



HAL
open science

Enhanced Turbo Codes for NR: Implementation Details

Charbel Abdel Nour

► **To cite this version:**

Charbel Abdel Nour. Enhanced Turbo Codes for NR: Implementation Details. [Technical Report] 3GPP TSG-RAN WG1 Meeting #86. 2016. hal-02976830

HAL Id: hal-02976830

<https://hal.science/hal-02976830v1>

Submitted on 23 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Agenda item: 8.1.4.1

Source: Orange and Institut Mines-Telecom

Title: Enhanced Turbo Codes for NR: Implementation Details

Document for: Discussion

1 Introduction

The first part of this document (sections 0 to 5) describes the implementation of the enhanced Turbo Coding scheme proposed in R1-164635 at RAN1#85 meeting. The following items are described:

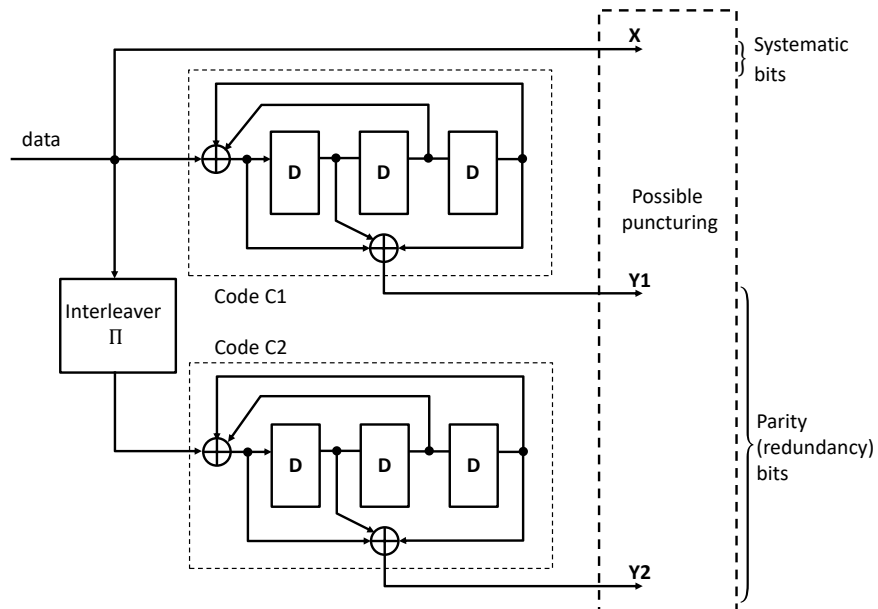
- Component codes
- Tail-biting trellis termination technique
- Puncturing patterns
- Interleaving equations and parameters

The second part of the document (sections 6 and 7) provides the puncturing patterns and the interleaving parameters used to obtain the error rate performance curves presented in R1-164635 (section 6) and in R1-167414 (section 7)

