3GPP TSG-RAN WG1#48bis **R1-071771**
St. Julians, Malta, March 26 – 30, 2007

Source: **Ericsson, Motorola**
Title: **Further Consideration on Code Shortening Procedures for Turbo Coding**
Agenda Item: **7.5**
Document for: **Discussion and Decision**

1. Introduction

For LTE turbo coding, a set of 188 byte-aligned block sizes from 40 to 6144 bits is defined [1]. To support transport blocks of variable sizes, code block segmentation [2] and code shortening techniques [3, 4] are adopted. The later involves padding the first code segment up to the next supported turbo coding block size prior to turbo encoding [5]. However, discussion and decision have not been made regarding further processing of these padding bits. In this paper, we analyze the performance of three such approaches.

Based on the performance results, we recommend removing all padding bits, which can be implemented with simple operations. That is, the pre-padding and depadding procedures should be conceptually treated as integral parts of the LTE turbo encoder. This change will only be applied to the first code segment [5] since the rest of the segments are not padded.

The full and partial depadding approaches achieve essentially identical performance. In the full depadding approach (illustrated in Figure 1(a)), additional operations are performed during encoding of the first segment such that the rate matching agents for all code segments behave identically. In the partial depadding approach (illustrated in Figure 1(c)), these operations are deferred into rate matching and, as a result, rate matching agent for the first code segment behaves differently than those for the rest of code segments. However, the choice of the partial depadding approach could have an impact on the rate matching algorithms since the two parity streams have different lengths. Therefore, the full depadding approach (Encoder I) is recommended for LTE.

2. Alternative Shortening Procedures

It is expected transport blocks in LTE will have byte-aligned sizes. Hence, code shortening procedure will not be used for code segments shorter than 512 bits. For a segment size K larger than 512, the supporting interleaver size S could be larger than K [5]. A block of $(S-K)$ zeros are pre-pended to the information bits before turbo encoding. The maximum amount of padding overhead depends on the segment size:

- For $512 < K < 1024$, maximum padding is 8 bits (up to 1.5%).
- For $1024 < K < 2048$, maximum padding is 24 bits (up to 2.3%).
- For $2048 < K < 6144$, maximum padding is 56 bits (up to 2.7%).

The turbo coded bits corresponding to the padding positions can be further processed in three different approaches as illustrated in Figure 1.

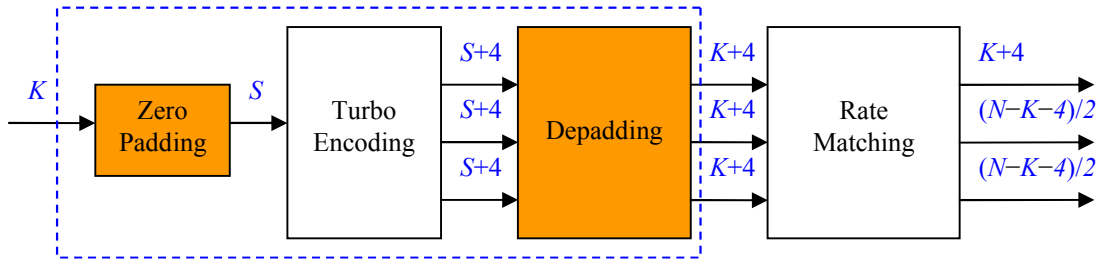
In the first full depadding approach (as recommended in [4]), the systematic, 1st parity, and 2nd parity bits corresponding to the padding bits are all removed. Thus, for a segment of K information bits, the output from the overall encoder consists of $3K+12$ bits. To the subsequent rate matching agent, the length of the systematic stream is $K+4$. If a systematic redundancy version of N bits is needed, the rate matching agent selects $K+4$ bits from the systematic stream and $(N-K-4)/2$ bits from each of the parity streams.

In the second approach, all bits corresponding to the padding zeros are passed through. Thus, for a segment of K input bits, the output from the overall encoder consists of $3S+12$ bits. To the subsequent rate matching agent, the length of the systematic stream is $S+4$. If a systematic redundancy version of N bits is needed, the rate matching agent selects $S+4$ bits from the systematic stream and $(N-S-4)/2$ bits from each of the parity streams. In other words, this raises the effective coding rate instructed to the rate matching agents. More specifically, the total number of coded bits is given by K/r if the desired code rate is r. Since the lengths of streams entering the rate matching agent are $S+4$ instead of $K+4$, the effective code rate instructed to the rate matching agent is higher: $r'=r\cdot S/K$. For instance, a target $r=0.9$ case would appear effectively as $r'=0.91$—0.92 to the rate matching agent.

