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

**TSG-R1#6(99)A56**

# Title: Analysis of Commonalities of Turbo Code Puncturing Rate Matching Proposals

## Source: Nortel Networks[1]

## 0.0 Summary

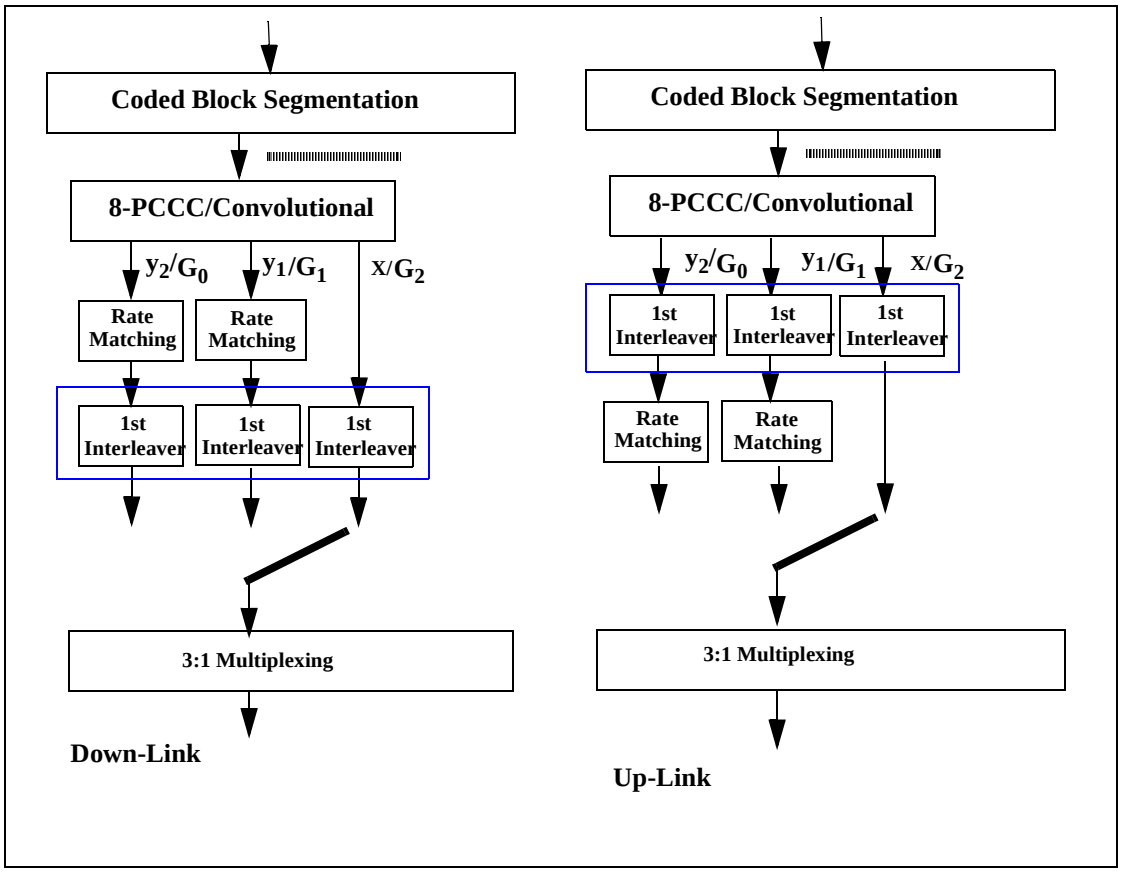
In this contribution, we discuss the proposals for rate matching puncturing in both down/ up links for Turbo codes. The key commonality is to exclude the systematic bit stream from 8-PCCC Turbo encoder output from rate matching puncturing. The major difference are twofold: (1) apply same or difference puncturing pattern to the 2 parity-bit streams (2) How and where to multiplex and de-multiplex the 2 parity bit streams.

**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

# 1.0 Two/Three Bit Streams and Associated Multiplex

In Figure 1, a generic rate matching multiplexing scheme for Turbo code puncturing is shown for down-link and up-link.