2 Overall turbo encoder structure

2.1 Coding rates $R \geq 1/3$

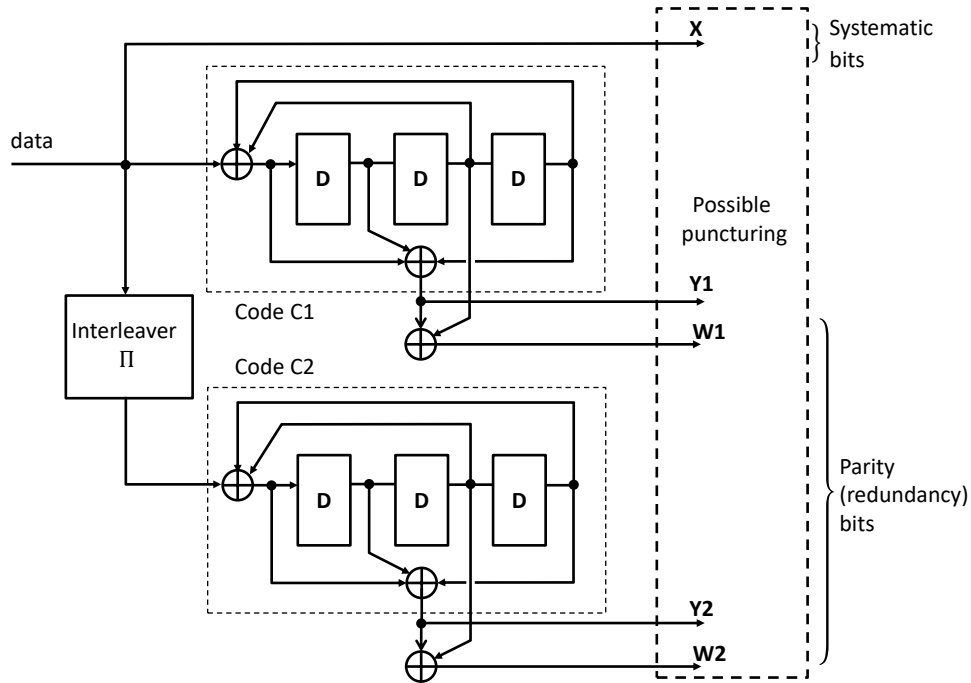
The proposed encoder structure uses the same component encoders as in LTE.



Generator polynomials for C1 and C2: $(1, (1+D+D^3)/(1+D^2+D^3))$

2.2 Coding rates $1/5 \leq R < 1/3$

If coding rates lower than $1/3$ are required, an additional parity is computed for each component convolutional encoder.



Generator polynomials for C1 and C2: $(1, (1+D+D^3)/(1+D^2+D^3), (1+D+D^2+D^3)/(1+D^2+D^3))$

3 Trellis termination: tail-biting turbo codes

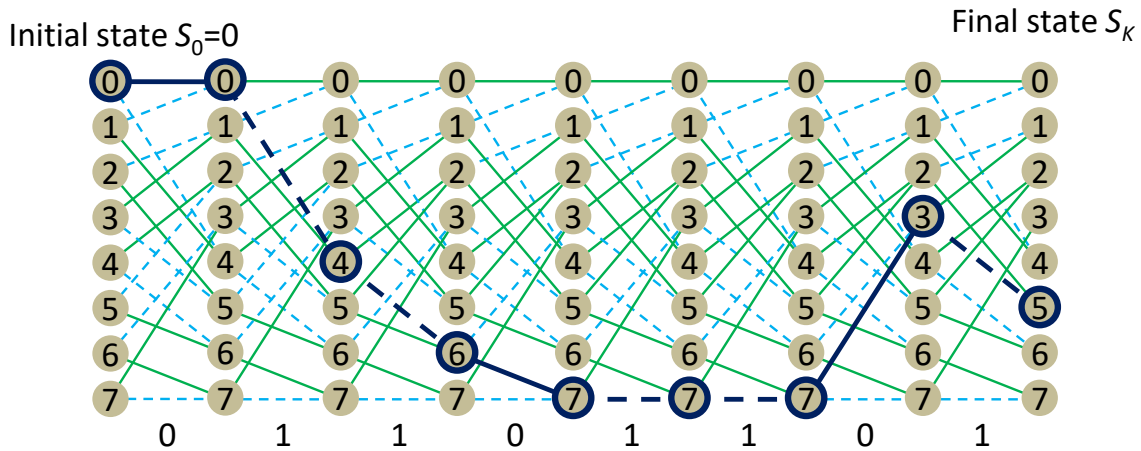
Among the different techniques aiming at transforming a convolutional code into a block code, tail-biting, also called circular encoding, ensures that, when encoding a message of length K , the initial and the final states are identical for each component encoder C1 and C2 [2].