For a given amount of channel bits N, different number parity bits (as well as rate matching patterns) are employed for the two different shortening approaches. Since $S>K$, more protective parity bits are transmitted in the first depadding approach. Furthermore, transmission energy is unnecessarily consumed by the redundant zeros if the second approach is adopted. Hence, the first depadding approach can be expected to outperform the second approach.

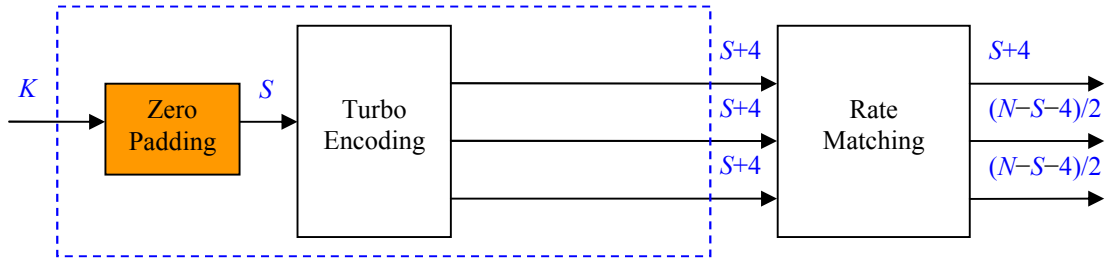
In the third partial depadding approach (as recommended in [3]), the systematic and 1st parity bits corresponding to the padding bits are removed but all 2nd parity bits are passed through. Thus, for a segment of K information bits, the output from the overall encoder consists of $2K+S+12$ bits. To the subsequent rate matching agent, the length of the systematic stream is $K+4$ but the two parity streams are of different lengths: $K+4$ and $S+4$. If a systematic redundancy version of N bits is needed, the rate matching agent selects $K+4$ bits from the systematic stream and $(N-K-4)/2$ bits from each of the parity streams.

The padding bits can be removed without impacting performance because the padding bits in the systematic and 1st parity streams are always zeros and hence carry no information. Though 2nd parity bits at the padding positions can be of arbitrary values, the corresponding systematic bits are known to be zeros. With this *a priori* knowledge, the decoder effectively operates on only half of the trellis branches (those corresponding to systematic bits being zero). Thus padding bits in the 2nd parity stream carry hardly any incremental information.



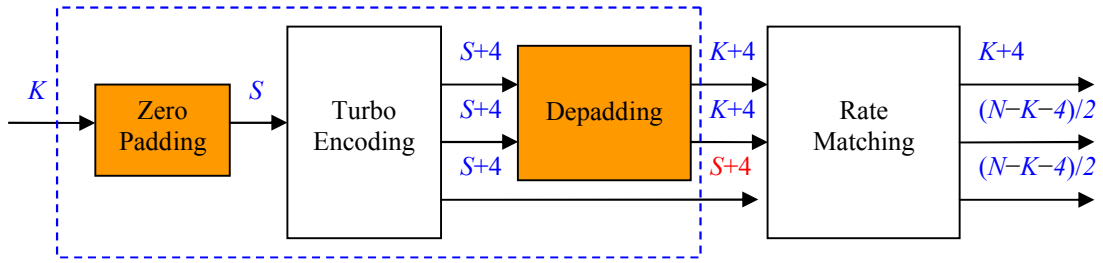
Overall Encoder I

(a) Full depadding approach



Overall Encoder II

(b) No depadding approach



Overall Encoder III

(c) Partial depadding approach

Figure 1 Alternative code shortening procedures

3. Performance Analysis

We compare the performance of all three shortening procedures assuming a Rel6 rate matching agent [6] subsequent to the turbo encoder. The simulation parameters are listed in Table I. The test block lengths are uniformly selected to cover the full range from 512 to 6144 bits with worst-case padding.

