

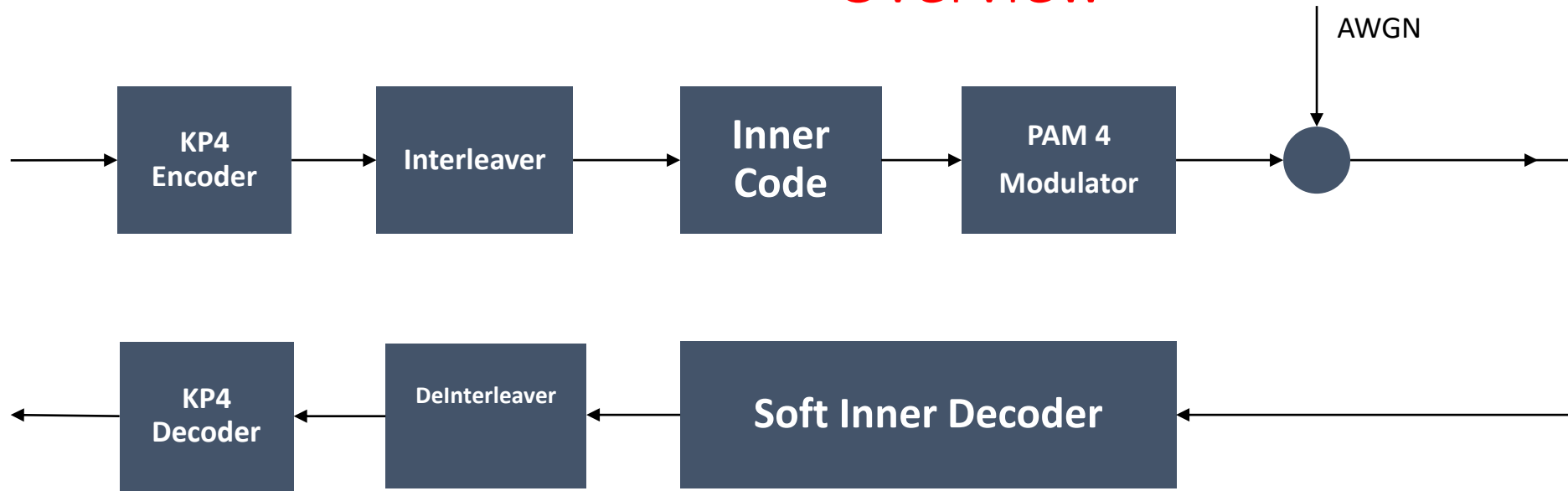
Soft Inner Hamming Net Coding Gain vs. Interleaver Latency

Will Bliss, Broadcom
802.3df May 17, 2022

Overview

- Strong interest in FEC systems with greater net coding gain than KP4 alone
 - A number of Inner Codes have been suggested
 - All such suggestions call out some level of KP4 codeword interleaving (needed for performance)
 - This presentation looks in detail at Latency vs net coding gain of a (128,120) extended Hamming code with Soft decoding
 - These specific code results will apply generally to all the suggested Inner codes
- Outline
 - Simple model w/ AWGN
 - Definition of Frame Loss Ratio, and resulting KP4 FEC 'failure to correct' requirement
 - KP4 FEC failure rate vs BER_{in}
 - KP4 FEC failure rate vs SNR w/ 'rate penalty' for AWGN channel
 - Net Coding Gain in dB vs Latency of interleaving
 - Why is such a long interleaver needed for maximum Coding Gain?
 - Should we adopt an Inner Code?
 - Issues If we do adopt an Inner Code