Tail-biting is the best known termination method for turbo codes:

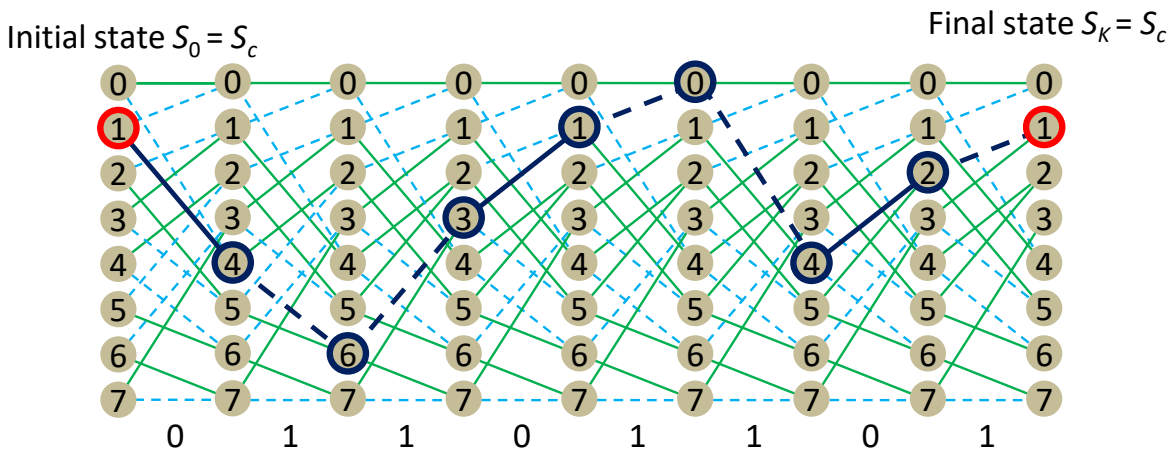
- First, **no extra bits have to be added and transmitted**; thus, there is **no rate loss** and the spectral efficiency of the transmission is not reduced;
- Next, tail-biting does not induce any side effect in the message. Consequently, all the information bits are protected in the same way by the turbo code and the circular property prevents the occurrence of low-weight truncated codewords since the trellis can be considered as with infinite length. Therefore, **tail-biting termination helps to lower the error floor**.

3.1 Tail-biting encoding

Tail-biting encoding for each component encoder C1 and C2 requires a two-step encoding process illustrated by the following figure:



(a) Circular encoding of message "01101101" with C1 or C2, first encoding step: the message is encoded from state 0. The final state is $S_k = 5$



(b) Circular encoding of message "01101101" with C1 or C2, second – actual – encoding step: the message is encoded from circulation state $S_c = 1$ and the final state is also $S_c = 1$

- The first step determines the initial/final state (called circulation state S_c): the information message is encoded from initial state zero. The resulting final state is S_K .
- The value of the circulation state S_c is a function of the final state S_K and of the value of $(K \bmod 7)$. For recursive generator polynomial $(1+D^2+D^3)$, S_c is given by the following table:

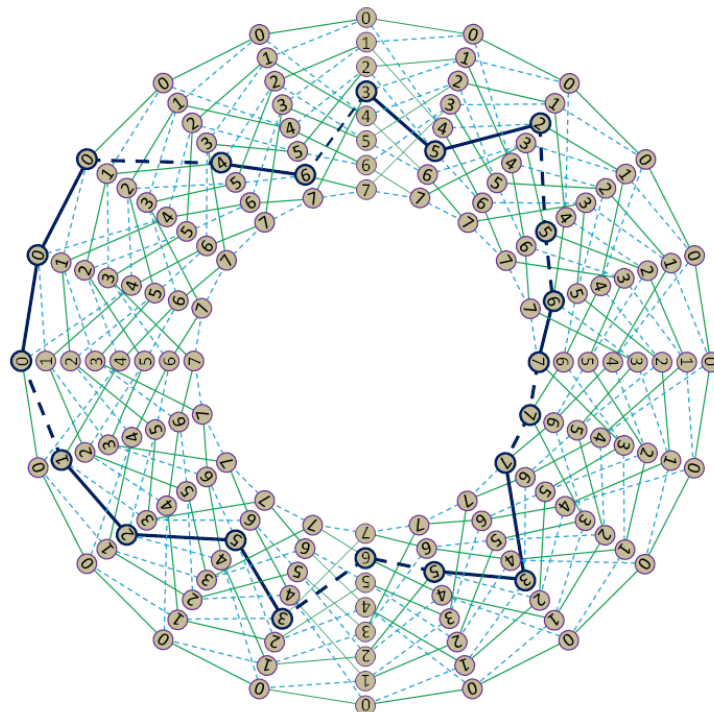
$S_K \backslash K \bmod 7$	1	2	3	4	5	6
0	0	0	0	0	0	0
1	6	4	3	2	5	7
2	3	5	4	6	7	1
3	5	1	7	4	2	6
4	7	2	1	5	6	3
5	1	6	2	7	3	4
6	4	7	5	3	1	2
7	2	3	6	1	4	5