Table 1 Simulation Parameters

Common Code Structure	Prepadding + QPP-based Turbo Coding
Proposed Encoders	Shortening Encoder I with full depadding Shortening Encoder II without depadding Shortening Encoder III with partial depadding
Test Block Lengths	K = 520, 600, 680, 760, 840, 920, 1016, 1032, 1192, 1352, 1480, 1640, 1832, 2024, 2056, 2504, 2952, 3592, 4360, 5192, 6088
Coding Rates	r = 0.4, 0.5, 0.6, 0.7, 0.8
Rate Matching Setting	Rel6, RV=0 for QPSK [6]
Decoding Algorithm	Improved Max-Log-MAP [4]
Iterations	8
Modulation	QPSK
Channel	Static AWGN

3.1. Comparison of Encoder I and Encoder II

The required E_b/N_0 values for 10% and 1% BLER targets are plotted in Figure 2 and Figure 3, respectively. The performance losses caused by transmission of padding bits can be observed over the full range of segment sizes from 512 to 6144 bits. In general, the losses increase with coding rates since fewer parity bits are transmitted for high coding rate cases due to transmission of padded systematic bits. The performance losses are highest when the segment sizes are just above 512, 1024 or 2048 bits, where padding overheads (as percentages of the segment sizes) are higher. The losses then decrease with segment sizes (until the next junction point) since overall padding overheads decrease. The losses can reach more than 0.3 dB and create jumpy performance targets that can interfere with link adaptation algorithms.

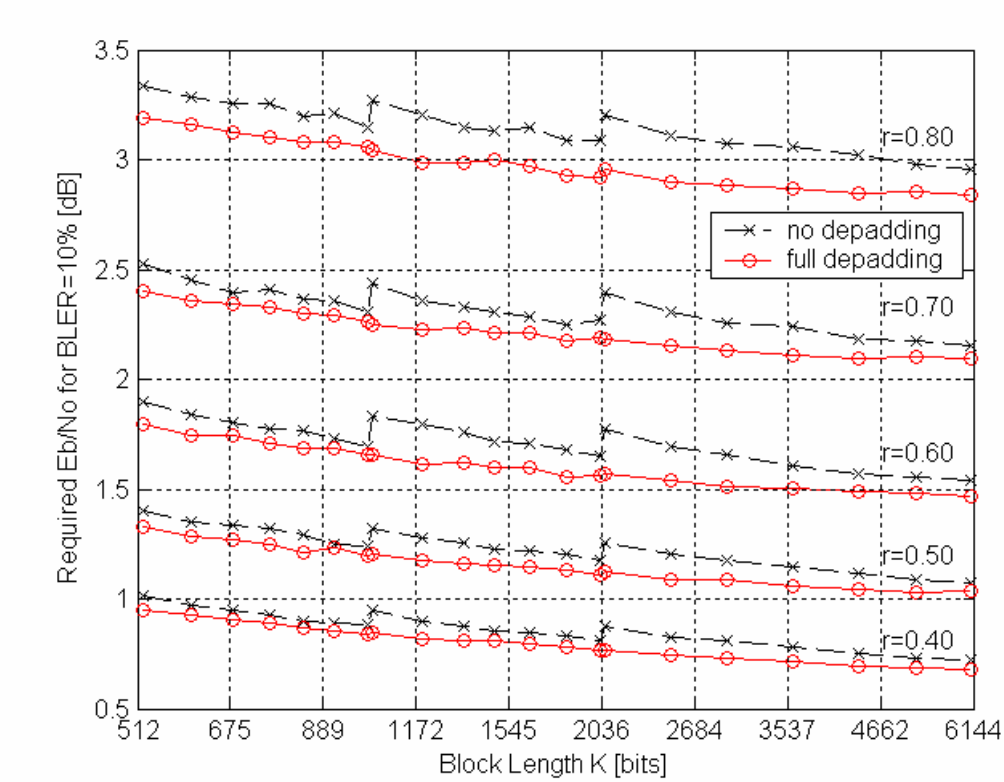


Figure 2 Performance comparison of Encoder I & II at BLER target=10%.