Overview

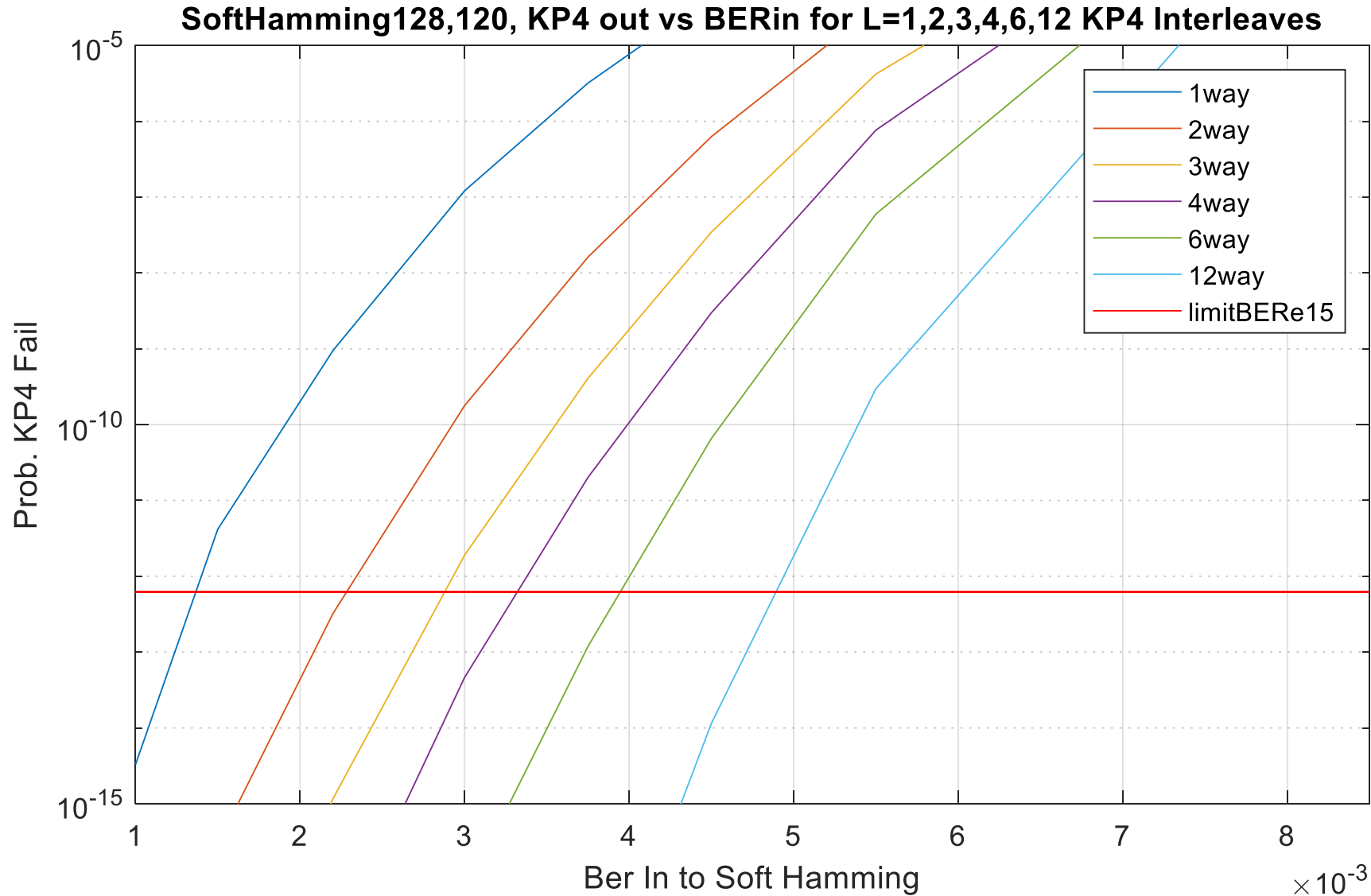


- The KP4 code is Reed Solomon (544,514) T=15 over $GF(2^{10})$, not interleaved
- The inner code used is (128,120) extended Hamming code
- The 'interleaver' will be varied from 'none,' up to 12-way KP4 codeword interleaving
 - E.g., for the latter, Each Hamming payload of 120 bits will be split over 12 different KP4 codewords, etc.
- The 'soft inner decoder' used was 'maximum likelihood', meaning it picks the Hamming codeword with the lowest total squared error
 - Other work has shown that practical 'Chase Decoding' based algorithms can perform very close to ML