Note that the tail-biting technique is not directly applicable when K is a multiple of 7.

- The second step is the actual encoding stage. The encoder starts in the correct initial state S_c and the corresponding valid codeword is delivered.

3.2 Tail-biting decoding

When tail-biting is adopted, the code trellis can be viewed as a circle (see figure below): the iterative decoding of such codes involves repeated and continuous turns around the circular trellis. The number of turns performed is equal to the required number of iterations. The state probabilities or metrics, according to the chosen decoding algorithm, computed at the end of each turn are used as initial values for the next turn. For instance, if the MAP, Log-MAP or Max-Log-MAP soft-in soft-out decoding algorithm is used, the final forward (respectively backward) state probabilities or metrics computed at the end of each iteration are used as initial values for the next iteration. The decoding process can start anywhere in the circular trellis. At the first iteration, the forward and backward processes can be initialized with equiprobable values.



4 Puncturing

Let us denote by K the size of the information message to be encoded.

Puncturing is performed using **periodic puncturing patterns** of length Q . K is assumed to be a multiple of Q . Typical values for Q are 4, 8, 16 or 32 but others values are possible, provided that Q is a divisor of K .

The selection of the puncturing patterns is performed on the basis of the joint analysis of the Hamming distance spectrum of the punctured component convolutional code and of the mutual information exchange between the two component encoders [1].

It was shown that, for a given coding rate, the same puncturing pattern can be used for a large range of information message lengths K . In practice, two sets of puncturing patterns are needed:

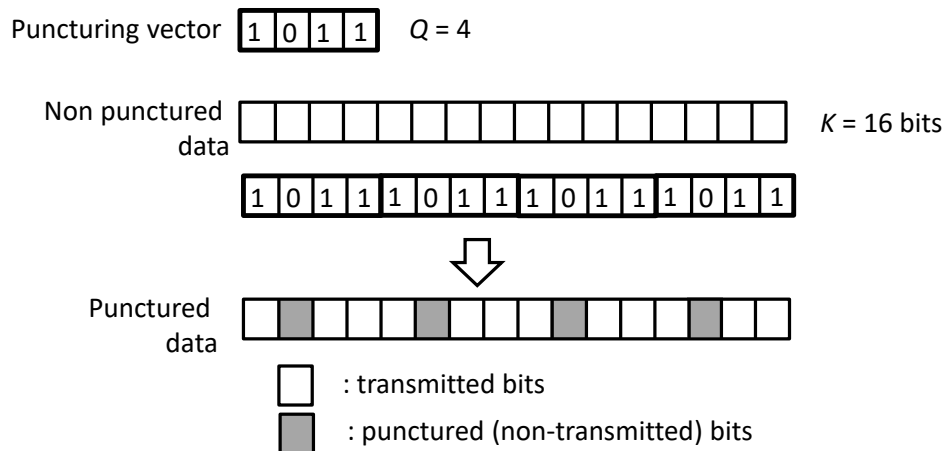
- A set of puncturing patterns for short block lengths (up to a few hundred bits): for **short data blocks, only parity bits are punctured, all the systematic bits are transmitted;**
- A set of puncturing patterns for long block lengths: for **medium and long data blocks, parity and systematic bits are punctured.**

4.1 Coding rates $R > 1/3$

For each coding rate greater than $1/3$, the **puncturing pattern consists of two puncturing vectors** of length Q : one for systematic bits (X), the second for the parity bits ($Y1$ and $Y2$). The same puncturing vector is applied to $Y1$ and $Y2$.

