TSG-RAN Working Group 1 meeting #17
Stockholm, Sweden
November 21-24, 2000

TSGR1#17(00)1269

Agenda item:

Source: Samsung Electronics Co., Ltd.

Title: Dynamic Split Mode for TFCI

Document for: Discussion and Decision

---

1. Introduction

There are two split types for DSCH TFCI which are hard split and logical split in current specification [1].

1.1. Logical split

Logical split can support variable bit length which can be defined for both DSCH and DCH while Hard split only can support 5 bit and 5 bit length split for DSCH and DCH respectively.

When UE moves from a cell (Cell 1 in Fig. 1) to another cell (Cell 2 in Fig. 1) which is in different RNS, DSCH should be sent over Iur (PDSCH 2 in Fig.1) before SRNC is reallocated. In the current specification [2], logical split however can not support DSCH transmission over Iur since scheduling for DSCH transmission can not be signalled from Mac-c to Mac-d. Therefore when UE moves from one RNS to another RNS, logical split should not be used or SRNC reallocation should be done for DSCH handover.

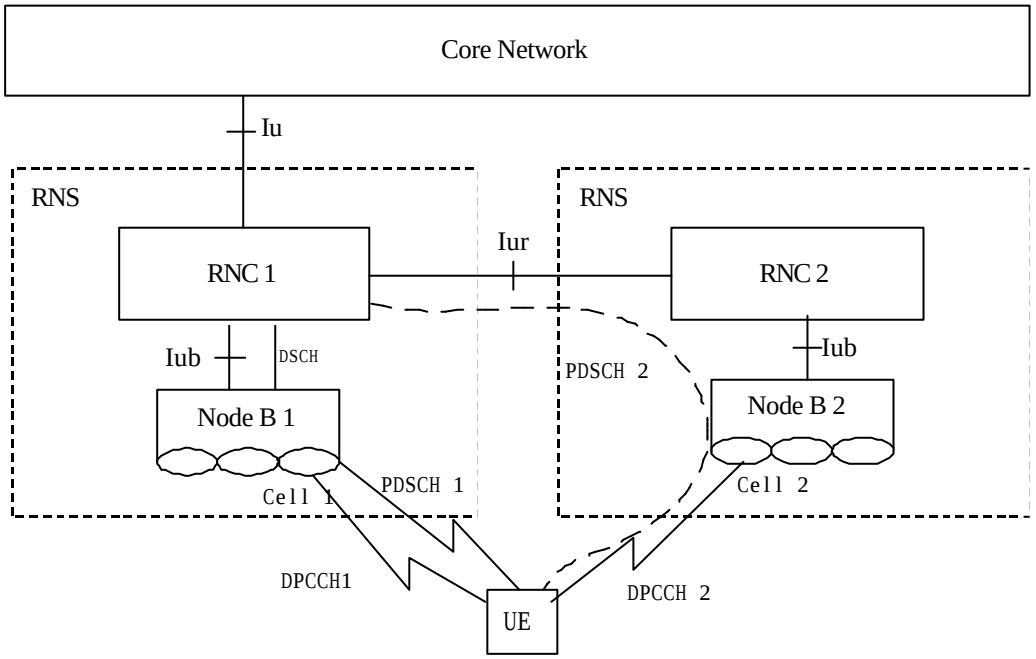

Fig 1. Hand-over with DSCH over Iur

1.2. Hard split

Hard split has advantage over logical split in the sense that hard split does not need signalling from Mac-c to Mac-d after scheduling and that hard split can be supported over Iur when DSCH need to be transmitted on Iur. Even though logical split can be supported over Iur by some signalling from Mac-c in DRNC to Mac-d in SRNC, it needs some delay which is not necessary in hard split.

Hard split can however only support that 5 bit long DSCH and DCH TFCI. Therefore only up to 32 number of transport formats can be supported for DSCH as well as DCH. That is because only (30,10) and (15,5) TFCI coding is defined in the specification [3]. When more than 32 TFCIs are needed for DCHs or DSCHs for a UE, hard split can not be used.

