NOKIA Bell Labs

LDPC Code Length Reduction

R. Borkowski, R. Bonk, A. de Lind van Wijngaarden, L. Schmalen – Nokia Bell Labs
B. Powell – Nokia Fixed Networks CTO Group

IEEE P802.3ca 100G-EPON Task Force Meeting, Orlando, FL, November 2017

Introduction

- During the last several meetings, several LDPC and RS codes have been proposed. The preferred length is between 2 kB and 4 kB. Current proposals focus on length-18493 LDPC codes, e.g., in May 2017, an [11×74×256] LDPC code was proposed, which was shortened to an (18493,15677) LDPC code (see [laubach_3ca_1_0517]).
- In previous meetings, a preference was expressed for using the same FEC code for upstream and downstream (for symmetry, and to simplify implementation and testing).
- The burst quantization unit, neglecting any overhead, is limited to 18493/25 Gbit/s = 739.7 ns
- One contribution [laubach_3ca_1_0317], considered the aggregate throughput in the upstream as a function of the burst size (200-20,000 bytes payload), and concluded that 20Gb/s throughput could be achieved for bursts of 12,000 bytes or longer by shortening the (last) LDPC codeword.
- Further analysis is needed to determine the code performance and rate for shorter burst lengths:
  - Low-latency service requirements
  - Minimum Ethernet packet size
  - Efficient method of sending US ONT queue reports
- A rate of 0.848 is needed to support 2×10G (net rate) channels within a 25G channel
- FEC input BER 10^{-2}, post-FEC error floor $<10^{-12}$ [laubach_3ca_3_0317]

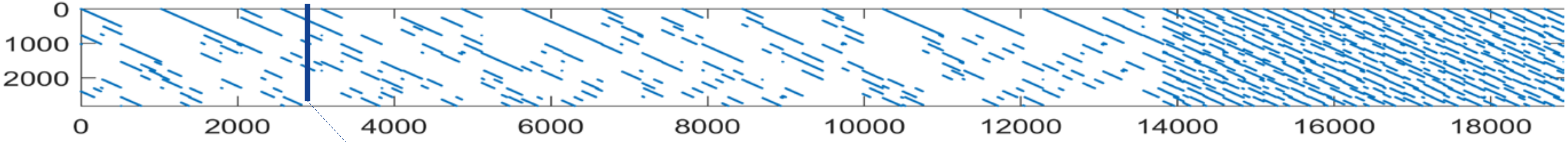
NOKIA Bell Labs

Shortening and Puncturing

- A code can be simultaneously shortened and punctured to maintain the same code rate while reducing transmitted codeword length
  - Shortening inserts 0's in place of some data bits, and these bits are not transmitted
  - Puncturing omits sending some of the codeword bits
- In a binary symmetric channel (BSC):
  - LDPC **shortening improves** performance because the log-likelihood ratio (LLR) of removed bits is set to a high value at the decoder, and they can be forced to remain 0's (i.e., shortened bits are certain)
  - LDPC **puncturing degrades** performance because the LLR of removed bits is set to a low value at the decoder (i.e., punctured bits are treated as erasures)

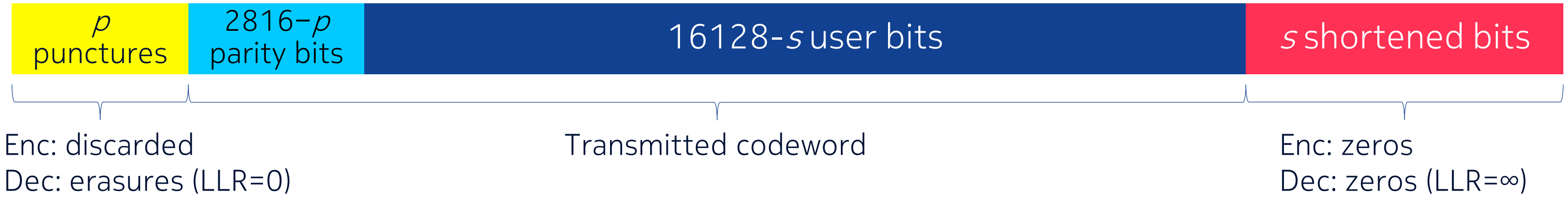
Shortening and Puncturing

Example - Proposed Broadcom LDPC matrix [11×74×256]



Natural codeword

Simultaneous shortening and puncturing. 100% codeword: $s = 451$, $p = 0$ [laubach_3ca_1_0517] (only shortening). Further length reduction obtained by increasing s and p at constant rate.