Puncturing vectors are binary vectors. "0" values indicate the bits to be punctured (not transmitted) and "1" values indicate the coded bits to be transmitted.

Example:



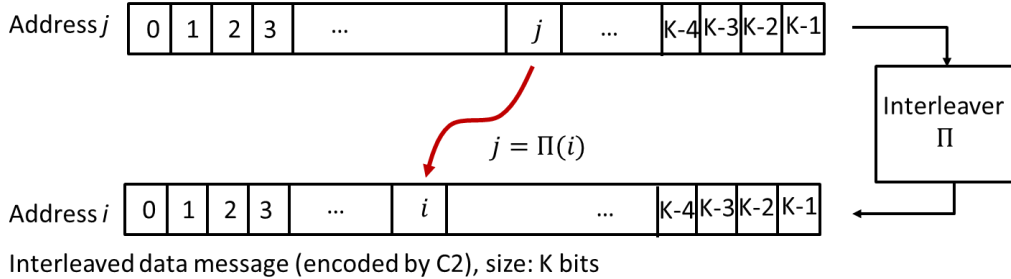
4.2 Coding rates $1/3 > R > 1/5$

For each coding rate greater than $1/5$ and lower than $1/3$, the **puncturing pattern consists of three puncturing vectors** of length Q : one for systematic bits (X), one for parity bits Y ($Y1$ and $Y2$) and one for parity bits W ($W1$ and $W2$). The same puncturing vector is applied to $Y1$ and $Y2$. The same puncturing vector is applied to $W1$ and $W2$.

5 Interleaver Π

5.1 Definition of the interleaving function Π

Non-interleaved data message (encoded by C1), size: K bits



The systematic bits are interleaved following an **Almost Regular Permutation (ARP)** process [4]:

For $i = 0 \dots K - 1$

$$j = \Pi(i) = [Pi + S(i \bmod Q)] \bmod K$$

Interpretation: the data value at address i in the interleaved message comes from address $j = \Pi(i)$ in the non-interleaved message.

S is a vector of size Q . $P, S(0) \dots S(Q - 1)$ values depend on K and on the coding rate.

Π is a bijection provided that:

- P and K are mutually prime numbers,
- Q is a divisor of K ,
- the values $S(0) \dots S(Q - 1)$ belong to the Q different congruence classes modulo Q .

5.2 Contention-free property

The ARP interleaver is contention-free for every window size W of length qQ , provided that qQ is a divisor of K , thus allowing a parallelism degree of qQ to be naturally achievable.

Proof: $\Pi(i + qQ) - \Pi(i) = qPQ \bmod K$. Consequently, if the memory is split into $\frac{K}{qQ}$ memory banks of size $W = qQ$, the symbols at interleaved addresses $i, i + qQ, i + 2qQ, i + 3qQ, \dots, i + (\frac{K}{qQ} - 1)qQ$, (i.e. the $\frac{K}{qQ}$ symbols at the same address in the $\frac{K}{qQ}$ different memory banks) come from the non-interleaved addresses $\Pi(i), (\Pi(i) + qPQ) \bmod K, (\Pi(i) + 2qPQ) \bmod K, (\Pi(i) + 3qPQ) \bmod K, \dots, (\Pi(i) + (\frac{K}{qQ} - 1)qPQ) \bmod K$, which represent the same address in the $\frac{K}{qQ}$ different memory banks, provided that P and K are mutually prime numbers.