Ethernet 'BER' Performance Definition

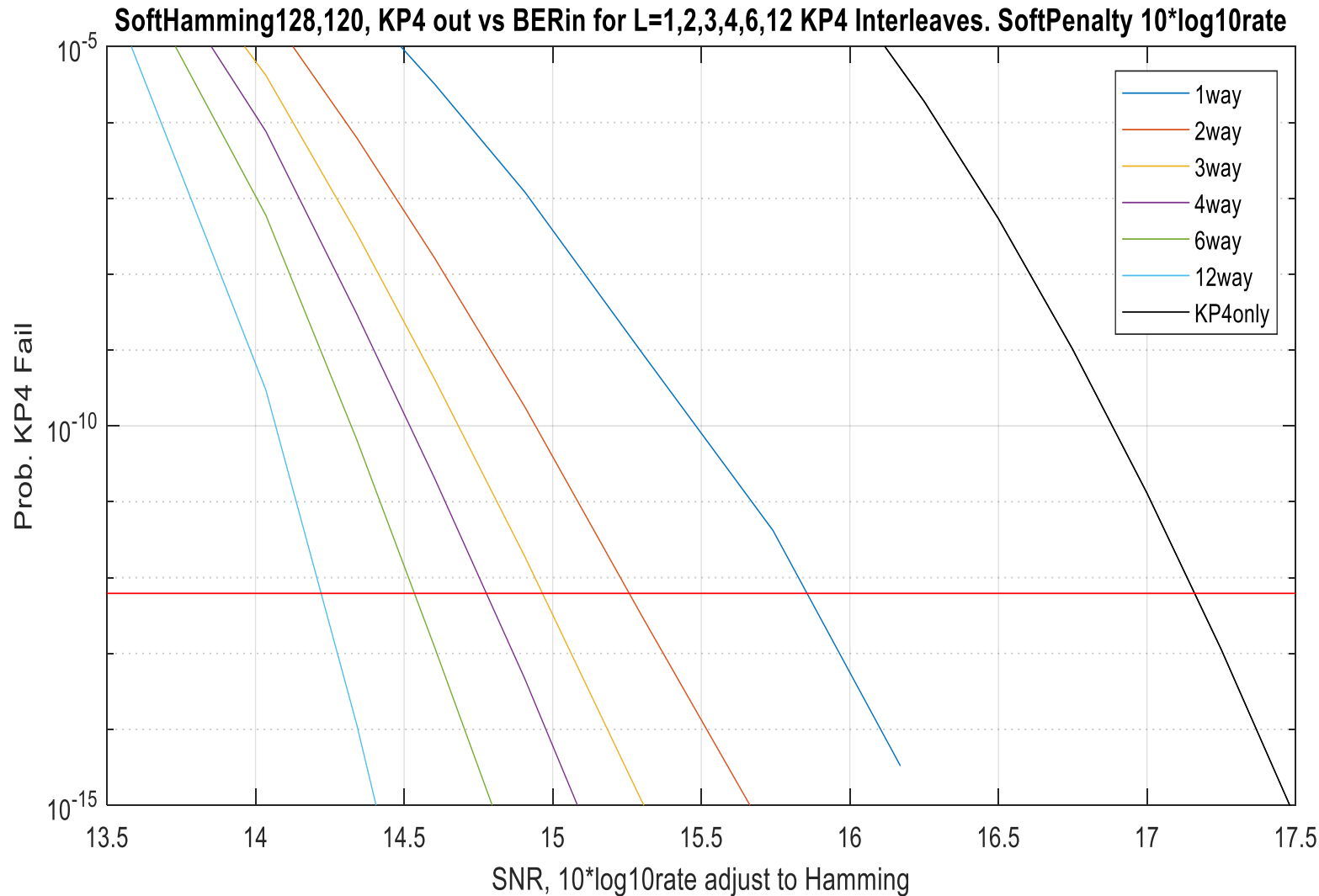
- The 802.3 Ethernet performance specification is commonly given in 'Bit Error Ratio' (BER), but our requirements have been defined as Frame Loss Ratio (FLR) = Frames lost per Frames Transmitted
- The required $FLR \leq 620 * BER_random$
 - Where BER_random is the equivalent BER specification IF the bit errors were randomly distributed
 - E.g., for the recently used $BER_random = 1e-15$, the requirement is $FLR \leq 6.2e-13$
 - See "Error performance objective for 400GbE," Pete Anslow, Ciena IEEE 400 Gb/s Ethernet Study Group, York, September 2013, [anslow_400_01_0913.pdf](#)
- For systems with FEC
 - FEC words that are 'not-corrected' are marked with 'corrupt xcoder sync-header' such that the Frames are dropped
 - e.g., Clause CL119.2.5.3 for 200G/400G, Clause CL91.5.3.3 for 100G
 - The FEC failure rate = FEC words failed to Correct per FEC words transmitted is then just the average FLR
 - So our target for ' $BER_random = 1e-15$ ' → $FEC\ fail\ rate \leq 6.2e-13$
 - So the number of bit errors in an uncorrected FEC codeword (and not passed to the sink) is irrelevant under the Ethernet requirements