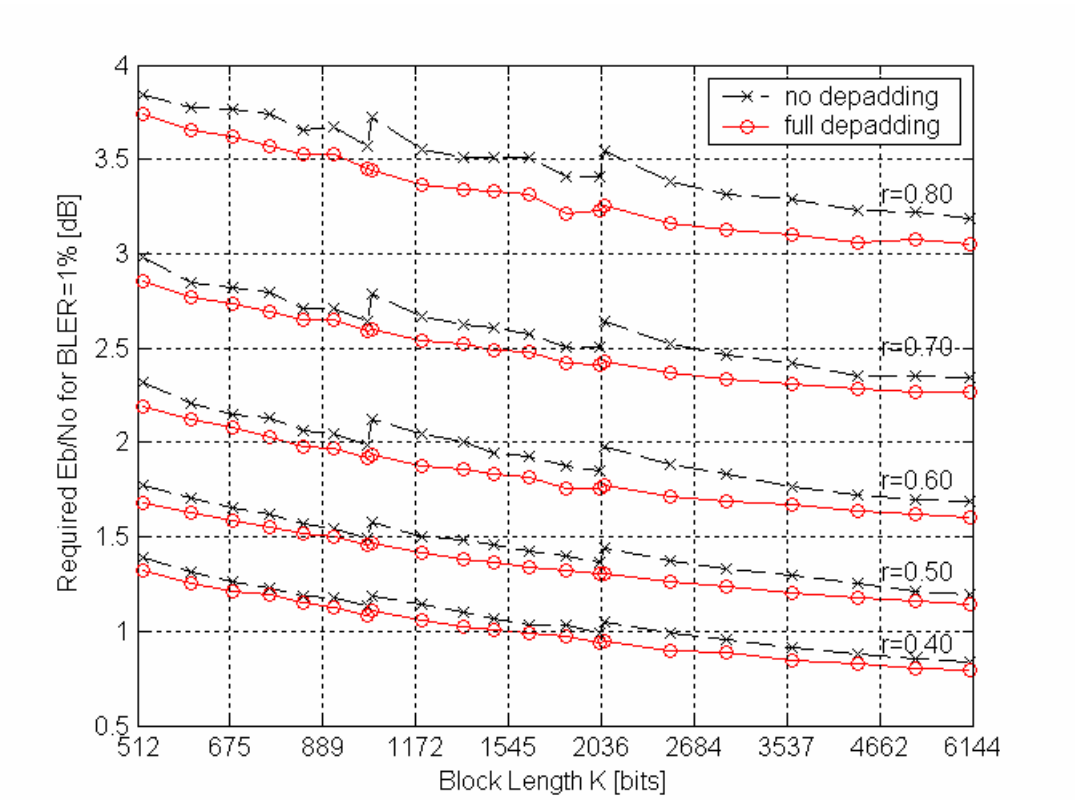


Figure 3 Performance comparison of Encoder I & II at BLER target=1%.

3.2. Comparison of Encoder I and Encoder III

The required E_b/N_0 values for 1% BLER targets are plotted in Figure 4. The two shortening encoders perform essentially identical though the performance of Encoder I is marginally better for smaller blocks at certain code rates.