Example: for information message length $K = 8000$, puncturing mask length $Q = 16$, the interleaver structure allows natural parallelism degrees equal to 2, 4, 5, 10, 20, 25, 50, 100, 125, 250, 500 (*i. e.* for integer values of $\frac{K}{qQ}$).

5.3 Principle of selection of the interleaver parameters, $S(0) \dots S(Q - 1)$ and P

Selecting the values of parameters $S(0) \dots S(Q - 1)$ and P involves the following steps, partially described in [1]:

- Selection of P : only the values of P ensuring a high minimum span of the regular interleaver ($\Pi(i) = P \times i \text{ mod } K$) are retained;
- In the puncturing pattern of length Q , the data positions are sorted according to their protection level;
- $S(0) \dots S(Q - 1)$: the values of these parameters are chosen in a such a way that data positions with a good level of protection in the puncturing pattern are connected with data positions with a poor level of protection.
- Only the resulting interleavers with a high minimum span and a high correlation girth are kept and their minimum Hamming distance is evaluated.

This selection process can be applied in different ways:

- To jointly optimize the code performance for each code rate and block length: a set of parameters is then computed for each rate/length combination (method applied to obtain the parameters described in section 7.1);
- To design rate-compatible codes: the parameters are optimized for the highest code rate. The selected set of parameters is valid for all coding rates (method applied to obtain the parameters described in section 7.2).

6 Puncturing patterns and interleaver parameters used for results presented at RAN1#85 in R1-164635

The puncturing masks and the ARP interleaver parameters for the different coding rates and information frame sizes are listed in the following tables:

R	K	Puncturing vectors : Systematic (X)/ parity (Y1 and Y2)
2/3	48	11111111 / 10001000
4/5	48	1111111111111111 / 0100000001000000
8/9	96	1111111111111111 / 0100000000000000
2/3	1504, 6144	01111110 / 11000001
4/5	1504, 6144	0111111111111101 / 1100000000000010
8/9	1504, 8000	0111111011111111 / 1010000000000000

Note : "1" means "non-punctured (transmitted) symbol" ad "0" means "punctured (non-transmitted) symbol"

		ARP interleaver parameters		
R	K	Q	P	$(S(0) \dots S(Q - 1))$
1/5	100	4	13	(0, 73, 99, 24)
2/3	48	8	17	(7, 5, 43, 9, 15, 13, 35, 1)
4/5	48	16	35	(10, 42, 2, 38, 18, 26, 26, 22, 10, 42, 2, 38, 18, 26, 26, 22)
8/9	96	16	35	(8, 60, 47, 78, 26, 67, 82, 78, 88, 81, 80, 92, 37, 84, 16, 68)
2/3	1504	8	651	(0, 89, 528, 852, 1501, 1396, 688, 490)
4/5	1504	16	365	(0, 1261, 1374, 1279, 417, 867, 549, 514, 730, 474, 1359, 285, 927, 670, 1176, 1078)
8/9	1504	16	487	(0, 943, 1030, 1320, 182, 955, 887, 441, 1040, 1452, 625, 364, 1195, 280, 835, 1363)
2/3	6144	8	701	(0, 2479, 1115, 5007, 2957, 3489, 4661, 1996)
4/5	6144	16	953	(0, 801, 3826, 5118, 210, 1698, 6065, 684, 5076, 2695, 4119, 5250, 775, 3768, 624, 11)
1/5	8000	16	1507	(8, 3443, 7934, 4420, 7894, 7903, 511, 7397, 2248, 4672, 1481, 3464, 1675, 3116, 7435, 4735)
8/9	8000	16	5153	(8, 4709, 3666, 4474, 4686, 1241, 1355, 6499, 952, 6114, 3037, 910, 3315, 5718, 7143, 2541)

7 Puncturing patterns and interleaver parameters used for results presented at RAN1#86 in R1-167414

7.1 Optimized Turbo Code for $K = 6000$ and $K = 8000$

A joint optimization of the puncturing patterns and of the ARP interleaver parameters has been carried out for two information block lengths, $K = 6000$ and $K = 8000$. The following subsections provide the tables describing the proposed puncturing masks and ARP interleaver parameters.

