Modified Method for FEC Protection of Prior 'Uncoded Bits'

Will Bliss, Broadcom Corp.

IEEE P802.3bq 40GBASE-T task force

Norfolk, VA

May 12, 2014

Motivation and Overview

- Industry recognizes the risks and costs of the 10GBASE-T format, that leaves a large fraction of bits only 'protected' by increased Euclidian distance, but with no protection from FEC
- Several good proposals already to provide FEC protection for all bits
- ***lagner_3bq_01_0514.pdf*** proposed keeping the same 'Frame Size' but changing multiple Bauds per Frame from DSQ-128 to DSQ-256 (effectively PAM-16) to provide the extra redundancy. These symbols have 3dB loss of Euclidian distance. This might create other problems or limitations?
- ***lo_3bq_01_0514.pdf*** proposed using higher rate 256/257 transcoding to provide the extra redundancy
  - This presentation suggests modification(s) and options based on ***lo_3bq_01_0514.pdf***

Review of 10GBASE-T Format

- Each Frame carries 50 ‘blocks’ of 64/65 transcoded data
- So only 50*64 = 3200 ‘data bits’ (ignoring control here)
- 512 DSQ symbols in each Frame, so 512*7=3584 total bits transmitted per Frame
  - 512*3=1536 ‘uncoded’ (aka MSBs) bits
  - 512*4=2048 ‘coded bits’ (aka LSBs) make up LDPC codeword
    - 325 parity bits + 1723 ‘data bits’ in LDPC codeword
  - Total 1723+ 1536 = 3259 information carrying bits
- So 3259-3200 = 59 bits available for transcoding + all else
  - in 10GBASE-T else is 8bit CRC + 1 bit AUX + 50 bits transcoding

Summary *lo_3bq_01_0514.pdf*

- Chose transcoding 12* 256/257 + 2* 64/65
- Makes a Frame with the same 3200 'data' bits
- Total of 14 bits used for transcoding, so 59-14=45 are available for 'all else'
- If 8 for CRC and 1 for Aux, then 45-9= 36 bits remain for FEC on 'uncoded bits' (lets start calling them the 'now coded' MSB bits)
- Enough for T=2 RS FEC over GF(2^8), r=32 bits
- Alternate w/o CRC with T=2 D=3 (3 symbol error detecting) RS FEC with r= 40 bits
  - Can we allow NO checking on the LDPC data bits? (no CRC?)
- Packing of MSBs into RS symbols not detailed?

Alternate2 Transcoding Choice

- Same basic approach as ***lo_3bq_01_0514.pdf***, except
- chose 12* 256/257 + 1* 128/129 blocks
  - Details of “generalizing’ 256/257 to 128/129 are omitted here!
  - Makes a Frame with the same 3200 ‘data’ bits
  - Total of 13 bits used for transcoding, so 59-13=46 are available for ‘all else’
  - If 1 Aux, then 46-1= 45 bits remain for FEC on ‘now-coded MSB bits’
  - Enough for T=2 D=3 RS FEC over GF(2^9), r=45 bits
  - Form 9-bit RS symbols by grouping 3 consecutive ‘MSBs’ (3-bit chunks) from the same TP
    - Some irregularity at end
  - Include LSBs (of LDPC code) in the RS code for error detection (replaces CRC)
    - The RS code over GF(2^9) is long enough to cover all bits

Motivation 1 for 9-bit RS symbols and packing

- The DSQ-128 mapping of MSBs is not Gray
  - So some minimum Euclidian distance errors produce 2 bit errors in decoded output
  - Using 8-bit symbols means that some of those 2-bit errors will cross the RS symbol boundary, using the full T=2 correction power of the FEC

Motivation 2 for 9-bit RS symbols and packing

- We're concerned about certain impulsive noises
  - That can produce 'MSB errors' that are larger Euclidian distance than the minimum, which means more cases of 2 or 3 bit errors in decoded output
- Vast majority of 'impulsive events' observed in the lab focus on one TP
  - Thus, safer to pack 3 consecutive 'MSBs' of 3 DSQ symbols from the same TP into one RS symbol
- T=2 RS over GF(2^9) can correct all bursts (impulses here) of length 7 Bauds
  - Over GF(2^8) the guarantee is only length 5 Buads

Motivation 3 for 9-bit RS symbols and packing

- Supports guaranteed increased distance for ‘detection’ as desired, as an alternative to using 8-bit CRC
  - Details of mapping (ordering) of LDPC bits into RS-FEC are TBD (not too complex, but again slightly irregular)
- This would be in addition to RX using LDPC syndrome = {0} to allow data release

Other Alternate Options for 9-bit RS symbols

- The transcoding proposal of ***lo_3bq_01_0514.pdf*** can be used with RS FEC w/ 9-bit symbols in at least two ways;
  - Drop AUX bit (compatibility issues?), leaving 59-14= 45 bits for T=2 D=3 RS FEC
  - Keep 1 AUX bit, implement RS T=2 and use remaining 8 bits for CRC

Summary and Conclusions

- A modification of ***lo_3bq_01_0514.pdf*** is proposed
  - Using RS symbols of 9 bits vs. former 8 bits
  - Increased guaranteed correction and detection
  - Including using T=2 and D=3 (3 symbol error detection) to cover ‘LDPC = LSB’ bits as an alternative to CRC
- Can work with the same proposed transcoding, or with ‘generalization’ from the 256/257 standard to include 128/129

Thank you