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




As indicated in Figure 1, several possible ways can be used to achieve Turbo code puncturing rate matching rules presented in Ref [2]. The separated treatment of the 1st interleaving for all the three data flow is a simple and straightforward approach, which essentially costs no complexity increase, since the interleaving address and rate matching position are only required to compute, and is can be applied to other two data flows. In this case, both 1st interleaving and rate matching computing is reduce to 30% and 50% respectively.

In the case of using a single 1st interleaving, for Ref[8], additional rules are imposed to the rate matching algorithm in order to separate systematic bit stream from puncturing, for Ref[9] the 1st interleaver structure and bit multiplexing scheme is design to separate systematic bits stream from puncturing. While the fundamental difference between Ref[8][9] and Ref[7] is the bit-wise multiplex or block-wise multiplex is employed for up-link as shown in Figure 1.

# 2.0 Scheme -I (Ref[7])

**TABLE 1. Input Matrix to 1st Interleaver [Block MUX]**

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

**TABLE 2. Output Matrix of 1st Interleaver [Block MUX]**

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

**TABLE 3. Rate Matching on Parity Bit #1 and 2 Block [Block MUX]**

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

**TABLE 4. Radio Frame Output [Block DeMUX]**

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

# 3.0 Scheme-II (Ref [9][8])

**TABLE 5. Output Matrix of 1st Interleaver [Bit MUX]**

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

**TABLE 6. Output Matrix of 1st Interleaver [Bit MUX]**

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

**TABLE 7. 3:1 Bit DeMUX and Rate Matching**

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

**TABLE 8. BitMUX and Radio Frame Segmentation**

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

Note that in Ref [8] instead of using bit demux in Table 7, a algorithm is designed to separate the three bit streams.

# 4.0 Comparison of Multiplex Flow

As we can see from Figure 2 & 3, the key difference for Scheme-I and Scheme-II is that the Scheme-II can preserve shuffled parities and systematic bits stream at the input to the 2nd multiplexing block.

