3GPP TSG RAN WG1#14 Tdoc R1-00-0854
July 4 - 7, 2000, Oulu, Finland

Agenda Item:	AH27, Radio Link Performance Enhancements
Source:	Siemens AG
Title:	Improved Rate Matching Scheme for Convolutional Codes (Revision of R1-00-649)
Document for:	Discussion

1. Introduction

Up to now it was pointed out in different papers that there is some potential to enhance the current channel coding scheme especially for short code blocks with nearly no additional expenditure [1][5]. Unfortunately there was not enough time to finish the discussion and include such a scheme in release 1999. The aim of this proposal now is to re-start work again and present a generally applicable method for convolutionally coded blocks which can be easily implemented within rate matching. The additional complexity of the proposed method is low. Retaining the simulation conditions as described in [1] we investigate the impact for selected realistic transport format combinations. For AMR 4.75 kbps and AMR 7.40 kbps our results show an improvement between 0.10 and 0.17 dB.

2. Principle

As mentioned in [2] there is an imbalance in the protection of information bits after channel coding, because of the pre-known start and ending state of the convolutional coder at the beginning and the end of a code block. The characteristic of that imbalance is that the bits near the termination have a lower BER than the bits in between. This effect means that bits transmitted near the beginning or the end of a code block are protected better against errors than others. But for most applications the bits should be transmitted nearly equally protected (except of course if they are transmitted via different transport channels). To avoid this waste of energy, to achieve a better overall performance and to keep the additional complexity increase low, we propose an improved method by integrating end puncturing into the normal rate matching scheme. This is realised by extending the rate matching pattern generation algorithm as currently described in TS 25.212 and TS 25.222 and allows not only uniform puncturing or repetition, but also heavier puncturing at the end of the coded bits and less heavy puncturing in the middle. For the case of repetition this would mean light repetition at the end but more repetition in the middle. If the number of bits after rate matching is very close to the number of input bits, it may even be that some puncturing is done at the ends compensated by some repetition in the middle.

3. Description of the Algorithm

In the present formulation of the algorithm the amount of puncturing/repetition is basically determined by the relation of the parameters e_{plus} and e_{minus}. Both the parameter e_{plus} and e_{minus} are fixed for all bits of a transport channel. If e_{minus} is allowed to be varied and takes different values for the bits at the end of a block as compared to the bits in the middle of the block, then different local repetition/puncturing rates can be achieved in the different areas. The repetition/puncturing patterns within such areas are still evenly distributed. Even at the border of those areas there is generally no repetition/puncturing of adjacent or very close bits and also no extremely long gap. However, with the current algorithm this simple modification does not allow puncturing and repetition in the same frame, because puncturing and repetition is executed in two different algorithms. But with some modification even this is possible. We can avoid having two separate algorithms for repetition and puncturing as previously described in [3].

For the integration of end puncturing into rate matching, which was already discussed for release 1999 [2], we use the weights w_m (m=1..X_i) for fine tuning the rate matching pattern. For the simulations described within this document we used a weight pattern defined by:

w_m={1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,4,4,4,4,...,4,4,4,4,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1}

We have selected w_m to be constant in blocks of 8 bits in order to ease the implementation in particular if rate matching is to be applied after first interleaving. If w_m is set to 1, evenly distributed rate matching similar to the puncturing/repetion rule applied for release 1999 results. W is the sum of weights. In this formulation the meaning of e_{minus} is changed slightly. It is the sum of e_{minus} (but taking the absolute value over the given formulas for calculation not into consideration) used by the current specification and e_{plus}. To generate the advanced rate matching patterns we propose the following algorithm:

```
$e = e_{ini}$                              -- initial error between current and desired puncturing ratio
$m = 1$                                    -- index of current bit
$e_{plus} = e_{plus} - a*(X_i - W)$
do while $m <= X_i$
    $e = e – w_m*e_{minus}$                -- update error
    do while $e <= 0$
        select bit $x_{i,m}$ for transmission
        $e = e + e_{plus}$                 -- update error
    end do
    $m = m + 1$                            --  next bit
end do
```