7.1.1 Puncturing patterns for $K = 6000$ and $K = 8000$

R	K	Puncturing vectors : Systematic (X)/ parity (Y1 and Y2)
2/5	6000, 8000	111111110111110 / 1011101111101111
1/2	6000, 8000	1111111100111111 / 1011001111001010
2/3	6000, 8000	01111110 / 11000001
3/4	6000, 8004	111111011110 / 010001010000
5/6	6000, 8000	111110111110111111 / 10000010000100000000
8/9	6000, 8000	0111111101111111 / 1010000000000000

Note : "1" means "non-punctured (transmitted) symbol" ad "0" means "punctured (non-transmitted) symbol"

7.1.2 ARP interleaver parameters for $K = 6000$

		ARP interleaver parameters			
R	K	Q	P	$(S(0) \dots S(Q - 1))$	
1/5	6000	16	2029	(8, 3769, 5626, 2022, 5486, 2637, 3283, 1935, 4648, 5990, 2373, 1258, 1395, 2426, 31, 4569)	
1/3	6000	16	3761	(8, 1381, 2514, 1882, 254, 4921, 1915, 1811, 648, 4658, 3261, 4030, 4771, 2102, 3351, 3197)	
2/5	6000	16	3703	(13, 1100, 5158, 2492, 2374, 5123, 3163, 90, 2084, 96, 2872, 4314, 4660, 4837, 4376, 3952)	
1/2	6000	16	1613	(11, 5089, 5995, 2886, 2914, 49, 4918, 3583, 2849, 5731, 421, 3265, 5203, 4650, 1083, 2937)	
2/3	6000	16	2081	(7, 5923, 2036, 5442, 2557, 2014, 4028, 1289, 4007, 627, 3860, 98, 1501, 1742, 3708, 5721)	
3/4	6000	12	2741	(8, 5042, 2159, 1213, 3163, 4701, 4169, 5690, 3092, 1613, 3747, 2615)	
5/6	6000	20	3469	(1, 2091, 1455, 2702, 1392, 3207, 4385, 1468, 3192, 902, 2848, 4068, 937, 385, 2210, 4982, 3150, 2062, 5008, 5395)	
8/9	6000	16	943	(8, 2807, 3062, 5712, 3606, 1411, 2263, 5825, 5592, 292, 1121, 852, 4523, 272, 2947, 299)	

7.1.3 ARP interleaver parameters for $K = 8000$

		ARP interleaver parameters			
R	K	Q	P	$(S(0) \dots S(Q - 1))$	
1/5	8000	16	1507	(8, 3443, 7934, 4420, 7894, 7903, 511, 7397, 2248, 4672, 1481, 3464, 1675, 3116, 7435, 4735)	
1/3	8000	16	4989	(8, 121, 7258, 5302, 7390, 1037, 4819, 6031, 7656, 486, 4757, 7290, 7747, 7626, 1087, 6873)	
2/5	8000	16	1011	(13, 4496, 7502, 1624, 6454, 1063, 5587, 7782, 1156, 7364, 2976, 4486, 2452, 2185, 6256, 4764)	
1/2	8000	16	5451	(11, 5251, 2991, 1164, 7450, 6827, 962, 5197, 1089, 7573, 6489, 2375, 3979, 20, 4743, 5639)	
2/3	8000	16	7787	(7, 4521, 5152, 6836, 2325, 3052, 7648, 1987, 4023, 4105, 2048, 4772, 3221, 6572, 6224, 6259)	
3/4	8004	12	5237	(8, 902, 3911, 181, 4615, 7137, 6017, 7478, 4028, 1817, 1911, 431)	
5/6	8000	20	2469	(1, 3091, 4555, 3962, 3252, 3127, 4425, 1848, 4672, 6622, 388, 7868, 2897, 2065, 6630, 262, 710, 3422, 1268, 3415)	
8/9	8000	16	5153	(8, 4709, 3666, 4474, 4686, 1241, 1355, 6499, 952, 6114, 3037, 910, 3315, 5718, 7143, 2541)	