**FIGURE 2. (Ref [7]) Example of Unified Rate Matching Flow (40ms Case)**

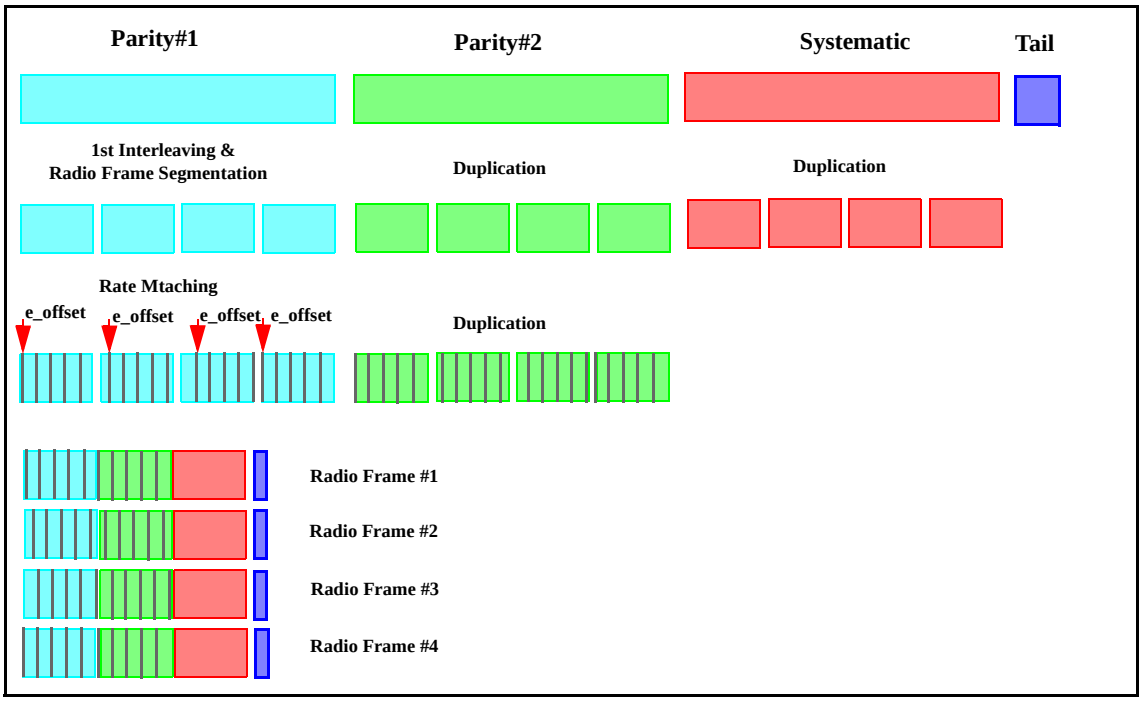
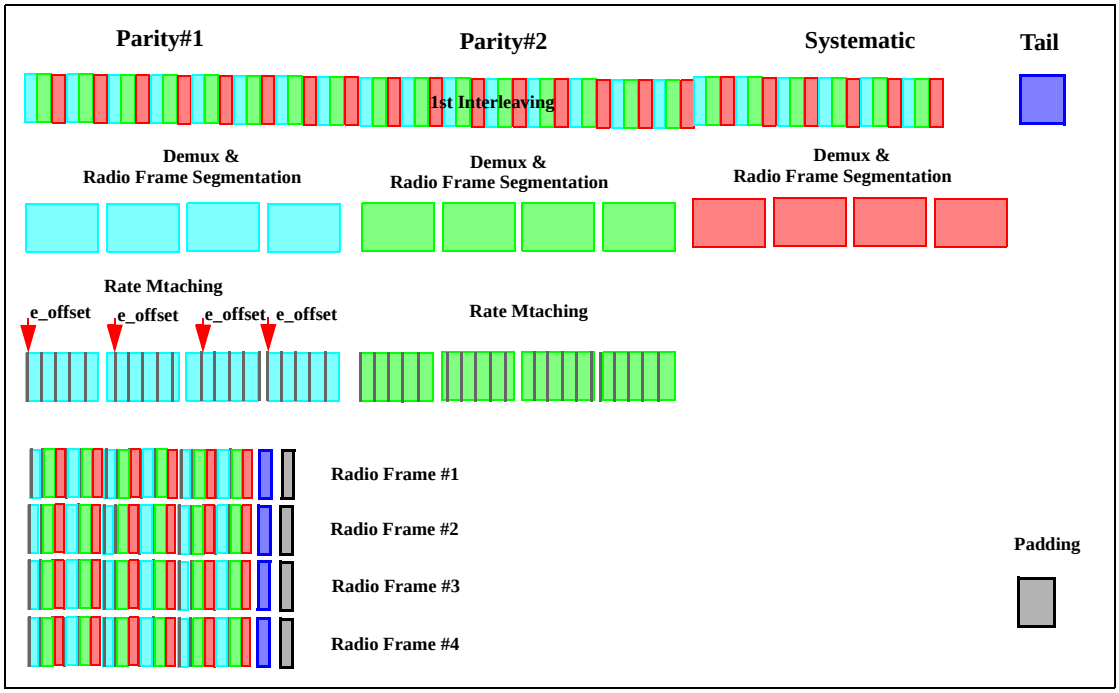



**FIGURE 3. (Ref [8],[9]) Example of Unified Rate Matching Flow (40ms Case)**



## 4.1 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 Algorithms” TSGR1 #5 (99) 654

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

[6] Nortel Networks:” Optimum Rate Matching of Turbo/Convolutional Coding for 3GPP Up/Down Links” TSGR1 #6 (99) 967

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

[8] LGIC: “Alternative Method of Turbo Code Puncturing of Up-Link” TSGR1#6(99)909

[9] Samsung:” Unified Rate Matching for Turbo Convolutional Codes and Up/Down Links” TSGR1#6 (99)919