4. Simulation Results for AMR

In this chapter simulations about the performance of the AMR 4.75 kbps codec mode for AWGN and BER curves for the 7.40 kbps mode are shown. The results with the current multiplexing and channel coding scheme are compared with the new proposal. The main difference between the two methods is the use of the modified rate matching algorithm including weights for fine tuning the resulting rate matching pattern. In the following the simulation assumptions are summarised:

- AWGN channel type
- Ideal phase estimation
- Convolutional Coding
- No overhead for pilot or PC bits etc. considered

For the mapping onto a physical channel the following considerations are made. The AMR 4.75 kbps speech codec delivers 42 Class A and 53 Class B bits and the 7.40 kbps codec mode produces 61 Class A bits and 87 Class B bits per 20 ms. Both modes fit into a SF 256 slot structure. We used as an example DL slot format 2 with 2 pilot symbols, rate 1/3 coding and no TFCI. In the 4.75 kbps case we used three transport channels: one for Class A, one for Class B and one for a 2 kbps information rate dedicated signalling channel. The physical channel partitioning, i.e. the semistatic rate matching attributes were chosen such that Class A target BER is 0.01% and Class B target BER is 0.03% [4]. Table 1 shows the bit allocation for the AWGN simulation. As the bits of Class B have a higher target BER they are protected with a higher effective code rate needing a lower number of gross bits.

	Class A	Class B
Speech Bits delivered per 20 ms	42 + 8 (CRC)	53
Addition of Tail Bits and Convolutional Coding R=1/3	174	183
Semistatic Rate Matching Parameter	1.0	0.94
Rate Matching	174 → 158	183 → 156

	Dedicated Control Channel
Bits delivered for channel coding per 40 ms	112
Addition of Tail Bits and Convolutional Coding R=1/3	360
Semistatic Rate Matching Parameter	1.0
Rate Matching	360 → 332

Table 1: *Bit allocation for the AMR 4.75 kbps AWGN simulation*

The following results from the AMR 4.75 kbps simulations. As can be seen we achieve a gain of about 0.17 dB compared to the current scheme.

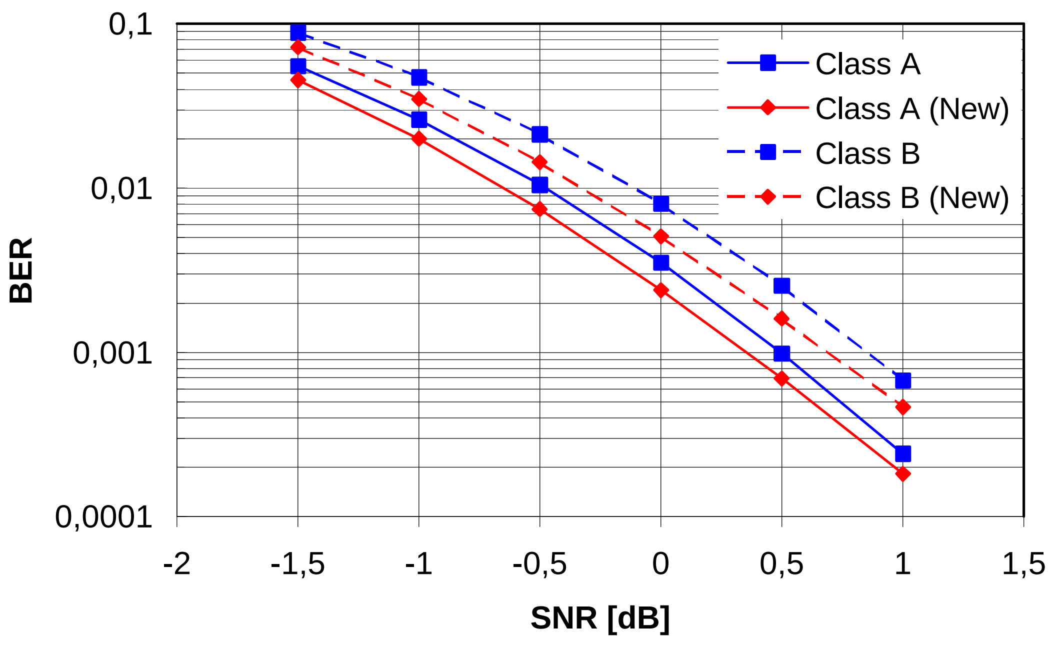

Figure 1: *Performance of normal puncturing and improved puncturing for AMR 4.75 kbps (AWGN)*

Table 2 shows the bit allocation for the AMR 7.40 kbps AWGN simulation.