Performance; KP4 failure rate vs BERin, for different interleavers



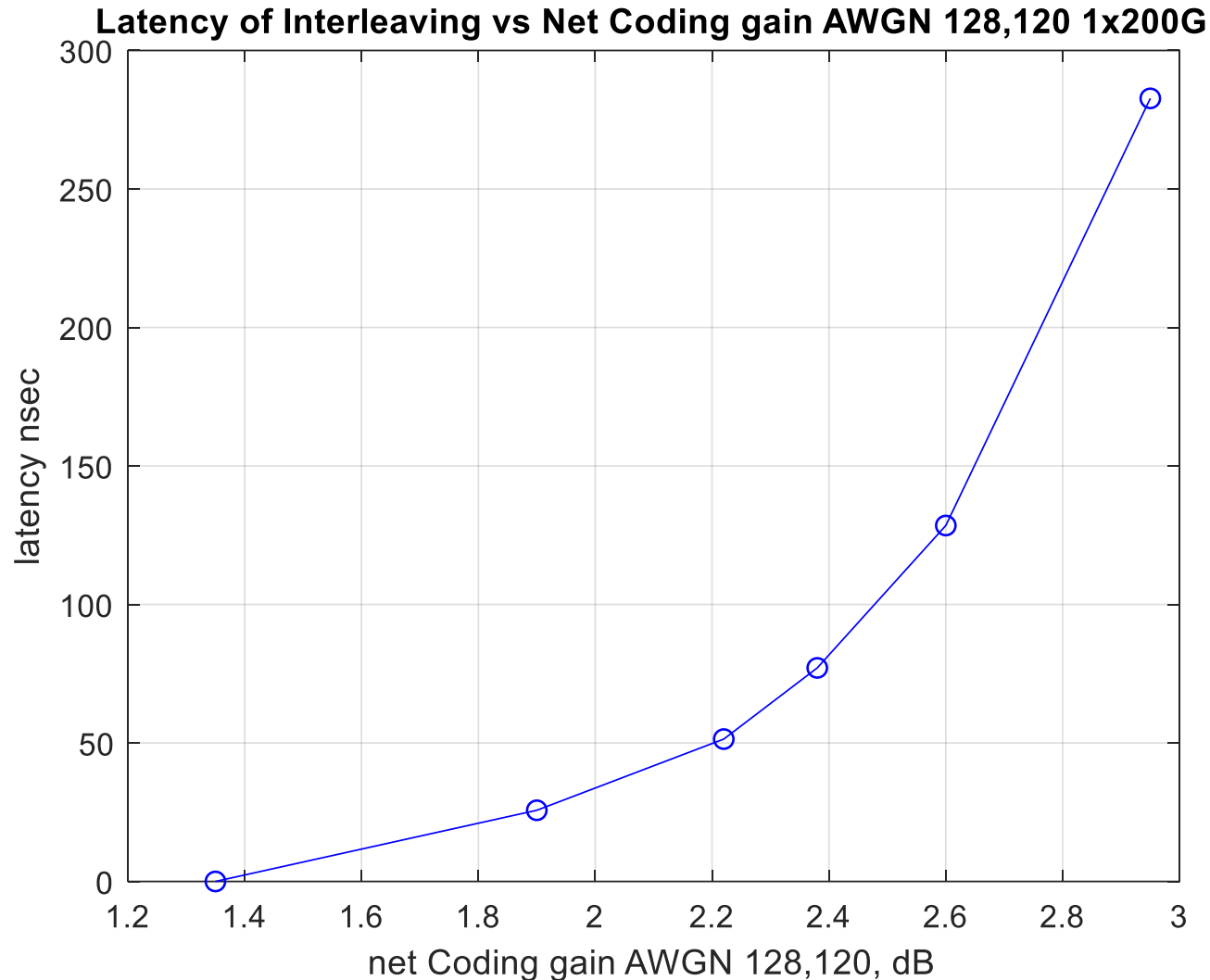
- A very large difference, of over 10 orders of magnitude FEC fail rate, between 'No Interleave' (1way) and 12 way
- This description obscures net coding gain because code rate isn't 'penalized'

Performance; KP4 failure rate vs SNR, for different interleavers;



