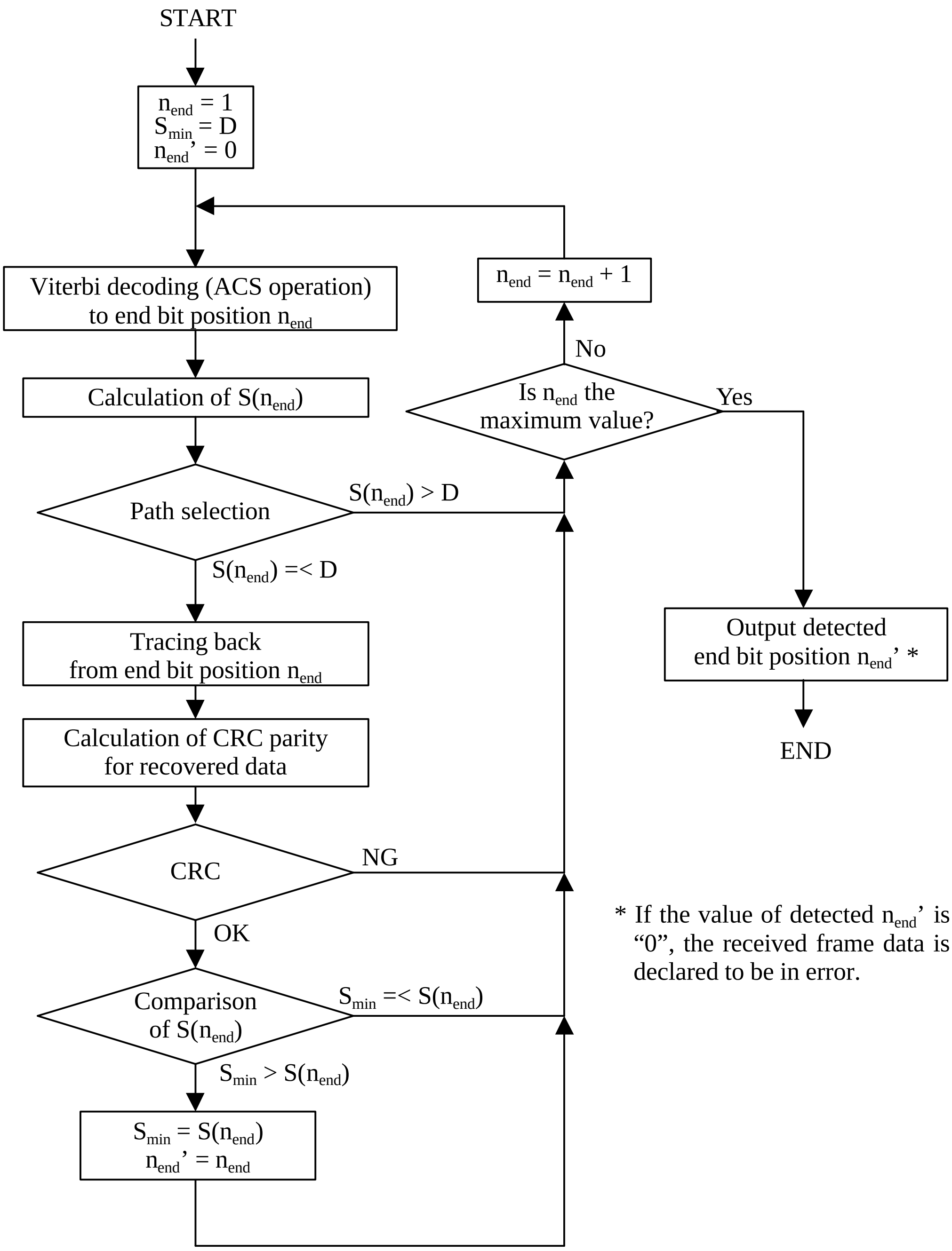

Figure A-2: Basic processing flow of blind transport format detection

Annex B: Data flow from radio frame segmentation to physical channel segmentation