1.3. Summary

In summary, if

- more than 32 TFCIs are needed for DCHs or DSCHs
- UE moves from a RNS to another RNS during a connection with DSCH

then there is no way to support DSCH transmission.

But, if variable hard split mode is supported, this problem can be solved. To do this, we need a suitable TFCI coding scheme.

2. Preliminary

In this section, we present proposed coding scheme for variable hard split mode. Actually, in order to support variable hard split mode, various lengths of block code must be available. Therefore, it is the biggest problem whether we can materialise a coding scheme that satisfies the following criterions:

- Minimise H/W Complexity
- Optimise the performance of Error Correcting code

Fortunately enough, the coding scheme suiting the criterions listed above is possible, and this contribution suggests solutions to the problem.

Before describing the proposed coding scheme, we consider the coded symbol rates relative to TFCI information ratios. The most natural way is to maintain uniform code rate. The practical TFCI code rate is 1/3. Consequently, if the code rate is maintained uniform, the code lengths relative to TFCI information ratios are as shown in Table 1:

TFCI Ratio	Code Length		Code Rate
1:9	(3,1) Code	(27,9) Code	1 / 3
2:8	(6,2) Code	(24,8) Code	1 / 3
3:7	(9,3) Code	(21,7) Code	1 / 3
4:6	(12,4) Code	(18,6) Code	1 / 3

Table 1. Code Length for TFCI information ratio

Therefore, to keep the code lengths as above, we need various lengths of block codes. Hence we need a coding method that presents the maximum functions out of one structure so as to reduce impact on hardware. Actually, when creating a code out of a certain length of code the shortening method is usual in coding theory. And the coding scheme of our proposal is the shortening method itself, i.e., shortening (32,10) the Sub-code of the Second Order Reed-Muller code that is currently applied as the TFCI Coding Scheme. Consequently, by applying this method, the hardware complexity problem can be perfectly settled, and we can also expect improvement in quality.

3. Performance Issue

We suggested some motivation to a simple method in previous chapter, and in this chapter, we present the theoretic comparison between the optimal code of general block codes and that of the method described above. In coding theory, there is an almost accurate measurement in designing linear block code, which is the minimum distance of the code. In Table 2,

the minimum distance of the optimal code using the shortening method, d_{min} is listed relative to the code lengths introduced in the previous chapter in comparison with optimal bounds of minimum distance in general linear block codes.

Code Length	Optimal Bound	d_{min}
(3,1)	3	3
(6,2)	4	4
(9,3)	4	4
(12,4)	6	6
(18,6)	8	7
(21,7)	8	8
(24,8)	8	8
(27,9)	10	9

Table 2. Performance of proposed coding scheme

4. Encoding Scheme

In this chapter, the encoding scheme concerning the proposed scheme is presented in concrete terms. As mentioned in the previous chapter, the coding scheme of our proposal is shortening (32,10) the Sub-code of the Second Order Reed-Muller code, the existing TFCI Coding Scheme. That is to say, though (32,10) the sub-code of the second order Reed-Muller code is the linear combination of 10 basis code words, in order to use (27,9) code, e.g., we can choose 9 basis code words and encode, and then puncture 5 coded symbols out of 32 code lengths. Table 3 shows basis code words and puncturing pattern that are used in materialisation by the shortening method.

Code Length	Puncturing Pattern	Used basis
(3,1)	1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	M_0
(6,2)	3, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	M_0, M_1
(9,3)	7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	M_0, M_1, M_2
(12,4)	0, 1, 2, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	M_0, M_1, M_2, M_3, M_4
(18,6)	0, 7, 9, 11, 16, 19, 24, 25, 26, 27, 28, 29, 30, 31	$M_0, M_1, M_2, M_3, M_4, M_5$
(21,7)	0, 2, 6, 7, 9,10,12,14,15,29,30	$M_0, M_1, M_2, M_3, M_4, M_6, M_7$
(24,8)	1, 7, 13, 15, 20, 25, 30, 31	$M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7$
(27,9)	0, 2, 8,19,20	$M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8$

where M0 = 10101010101010110101010101010100

M1 = 01100110011001101100110011001100

M2 = 00011110000111100011110000111100

M3 = 00000001111111100000001111111100

M4 = 00000000000000011111111111111101

M5 = 11111111111111111111111111111111

M6 = 01010000110001111100000111011101

M7 = 00000011100110111011011100011100

M8 = 00010101111100100110110010101100

Table 3. Puncturing pattern and used basis

The encoder structure of the proposed coding scheme can be materialised as shown in Fig 2.