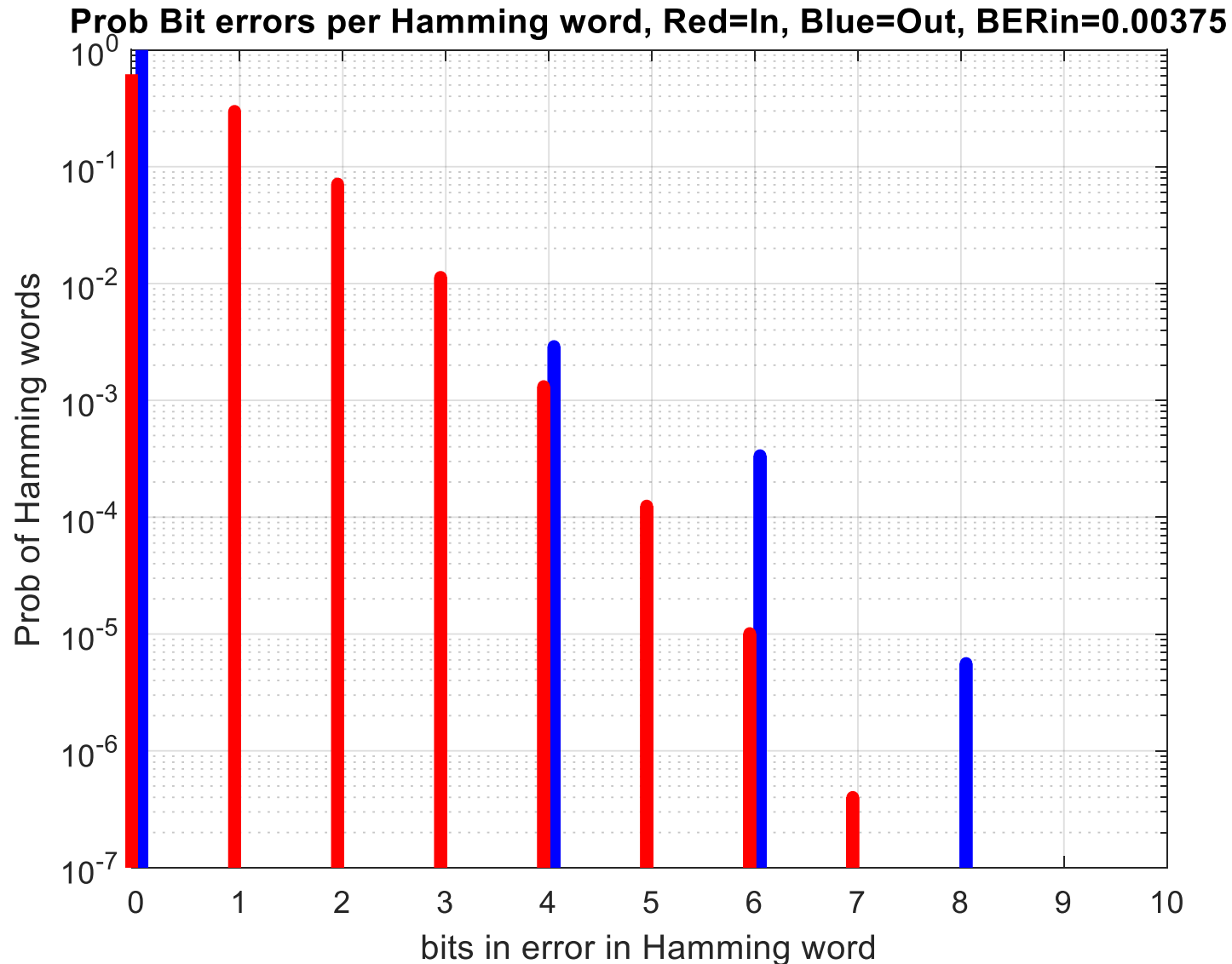
- The Hamming codes are penalized $10 \cdot \log_{10}(\text{rate}) = 0.28$ dB (result of AWGN model)
- Without interleaver, the 'net coding gain' is only ~ 1.35 dB
- A 12 way interleaver is optimal (fully randomizes the errors within a Hamming payload) and gives net coding gain of ~ 2.95 dB

Interleaver Extra Latency @ 200Gbps Breakout vs. Net Coding Gain



- The Theoretical minimum latency **added** to a 200Gbps physical lane to achieve the Interleaving points of
 - {1=none, 2,3,4,6, and 12}
- Doesn't include any extra implementation related latencies
- Doesn't include any other sources of latency
- Convolutional interleaver form is used, which has lower latency than 'block' forms
- 2-way interleave requires 25.7 nsec at 1x200Gbps and gives less than 2dB gain, (<65% of possible gain) leaving over 1dB gain unrealized
- 4-way interleave achieves 81% of possible gain, but the extra latency of 77.1 nsec at 200Gbps may be unacceptable to many users / applications
- Full gain requires 12 way interleaving with 282.7 nsec latency, which seems unacceptable for almost all applications

Why are Large Interleavers needed, even on the AWGN channel?