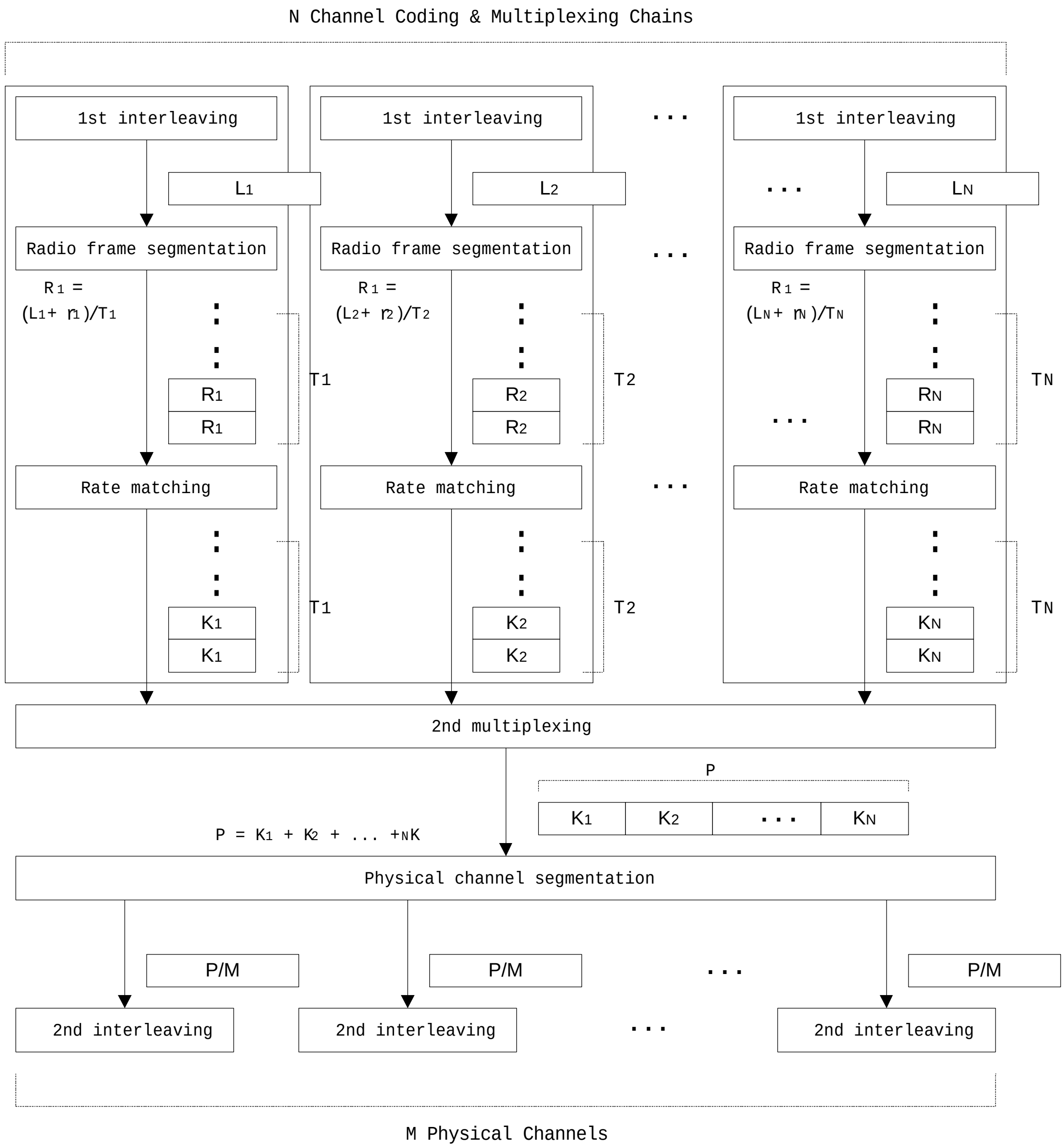

Figure B-1: Part of uplink channel coding and multiplexing chains

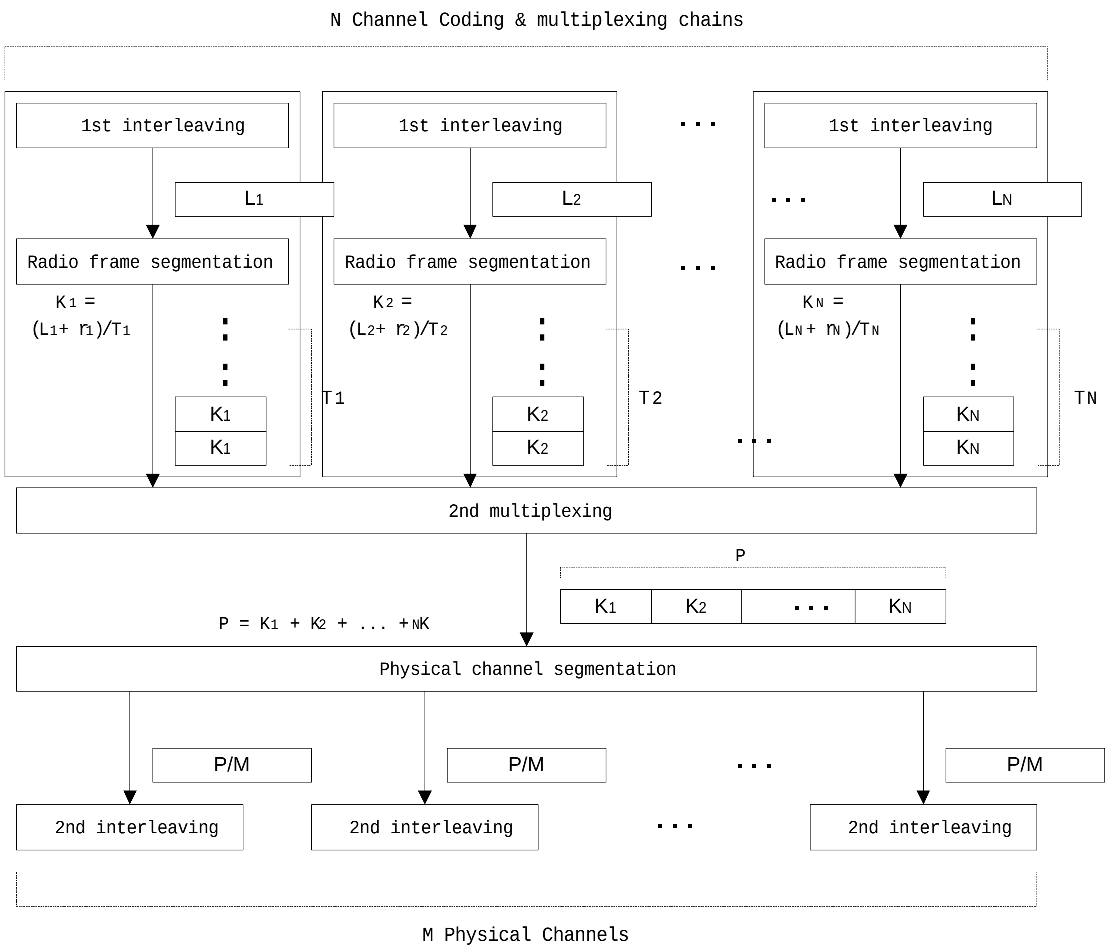

Figure B-2: Part of downlink channel coding and multiplexing chains

5 History

Document history		
V0.0.1	1999-02-12	First version created by the editor on the basis of XX.04 and the Volume 3 of the ARIB specification.
V0.1.0	1999-02-26	Version approved by WG1#2 meeting (Yokohama). The changes agreed at the meeting to incorporate e.g. Ad Hoc conclusions not yet included.
V1.0.0	1999-03-05	Version approved by RAN. Identical to V0.1.0
V1.0.1	1999-03-17	Document updated based on Ad Hoc conclusions and comments at the WG1#2 meeting (Yokohama). Editorial changes also included.
V1.0.2	1999-03-23	Document updated based on Ad Hoc conclusions at the WG1#3 meeting. Editorial changes and corrections of mistake also included.
V1.0.3	1999-03-25	Document updated based on comments at the WG1#3 meeting. Editorial changes also included.
