TSG-RAN WG-1
Meeting # 6
Espoo, Finland
July 13-16, 1999

TSG-R1#6(99)967

Title: Optimum Rate Matching of Turbo/Convolutional Coding for 3GPP Up/Down Links

Source: Nortel Networks[1]

0.0 Summary

In this contribution, we discuss a unified optimum scheme for Turbo/Convolutional Coding in up/down links. Nortel investigated some optimal scheme for puncturing Turbo codes and convolutional codes. It turns out that the scheme we investigated is very similar to what is proposed in [6] and leads to similar data flows and puncturing positions. However, only in the case where the number of systematic bits is a multiple of 8, the scheme in 6 can avoid puncturing of Turbo codes systematic bits in a straightforward way, while the framework proposed here can ensure that for all number of systematic bits in a simple way.

Therefore we recommend to use the multiplex bit stream proposed in [6] by applying the rate matching and 1st interleaving procedure presented in this contribution.

1 Contact Persons: Wen Tong, Catherine Gauthier, Nortel Networks
1 Place des Freres Montgolfier, 78042 Guyancourt, France
Tel:+33 1 39 44 57 47
Fax:+33 1 39 44 50 12
e_mail: wentong@nortelnetworks.com / gauth@nortelnetworks.com

0.1 Introduction

Based on the working assumption of rate matching algorithm [1] proposed by Siemens, Fujitsu [3] and LGIC [4] modified such an algorithm to the puncturing of 8-PCCC Turbo coding. This leads to different rate matching scheme for different codes. Furthermore such rate matching schemes are designed for downlink. Nortel [2] proposed a multiplex solution for up link puncturing of 8-PCCC Turbo coding. In this contribution, based on our finding of optimum puncturing of 8-PCCC and convolutional code defined in 3GPP in an companion contribution [5], we propose a unified scheme to combine the optimum rate matching into a single framework, *without* modification of current working assumptions on rate matching algorithm and 1st channel interleaver for both up down links. It is shown that the proposed scheme can provide better performance than the arrangement in [1], [2],[3]. (detailed investigation see [5]). It is shown the scheme is very similar to what is proposed in [6].

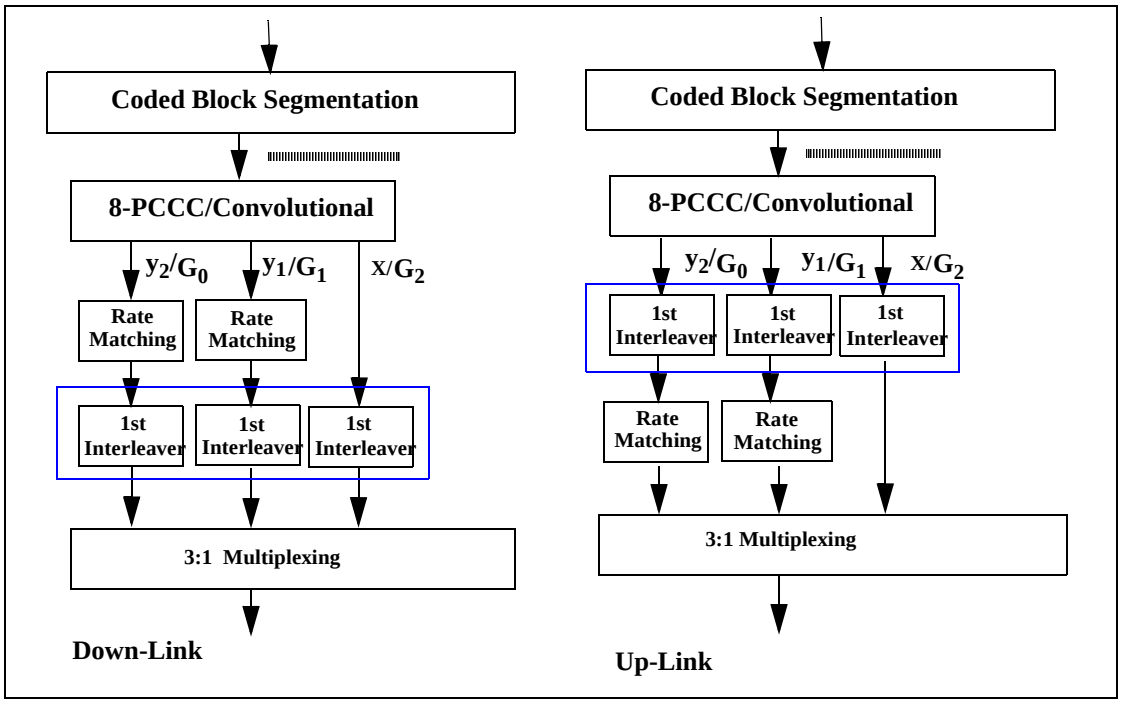
In a first part, we present this optimal unified scheme. In a second part, we show that it is very similar to the scheme presented in Ref [6], while being simpler to ensure that the systematic bits are not punctured in the case where the number of systematic bits is not a multiple of 8.