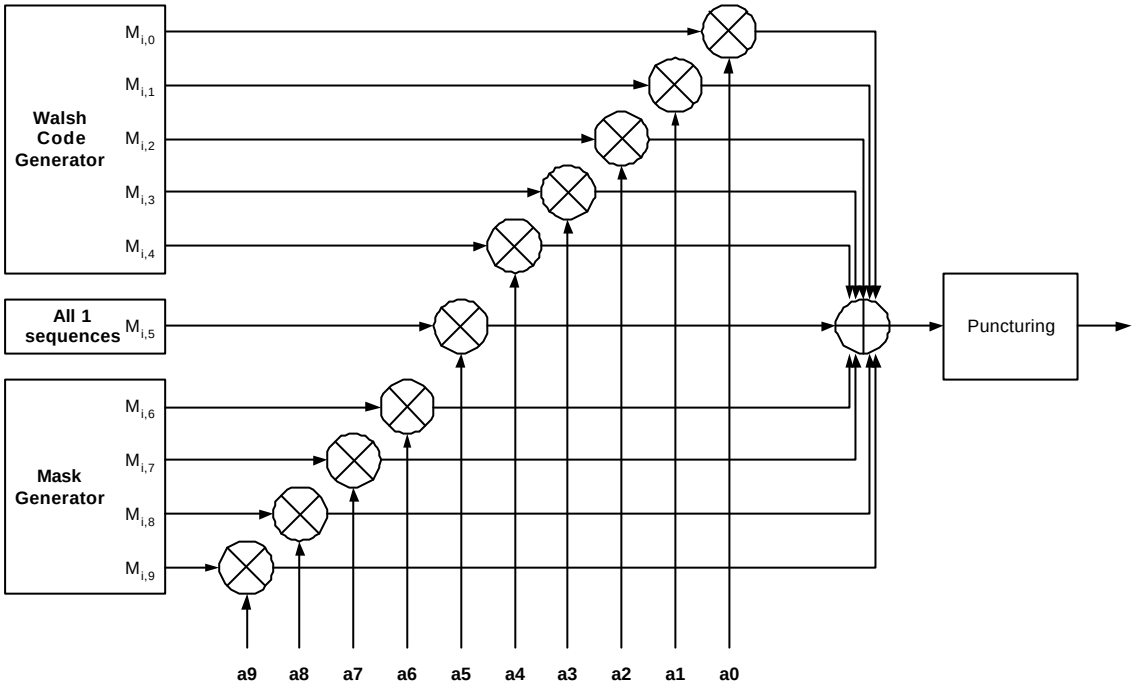

Fig 2. Encoder Structure

As shown in Fig 2, the only difference from the encoder structure of the existing TFCI coding scheme is the supplement of puncturing pattern. Therefore, by supplementing puncturing pattern in existing TFCI coding scheme without any impact on hardware, we can support hard split mode, and what's more, the quality can remain optimal.

5. Proposal

It is proposed to add TFCI coding for (3*n, n) where n = 1,2,3,4,6,7,8,9 for supporting variable split of DSCH.

Hard split does not need signalling from Mac-c to Mac-d and with the proposed coding scheme variable split can be supported for hard split and therefor DSCH can be enhanced with the hard split.

Reference

[1] 3G TS 25.331 RRC Protocol Specification

[2] 3G TS 25.303 Interlayer Procedures in Connected Mode

[3] 3G TS 25.212 Multiplexing ad channel coding(FDD)

Contact Points

Yongjun Kwak	evatt@samsung.com	+82 31 779 6626
Hyeonwoo Lee	woojaa@samsung.com	+82 31 779 6613
Jaeyoel Kim	kimjy@samsung.com	+82 31 779 6885
Sungho Choi	schoi@samsung.com	+82 31 779 6624