7.2 Rate-Compatible Turbo Codes for $K = 96$ and $K = 4000$

A Rate-Compatible family of Turbo Codes can also be designed, which uses the same ARP interleaver parameters for all coding rates. Such a Turbo Code is compatible with the current circular buffer structure adopted in LTE and LTE-A. The following subsections describe the tables providing the

proposed puncturing masks and ARP interleaver parameters for block length $K = 96$ and $K = 4000$. As described in section 4, the puncturing patterns are different for short and long block lengths: for **short data blocks, only parity bits are punctured, all the systematic bits are transmitted** and for **medium and long data blocks, parity and systematic bits are punctured**.

7.2.1 Puncturing patterns for $K = 96$

R	Parity puncturing vectors (Y1 and Y2) for systematic (X) = 1111111111111111
8/23	1101111111111111
4/11	1101111111101111
8/21	1101111101101111
2/5	1101011101101111
8/19	1101011101101110
4/9	1101011101001110
8/17	1101011100001110
1/2	1101011100001010
8/15	1101010100001010
4/7	1101010000001010
8/13	1101010000000010
2/3	1100010000000010
8/11	1100000000000010
4/5	0100000000000010
8/9	0100000000000000

Note : "1" means "non-punctured (transmitted) symbol" ad "0" means "punctured (non-transmitted) symbol"

7.2.2 Puncturing patterns for $K = 4000$

R	Parity puncturing vectors (Y1 and Y2) for systematic (X) = 0111111101111111
8/23	1111111111111111
4/11	1110111111111111
8/21	1010111111111111
2/5	1010111110111111
8/19	1010111110111101
4/9	1010111010111101
8/17	1010111010110101
1/2	1010111000110101
8/15	1010111000110100
4/7	1010111000110000
8/13	1010110000110000
2/3	1010010000110000
8/11	1010000000110000
4/5	1010000000010000
8/9	1010000000000000

Note : "1" means "non-punctured (transmitted) symbol" ad "0" means "punctured (non-transmitted) symbol"

7.2.3 ARP interleaver parameters for $K=96$

ARP interleaver parameters ($K=96$)		
Q	P	$(S(0) \dots S(Q-1))$
16	37	(8, 26, 59, 72, 66, 57, 38, 32, 72, 79, 60, 38, 13, 10, 52, 54)

7.2.4 ARP interleaver parameters for $K=4000$

ARP interleaver parameters ($K=4000$)		
Q	P	$(S(0) \dots S(Q-1))$
16	1223	(8, 607, 1318, 472, 1478, 1963, 1895, 777, 2952, 3148, 3185, 1644, 2443, 1560, 1315, 163)

8 References

- [1] R1-164635, "Improved LTE turbo codes for NR," Orange.
- [2] C. Weiss, C. Bettstetter, and S. Riedel, "Code construction and decoding of parallel concatenated tail-biting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 366–386, Jan 2001.
- [3] R. Garzon Bohorquez, C. Abdel Nour, and C. Douillard, "Improving Turbo Codes for 5G with Parity Puncture-Constrained Interleavers," 9th Int. Symp. On Turbo Codes and Iterative Information Processing (ISTC'2016), Brest, France, Sept. 2016.
- [4] C. Berrou, Y. Saouter, C. Douillard, S. Kerouedan and M. Jezequel, "Designing good permutations for Turbo Codes: towards a single model," *IEEE Int. Conf. on Commun. (ICC'04)*, Paris, France, June 2004.
- [5] S. Crozier, "New high-spread high-distance interleavers for turbo-codes," *20th Biennial Symposium on Communications*, Queen's University, Kingston, Ontario, Canada, May 2000, pp. 3–7.