0.2 Unified Rate Matching Scheme

The unified Rate Matching scheme is shown in Figure 1. The principle is based on the rate matching characteristic of 8-PCCC code, i.e. the systematic bits which should not be punctured, and on the other hand, the two parity bit streams should be punctured evenly. In light of this rule, we found also for the R=1/3 convolutional code, the contribution of the three polynomials to the code distance in un-equal to the free distance. Simulation results have verified this observation, therefore we can unify the rate matching scheme for convolutional code and 8-PCCC into a single framework. See Figure 1.

In order to *not* modify the existing working assumptions on rate matching algorithm and 1st channel interleaver, radio frame segmentation and 2nd multiplexing, we apply the 1st interleaver and rate matching algorithm to each parity bit stream for the up-link. Such an arrangement casts zero complexity increase. For the down link, it is the same arrangement for the up-link, except that for the rate matching of parity bit#2 for 8-PCCC case, additional Turbo interleaving information is used. (see [5]).

FIGURE 1. Details for Unified Rate Matching Scheme for Up/Down Link



0.3 Commonality and Difference with [6]

In [6], it is proposed to use the essential the same scheme as [2], let's examine two examples in up link as follows: (considering tail bits are appended at end of systematic bits)

Example-1: For R=1/3 Turbo code. If the systematic bits is 16, Table 1 shows the input to the 1st interleaver, Table 2 shows the output of 1st interleaver by using [6]. Table 3 shows the output matrix of 1st interleaver based in Figure 1. We can see the scheme proposed in [6] and figure 1 are identical.

TABLE 1. Input Matrix

$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_1(4)$	$y_1(5)$	$y_1(6)$	$y_1(7)$	$y_1(8)$
$y_1(9)$	$y_1(10)$	$y_1(11)$	$y_1(12)$	$y_1(13)$	$y_1(14)$	$y_1(15)$	$y_1(16)$
$y_2(1)$	$y_2(2)$	$y_2(3)$	$y_2(4)$	$y_2(5)$	$y_2(6)$	$y_2(7)$	$y_2(8)$
$y_2(9)$	$y_2(10)$	$y_2(11)$	$y_2(12)$	$y_2(13)$	$y_2(13)$	$y_2(15)$	$y_2(16)$
x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)
x(9)	x(10)	x(11)	x(12)	x(13)	x(14)	x(15)	x(6)

TABLE 2. Output Matrix of 1st Interleaver [6]

$y_1(1)$	$y_1(5)$	$y_1(3)$	$y_1(7)$	$y_1(2)$	$y_1(4)$	$y_1(6)$	$y_1(8)$
$y_1(9)$	$y_1(13)$	$y_1(11)$	$y_1(15)$	$y_1(10)$	$y_1(12)$	$y_1(14)$	$y_1(16)$
$y_2(1)$	$y_2(5)$	$y_2(3)$	$y_2(7)$	$y_2(2)$	$y_2(4)$	$y_2(6)$	$y_2(8)$
$y_2(9)$	$y_2(13)$	$y_2(11)$	$y_2(15)$	$y_2(10)$	$y_2(12)$	$y_2(13)$	$y_2(16)$
x(1)	x(5)	x(3)	x(7)	x(2)	x(4)	x(6)	x(8)
x(9)	x(13)	x(11)	x(15)	x(10)	x(12)	x(14)	x(6)

TABLE 3. Output Matrix of 1st Interleaver [Figure 1]

$y_1(1)$	$y_1(5)$	$y_1(3)$	$y_1(7)$	$y_1(2)$	$y_1(4)$	$y_1(6)$	$y_1(8)$
$y_1(9)$	$y_1(13)$	$y_1(11)$	$y_1(15)$	$y_1(10)$	$y_1(12)$	$y_1(14)$	$y_1(16)$
$y_2(1)$	$y_2(5)$	$y_2(3)$	$y_2(7)$	$y_2(2)$	$y_2(4)$	$y_2(6)$	$y_2(8)$
$y_2(9)$	$y_2(13)$	$y_2(11)$	$y_2(15)$	$y_2(10)$	$y_2(12)$	$y_2(13)$	$y_2(16)$
x(1)	x(5)	x(3)	x(7)	x(2)	x(4)	x(6)	x(8)
x(9)	x(13)	x(11)	x(15)	x(10)	x(12)	x(14)	x(6)

Example-2: For R=1/3 Turbo code. If the systematic bits is 13. Table 4 shows the input matrix by using [6], Table 5 shows the associated output matrix. Table 6 shows the input of scheme in Figure 1, and Table 7 shows its associated output matrix.