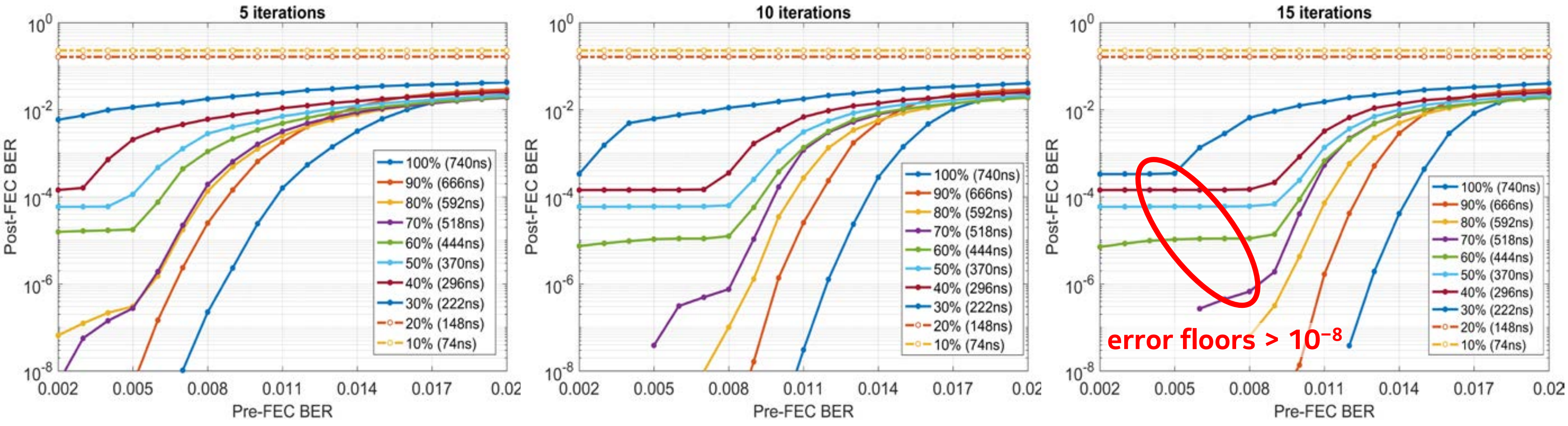
Shortening and Puncturing

Simulation Details

- Shortening from the rightmost matrix side by zeros → dense matrix part improves performance of the shortened code as certainty of shortened bits is shared across multiple equations.
- Puncturing from the leftmost matrix side → matrix already permuted so that consecutive erasures from the left will be distributed across multiple equations.
- Number of iterations: 5, 10 and 15
- BER calculated over data bits only
- Length reduced from 100%=739.7 ns (original code from laubach_3ca_1_0517) in steps of 1/10 while keeping constant rate of 0.848

%original length →		100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
Burst duration	*t*, ns	739,7	665,8	591,8	517,8	443,8	369,9	295,9	221,9	148,0	74,0
Transmitted bits	*n'*	18493	16644	14795	12946	11096	9247	7398	5548	3699	1850
Information bits	*k'*	15677	14110	12543	10975	9407	7839	6272	4704	3136	1569
Parity bits	*r'*	2816	2534	2252	1971	1689	1408	1126	844	563	281
Rate = k'/n'	*rate*	0.848	0.848	0.848	0.848	0.848	0.848	0.848	0.848	0.848	0.848
Shortened bits	*s*	451	2018	3585	5153	6721	8289	9856	11424	12992	14559
Punctured bits	*p*	0	282	564	845	1127	1408	1690	1972	2253	2535

Simulation Results



- Computer simulations are currently limited to BER 10^{-8}–10^{-9}
- Error floors appear for codewords shorter than or equal to 70% (517.8 ns) of the original size
- At 80% and 90% of the original length, the codeword floors below 10^{-8} but it is uncertain whether the net coding gain is sufficient

NOKIA Bell Labs

Conclusions

- Puncturing is non-trivial; we were not able to shorten the LDPC(18493,15677) code below 80% of the original length without loss of performance.
- If one solely relies on shortening, the error correction performance is maintained, but the code rate and achievable throughput become very low, not accounting for additional factors like laser switch on/off time and sync time. Interleaving complicates shortening.
- If one avoids short bursts by waiting for more data, this significantly increases latency and it may introduce additional jitter (data may need to wait in the buffer before a sufficient amount of user data for one codeword is available).
- It is believed that RS codes are more well behaved when shortened, as it is easier to adjust the number of parity symbols for a given number of information symbols; one can thus operate at a higher code rate and ensure the avoidance of an error floor for shorter code lengths.