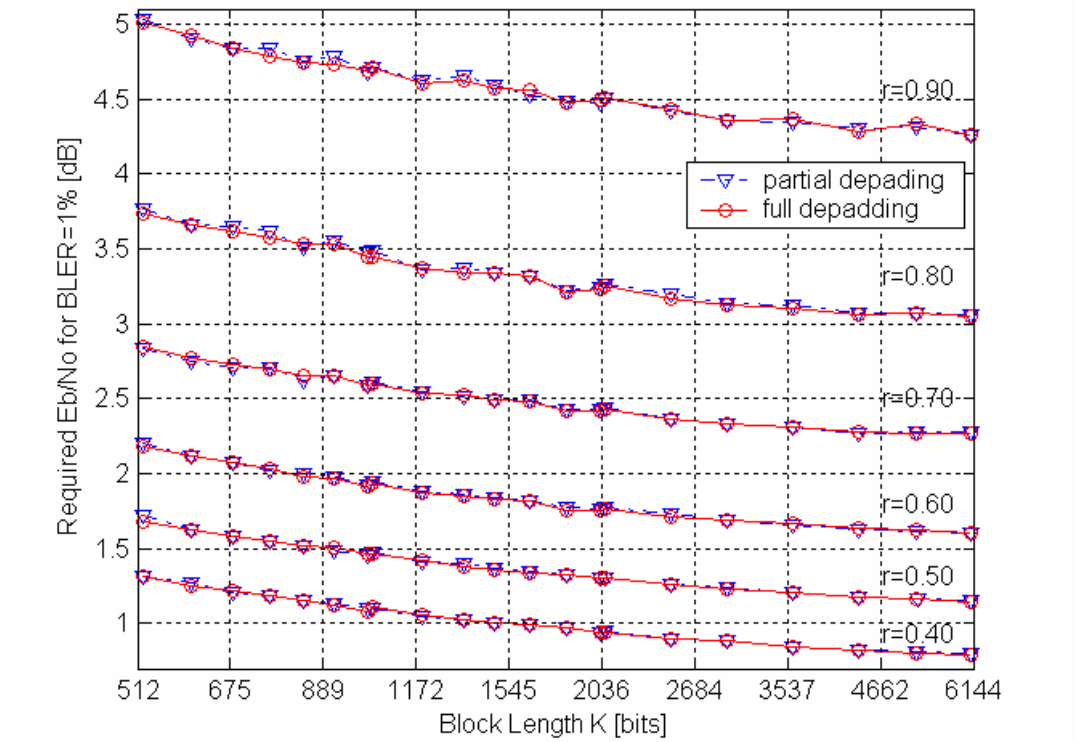


Figure 4 Performance comparison of Encoder I & III at BLER target=1%

4. Complexity Comparison of Encoders I & III

For both Encoder I and Encoder III, the depadding operations for the 1st constituent code are identical. The first ($S-K$) bits from the systematic and 1st parity streams are dropped.

For Encoder I, depadding for the 2nd constituent code can be implemented as follows. At the i-th clock tick, an information bit is read from the interleaving address $\Pi(i)$ and fed into the constituent encoder. If $\Pi(i) < S-K$, the corresponding parity bit is dropped. The complexity cost is apparently marginal. Since the three output streams are of the same length as illustrated in Figure 1(a), the streams can be multiplexed per Section 5.1.3.2 of TS 36.212 [5].

For Encoder III, all parity bits from the 2nd constituent code are retained and, hence, there is no additional operation during encoding. However, the subsequent rate matching agent needs to process ($S-K$) positions for the 2nd constituent code. Furthermore, the three output streams are of different lengths as illustrated in Figure 1(c). Revisions to TS 36.212 are needed to accommodate multiplexing of unequal lengths for the first code segment.

In one sense, the full depadding Encoder I terminates the code shortening procedure before rate matching since only length parameter, K+4, needs to be passed to the rate matching agent. On the other hand, the code shortening procedure continues into rate matching in the partial depadding Encoder III since two parameter, K+4 and S+4, need to be passed to the rate matching agent. As a result, the choice of between the two depadding approaches will have impact on the rate matching algorithms.

5. Conclusion

Transmission of padded bits results in performance losses for a wide range of segment sizes. The losses increase with coding rates and could be in excess of 0.3 dB. The losses further induce irregular performance targets that can interfere with link adaptation algorithms. Therefore, we recommend removing all these padded bits according to the full depadding Encoder I. That is, the pre-padding and depadding procedures are conceptually treated as an integral part of the LTE turbo encoder. This change requires very simple operations and will only affect the processing of the first code segment [5].

The full and partial depadding approaches achieve essentially identical performance. In the full depadding approach, additional operations are performed during encoding of the first segment such that the rate matching agents for all code segments behave identically. In the partial depadding approach, these operations are deferred into rate matching and, as a result, rate matching agent for the first code segment behaves differently than those for the rest of code segments. However, the choice of the partial depadding approach could have an impact on the rate matching algorithms since the two parity streams have different lengths.

6. References

[1] R1-071195, Ericsson, Motorola, Broadcom, ZTE, Nokia, Texas Instruments, Qualcomm Europe, Freescale, CATT, ITRI, NTT DoCoMo, Panasonic, Samsung, Mitsubishi, Nortel, "QPP Interleaver Parameters," 3GPP TSG RAN WG1#48, St. Louis, USA, Feb. 12 – 16, 2007.

[2] R1-071059, Motorola, " Code Block Segmentation for LTE Channel coding," 3GPP TSG RAN WG1#48, St. Louis, USA, Feb. 12 – 16, 2007.

[3] R1-062079, Motorola, "Eliminating tail bits in LTE channel coding," 3GPP TSG RAN WG1#46, Tallinn, Estonia, Aug. 28 – Sep. 1, 2006.

[4] R1-070464, Ericsson, "Performance comparison of LTE turbo internal interleaver proposals," 3GPP TSG RAN WG1#47bis, Sorrento, Italy, Jan. 15– 19, 2006.

[5] 3GPP TS 36.212 v1, "Multiplexing and Channel Coding (Release 8).

[6] 3GPP TS 25.212 v6, "Multiplexing and Channel Coding (FDD) (Release 6)."