TABLE 4. Input Matrix

$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_1(4)$	$y_1(5)$	$y_1(6)$	$y_1(7)$	$y_1(8)$
$y_1(9)$	$y_1(10)$	$y_1(11)$	$y_1(12)$	$y_1(13)$	$y_2(1)$	$y_2(2)$	$y_2(3)$
$y_2(4)$	$y_2(5)$	$y_2(6)$	$y_2(7)$	$y_2(8)$	$y_2(9)$	$y_2(10)$	$y_2(11)$
$y_2(12)$	$y_2(13)$	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)
x(7)	x(8)	x(9)	x(10)	x(11)	x(12)	x(13)	

TABLE 5. Output Matrix of 1st Interleaver [6]

$y_1(1)$	$y_1(5)$	$y_1(3)$	$y_1(4)$	$y_1(2)$	$y_1(6)$	$y_1(4)$	$y_1(8)$
$y_1(9)$	$y_1(13)$	$y_1(11)$	$y_1(12)$	$y_1(10)$	$y_2(1)$	$y_1(12)$	$y_2(3)$
$y_2(4)$	$y_2(8)$	$y_2(6)$	$y_2(7)$	$y_2(5)$	$y_2(9)$	$y_2(7)$	$y_2(11)$
$y_2(12)$	x(3)	x(1)	x(2)	$y_2(13)$	x(4)	x(2)	x(6)
x(7)	x(11)	x(9)	x(10)	x(8)	x(12)	x(10)	

TABLE 6. Input Matrix [Figure 1]

$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_1(4)$	$y_1(5)$	$y_1(6)$	$y_1(7)$	$y_1(8)$
$y_1(9)$	$y_1(10)$	$y_1(11)$	$y_1(12)$	$y_1(13)$			
$y_2(1)$	$y_2(2)$	$y_2(3)$	$y_2(4)$	$y_2(5)$	$y_2(6)$	$y_2(7)$	$y_2(8)$
$y_2(9)$	$y_2(10)$	$y_2(11)$	$y_2(12)$	$y_2(13)$			
x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)
x(9)	x(10)	x(11)	x(12)	x(13)			

TABLE 7. Output Matrix of 1st Interleaver [Figure 1]

$y_1(1)$	$y_1(5)$	$y_1(3)$	$y_1(7)$	$y_1(2)$	$y_1(6)$	$y_1(4)$	$y_1(8)$
$y_1(9)$	$y_1(13)$	$y_1(11)$		$y_1(10)$		$y_1(12)$	
$y_2(1)$	$y_2(5)$	$y_2(3)$	$y_2(7)$	$y_2(2)$	$y_2(6)$	$y_2(4)$	$y_2(8)$
$y_2(9)$	$y_2(13)$	$y_2(11)$		$y_2(10)$		$y_2(12)$	
x(1)	x(5)	x(3)	x(7)	x(2)	x(6)	x(4)	x(8)
x(9)	x(13)	x(11)		x(10)		x(12)	

As we can see, in the case when the number of systematic bits is a multiple of 8 bits, the two unified schemes are identical, while for the other numbers of systematic bits, it is more complicated with [6] scheme to control that no systematic bits is punctured. This is due to the fact that the starting position of systematic bits for the segmented radio frame are not the same. For the scheme described in Figure 1, we can easily apply rate matching to one of the parity bit stream. The second parity bit stream puncturing does not need to be re-computed.

And one feature for the scheme in Figure 1 is that there is zeros increase of complexity for the 1st interleaver, since we need only compute the address for the 1st parity bits stream, the 2nd parity bits and systematic bits stream interleaving does not need to re-compute the interleaving address.