V1.1.0	1999-03-26	Version approved by WG1#3 meeting (Nynashamn). Identical to V1.0.3
V1.1.1	1999-04-19	Document updated based on text proposals approved in WG1#4 Day 1.
V1.1.2	1999-04-20	Document updated based on text proposals approved in WG1#4 Day 2.
V2.0.0	1999-04-20	Version approved by WG1#4 meeting (Yokohama).
V2.0.1	1999-04-22	This is a results from the drafting group in RAN#3 based on the comments received.
TS 25.212 V1.0.0	1999-04-22	Noted by TSG-RAN as TS 25.212 V1.0.0
V1.0.1	1999-06-04	Document updated based on text proposals approved in WG1#5 (Cheju).
V1.1.0	1999-06-04	Version approved by WG1#5 meeting (Cheju).
V2.0.0	1999-06-22	Endorsed by TSG-RAN as version 2.0.0
V2.0.1	1999-08-17	Document updated based on text proposals approved in WG1#6 (Espoo).
V2.1.0	1999-08-31	Version approved at August 30 by WG1#7 meeting (Hannover).

Editor for TS 25.212, Multiplexing and channel coding (FDD), is:

Yoshinori Tanaka
Fujitsu Laboratories Ltd.
Email: yoshi@flab.fujitsu.co.jp

This document is written in Microsoft Word 97.