	Class A	Class B
Speech Bits delivered per 20 ms	61 + 8 (CRC)	87
Addition of Tail Bits and Convolutional Coding R=1/3	231	183
Semistatic Ratematching Parameter	1.0	0.94
Rate Matching	231 → 222	285 → 258

Table 2: *Bit allocation for the AMR 7.40 kbps simulation*

Figure 2 shows the performance for AMR 7.40 kbps. The improvement is smaller because of the larger blocksizes, but it is still substantial (~ 0.1 dB).

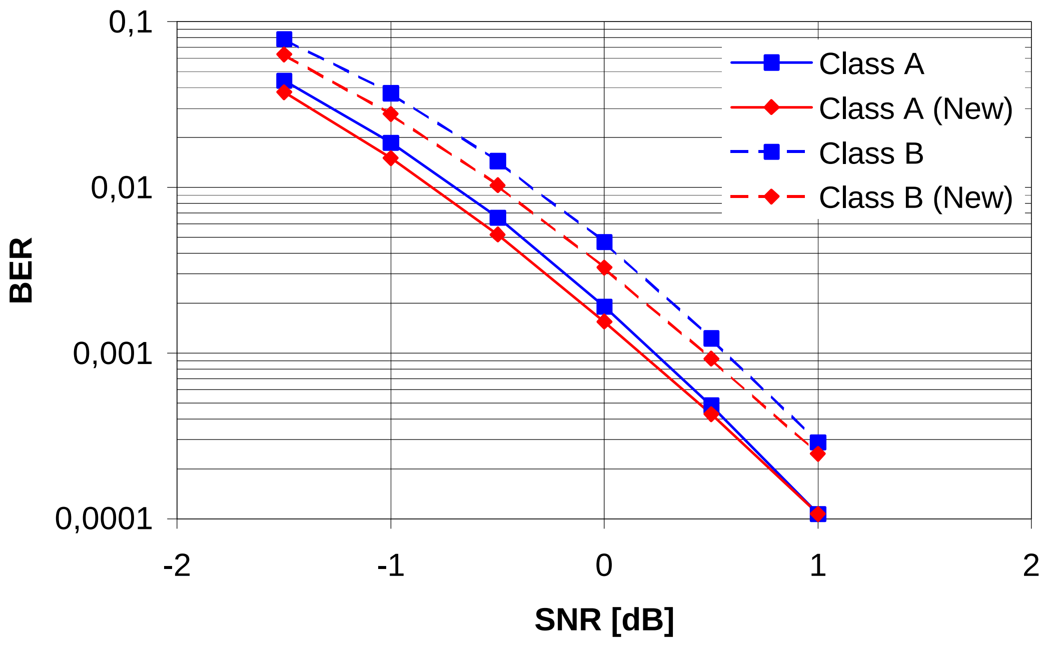

Figure 2: Performance of normal puncturing and improved puncturing for AMR 7.40 kbps (AWGN)

5. Simulation for various Puncturing and Repetition Rates

To be on the save side we also compared the performance of the proposed scheme and release 1999 results for various puncturing and repetition rates. In general the simulation conditions are identical to the simulations described in chapter 4 (AWGN channel, ideal phase estimation, convolutional coding and no overhead for e.g. pilot or PC bits considered). The input blocksize for the rate matching unit was chosen to be 183 bits according to the AMR simulations shown above. Depending on a puncturing or repetition rate of –15% up to 15% the number of output bits per block lies between 156 and 210 bits. Figure 3 shows the BER at 5 dB (Eb/No) versus the puncturing/repetition rate. It can be seen, that the improved rate matching gives a lower BER for all cases except the 0% case, where the performance is the same for both schemes.

Further more it can be seen that the performance (BER vs. Eb/No) of the improved scheme depends much less on the puncturing/repetition rate than for the release 1999 scheme. The formula in section 4.2.7 in 25.212 for calculating the required rate matching N_{ij} for each Transport channel based on RM_i, the semi-static rate matching attribute for the transport channels implicitly assumes that the performance is independent from the rate matching. As this assumption is better fulfilled for the improved puncturing scheme, the QoS balancing between different transport channels will be maintained more accurately.