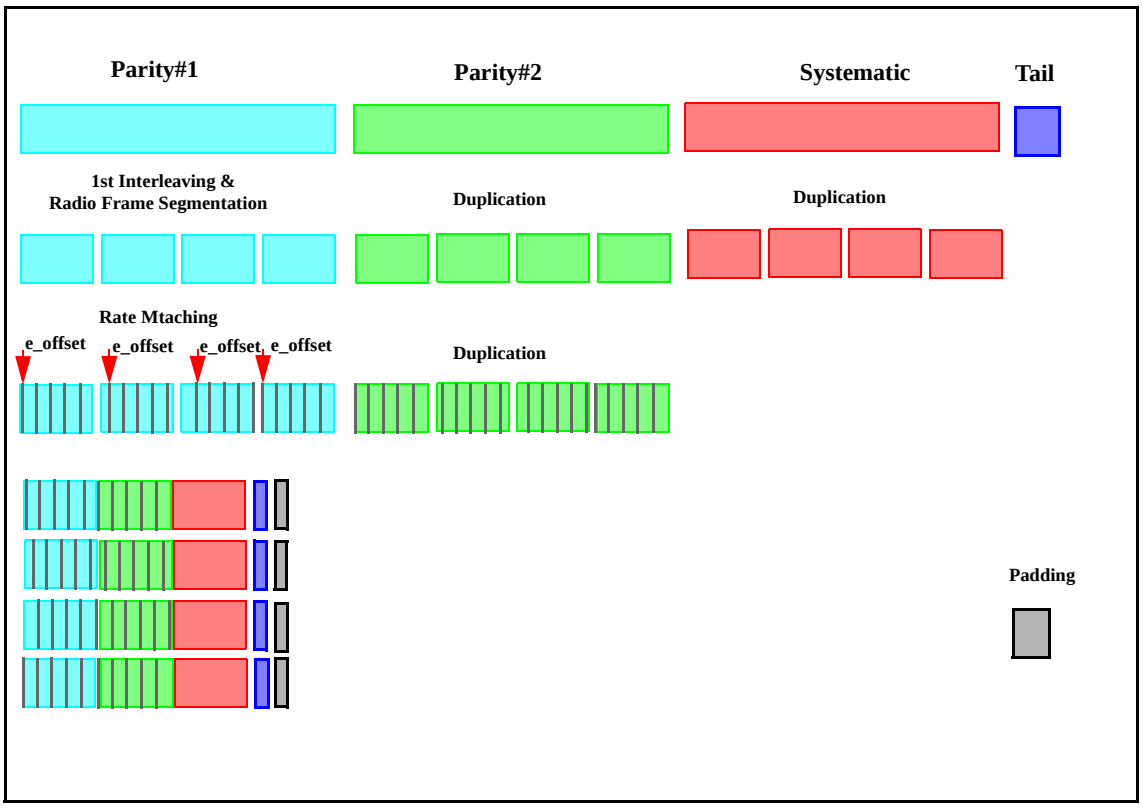
0.4 Recommendations

We recommend the 3 bits flow separation as proposed in Ref [2]. One of the detailed implementation can be done by multiplexing bit stream of the systematic, parity bit#1, parity bit #2 as proposed in [6].

$$y_1(1), y_1(2)\ldots y_1(N), y_2(1), y_2(2)\ldots y_2(N), x(1), x(2)\ldots x(N), t_a(1), \ldots t_a(12)$$

Instead to apply the entire stream to the 1st interleaver, and to perform rate match on the combined parity bits part, we recommend to apply 1st interleaving to systematic, parity#1 and parity#2 separately, and apply rate matching to parity#1 and parity#2 only. see Figure 2. Such an arrangement has zero complexity increase, enjoy the same interleaving performance as in [6] and with the advantage of additional algorithm to determine the systematic bit position when puncturing is performed.

FIGURE 2. Example of Proposed Unified Rate Matching Flow (40ms Case Option-1)



Note that the multiplexer in Figure 1 remains to be optimized. It can be implemented by bit-wise multiplexing and block-wise multiplexing, or some other rule of multiplexing. This remains to be F.F.S.

0.5 Reference

[1] 3GPP TSG RAN WG1 Multiplexing and Channel Coding (FDD) TS 25.212. v2.0.0
[2] Nortel Networks: “Proposal for Rate Matching for Turbo Codes” TSGR1#4 (99) 467
[3] Fujitsu,:” Optimized Puncturing Scheme for Turbo Coding” TSGR1#4(99)381

[4] LGIC:” Comparison of Downlink Puncturing Algorithns” TSGR1 #5 (99) 654

[5] Nortel Networks:” Rate Matching Puncturing of 8-PCCC and Convolutional Code” TSGR1 #6 (99) 950

[6] Fujitsu and Siemens: “ Universal Rate Matching for Up/Down Link Turbo/Convolutional Coding” TSGR1#6 (99)