Supplementary Material of LTE2B

Ruofeng Liu, Zhimeng Yin, Wenchao Jiang, Tian He
Department of Computer Science and Engineering, University of Minnesota
(liux4189,yinxx283,jiang832,tianhe)@umn.edu

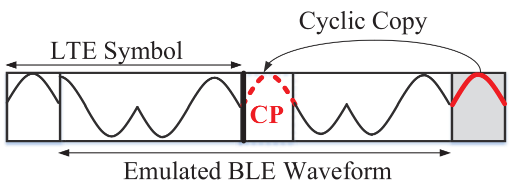

Figure 1: Time Domain Emulation of BLE in LTE2B.

1 TIME EMULATION OF BLUETOOTH LE

The time-domain emulation of BLE waveform is depicted in Fig.1 where LTE2B is able to emulate a BLE frame of legitimate length, i.e., 16-byte with 2 LTE symbols. Thus, the emulation of signal can avoid DMRS (mentioned in our paper) and only overlap with one LTE GI. To cope with one GI (also called CP) in the middle of the emulated signal, we propose the following designs. As depicted in the second LTE symbol in Fig.1, GI is cyclic copy of the last partition of a LTE symbol, i.e., the shadowed section. In order to create specific waveform in the GI LTE2B carefully manipulates the last partition of the 2^nd LTE symobl which will be copied to GI.

In this way, we are also control the 128 us waveforms to produce arbitrary BLE of 16 bytes. We evaluate BLE emulation with COTS BLE devices CC2650 and the result shows the emulation is accurate.

2 PROOF THAT TURBO&CRC ARE LINEAR

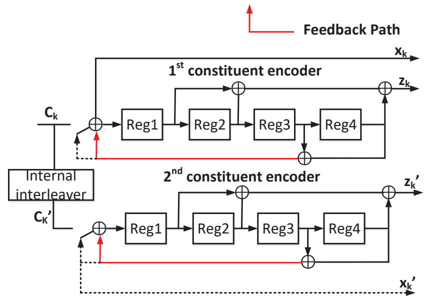

Figure 2: Turbo Encoder.

THEOREM 1. *The turbo coding calculations are systems of gf(2) linear equations.*

THEOREM 2. *The CRC calculations are systems of gf(2) linear equations.*

2.1 Proof of Theorem 1

PROOF. As Fig.2 shows, all the computation in the both encoders are XOR operations. We take the first constituent encoder as an example. We can express the relationships between input stream i_k, the output stream x_k, z_k and the values in the three registers with the following iterative equations.

$$Reg_i = Reg_{i-1}, i = 4, 3, 2 \tag{1}$$

$$Reg_1 = i_k \oplus Reg_3 \oplus Reg_4 \tag{2}$$

$$z_k = Reg_1 \oplus Reg_2 \oplus Reg_4 \tag{3}$$

$$x_k = i_k \tag{4}$$

The first equation performs the right shift of the registers. The second equation calculates the feedback bit and XOR the k^{th} input with it to get new value for the first register. The k^{th} output of z stream is given in the third equation by XOR the value in the $1^{st}, 2^{nd} and 4^{th}$ registers. Finally, the last equation gives the k^{th} output of x stream, which is identical with k^{th} input. These four steps will be repeated K+3 times. K is the length of "C bits". For the first K inputs, i_k equals the c_k. The last 3 tail inputs are coming from feedback.

Thus, after all the iterations we can expresses each x_k and z_k as XOR of a subset of input bits in the “C bit”. The same conclusion hold for x_k' and z_k'. In conclusion, turbo coding are systems of linear gf(2) equations. □

2.2 Proof of Theorem 2

PROOF. Here we take Type-24B CRC as an example. We model CRC calculation with following iterative equations, where Reg_i, with $i = 1...25$ are twenty-five shift registers for CRC calculation.

$$Reg_i = Reg_{i+1}, i = 1...24 \tag{5}$$

$$Reg_{25} = i_k \tag{6}$$

$$Reg_i = Reg_i \oplus (Reg_1 \times P_i), i = 1...25 \tag{7}$$

The first equation performs the left shift of all the registers. And then r in the second equation the k^th input are assigned to the last register. If the current value in the first register is one, then bit-wise XOR is performs on the values in the registers and a predefined PN sequence. We observe that these three operations are also linear and thus the relationship between source bits and their CRC bits can also be expressed as a system of 24 linear gf(2) equations. □