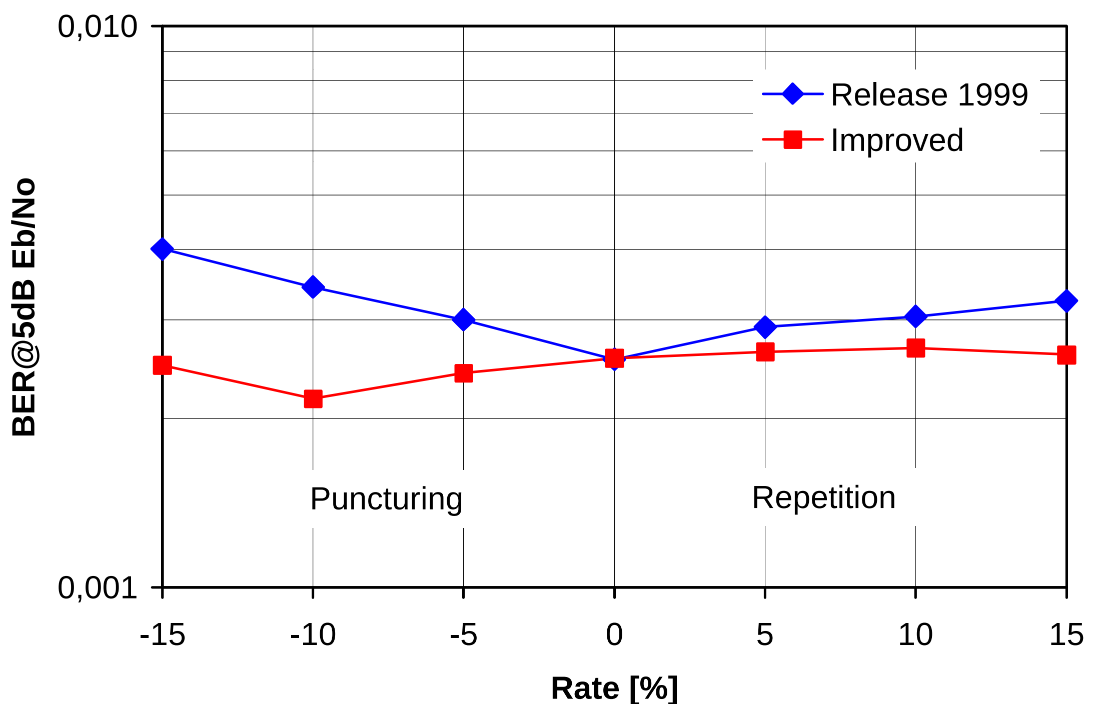

Figure 3: *Performance of rate matching for various puncturing/repetition rates*

6. Conclusion

The simulations show a performance enhancement compared to the evenly distributed rate matching approach of release 1999. For the AMR 4.75 kbps mode with block length of 174/183 bits (before rate matching) and the new puncturing scheme we achieve a gain of 0.17 dB. For AMR 7.40 kbps we achieve a gain of about 0.1 dB. The performance improvement in case of the presented AMR simulations translates into a reduction of the necessary power per user of 2-4% and a corresponding capacity increase. The paper shows that end puncturing has very interesting properties regarding performance especially when supporting channel coding with short information blocks (e.g. when used with AMR). The proposed scheme is integrated within the rate matching algorithm and generally applicable for short, middle and long blocks. The shown performance improvement should even be higher for shorter codes. But with increasing blocksize the performance improvement decreases. The simulations show, that the improved algorithm is advantageous and we propose to establish a proper work item in order to allow to present the corresponding CRs.

References

[1] TSG R1-00254, “Improved End Puncturing Scheme for Convolutional Codes”, February 2000, Siemens

[2] TSG R1-99J08, “End puncturing for short convolutional codes”, November 1999, Siemens

[3] TSG R1-00486, “Simplification of Rate Matching Description and Optional Correction of Rate Matching Pattern Offset for Repetition”, April 2000, Siemens, LGIC

[4] TSG R1-99B85, “Effect of EEP and UEP on channel coding for AMR”, August 99, Nokia

[5] TSG R1-00095, “Improvement of convolutional coding scheme for very low code block size”, January 2000, Alcatel