DISS, 002.501, 101ay, 2022

- Example for channel BER_{in}=3.75e-3
- The AWGN channel produces bit errors that are 'random'
- The probability of number of errors input to the Hamming block are binomially distributed
- The soft decoder corrects many / all Hamming words with a low number of errors
- But it leaves 'mis-corrections' and/or 'uncorrected' words with large number of bit errors
- Output words in error have at least 4 bits in error, demonstrating $d_{min}=4$ of the extended Hamming code
- NO Words with 5,7,9.. bit errors show the overall even parity of the extended Hamming code
- The distribution of errors in the Soft decoded output is very far from 'random'

Comments on Inner Codes and Interleaving of KP4 Codewords

- A number of different ‘Inner Codes’ have been described for 802.3df
 - Hamming of different lengths (rates)
 - BCH of $T \geq 2$ hard correction ability
 - Hard and Soft decoding of the various
 - E.g., [lu_3df_01_logic_220425](#)
- All of these codes have (correctly) been chosen to be significantly shorter than the KP4 code (5440 encoded bits)
- By their nature, all of these codes will exhibit similar loss functions for no KP4 interleaver and /or for too short of an interleaver
 - All of them have the behavior that they correct the ‘inner blocks’ with relatively fewer errors, and leave behind (and typically can make worse) blocks that started with relatively more errors
 - This means all these codes output errors are highly correlated
 - So to get full coding gain, these correlated blocks of errors must be spread over many KP4 codewords
 - AND we need the spreading random enough that the KP4 interleaves tend to be rather uniform
 - So all Inner codes of these types will need / benefit from KP4 codeword interleaving

Should 802.3df Select an Inner Code?

- Does this net Coding Gain vs. minimum latency increase ‘encourage’ us to pursue Inner Codes?
 - The ‘extra interleaver’ latency can’t be reduced for ‘breakout’ 1x200G, while it could for 4x200G by interleaving KP4 codewords across multiple lanes
 - But would anyone qualify a 4x200G mode when 1x200G fails?
- How much ‘net coding gain’ over KP4 will we need for a successful standard?
 - Will production optics be much improved over the first experiments and the first prototypes?
- How will performance and cost be influenced by real channel differences from the AWGN model?
 - How much higher rate loss than AWGN’s theoretical $10 \cdot \log_{10}(\text{Rate})$
 - IF the optical channel benefits significantly from DFE or MLSE, then errors will be correlated between adjacent PAM4 symbols. Correlated errors require more DSP
 - SOVA typically doubles MLSE cost (area and power)
 - A ‘Hamming Code Interleaver’ between the Inner encoder and the channel may be required such that correlated errors are spread to different Hamming words. E.g., a 3x64 PAM4 symbol block interleaver for this (128,120) code
- Note that ‘higher than KP4’ coding gain FEC without an inner code has been listed
 - E.g., RS(576,514) T=31 with modest ~1dB net coding gain over KP4 with ~20nsec increased latency from increased RS decoding steps [e.g., he_3df_01_220308](#)

Issues IF 802.3df Includes an Inner Code

- Are short copper segments also candidates for an Inner code? The same Inner Code? DAC or other longer copper channels? Etc.
 - How do we coordinate 'one architecture' for all these different channels?
- Where should the KP4 codeword interleaver be implemented?
 - If end-to-end FEC, then best if done at the 'ends', since the short copper segments will likely benefit from KP4 codeword interleaving
 - Less margin and longer error events for 200G copper segments motivates improvements
- Should we convert from legacy 'bit-Mux' to textbook RS 'symbol-Mux', and if yes, where should it be implemented?
 - 'Symbol Mux' == Five adjacent PAM4 symbols in the channel make one RS symbol of 10 bits
 - Again, Best done at the 'ends', since 200G copper segments will likely benefit from 'symbol-Mux'
 - But need to handle some backwards compatibility bit-muxing
- What Inner Code 'programmable options' might help the standard's success?
 - Inner code enabled or disabled, based on need
 - Programmable levels of KP4 interleaving (latency), based on need

Thank you