ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

100BASE-T1L PHY using PAM-3 8b6T Partial Response with Bounded Running Disparity and Optional Reed Solomon Forward Error Correction

Philip Curran
Brian Murray

Introduction

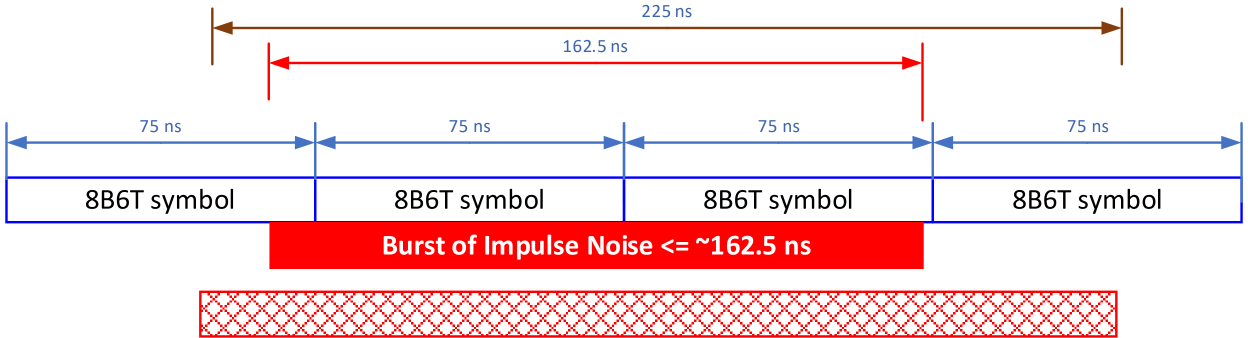
- This a proposal for a 100BASE-T1L PHY architecture using PAM-3 8b6T coding with partial response (PR) and bounded running disparity with two modes; a low latency mode and a burst error protection mode using RS FEC
- A low latency mode using a 16B/17B block code with PAM-3 and 8b6T at 80 MSym/s
- A burst error protection mode using a 64B/65B block code and a RS (128, 122, 3, 8) FEC code with PAM-3 and 8b6T at 80 MSym/s

Error Propagation

- Most BASE-T PHYs operating on difficult channels use Decision Feedback Equalization
- At long lengths, the 1st coefficient of the DFE feedback filter (FBF) becomes large
  - https://www.ieee802.org/3/dg/public/May_2024/murray_3dg_01_05132024.pdf
- Without mitigation this will cause error propagation
  - For PAM-3 with 1st DFE FBF coefficient equal to 1
    - if the previous decision is incorrect then then the probability of getting another error is 2/3
    - the probability of k consecutive errors is $(2/3)^k$
  - If FEC is used, multiple symbols in a code-word may be corrupted by a single error event
  - Severe error propagation may corrupt consecutive frames
- Could limit the 1st DFE FBF coefficient
  - This increases noise enhancement

Problem of Error Propagation with RS-FEC

- In 8b6T at 80 MSym/s each 8b6T symbol is 75 ns duration and thus 3 x 8b6T symbols is 225 ns duration
- A RS-FEC that can correct 3 8-bit symbols using a RS (128, 122, 3, 8) encoder, can correct for errors induced by an impulse noise burst of up to about 162.5 ns
  - Where 162.5 is 1T + 2 x 8b6T
- However, we must control error propagation or the 3 errors will propagate to > 3 errors



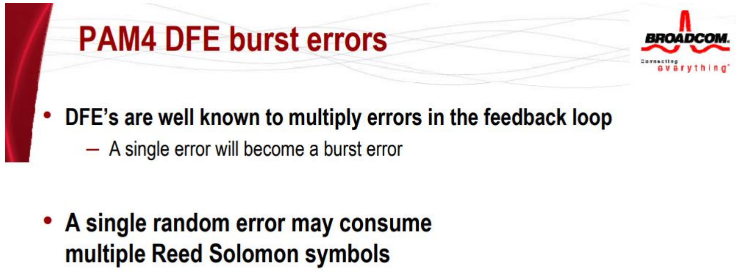
Burst of Impulse Noise of 225 ns can impact 4 8B6T symbols

- And the RS-FEC **cannot** correct a burst of 3 errors and an error due to AWGN in the same 9.6 us block
  - In fact, a single error due to AWGN can result in two 8b6T symbol errors
  - RS-FEC cannot double dip and correct burst errors & AWGN errors at the same time

Precoding to Mitigate Error Propagation

- Previous working groups have recognized the need to deal with error propagation when using RS FEC
  - https://www.ieee802.org/3/bj/public/sep11/parthasarathy_01_0911.pdf
  - https://www.ieee802.org/3/cd/public/May16/hegde_3cd_01a_0516.pdf
  - https://grouper.ieee.org/groups/802/3/ch/public/adhoc/souvignier_3ch_01_0818.pdf
- These typically use a partial response (PR) target for the equalizer

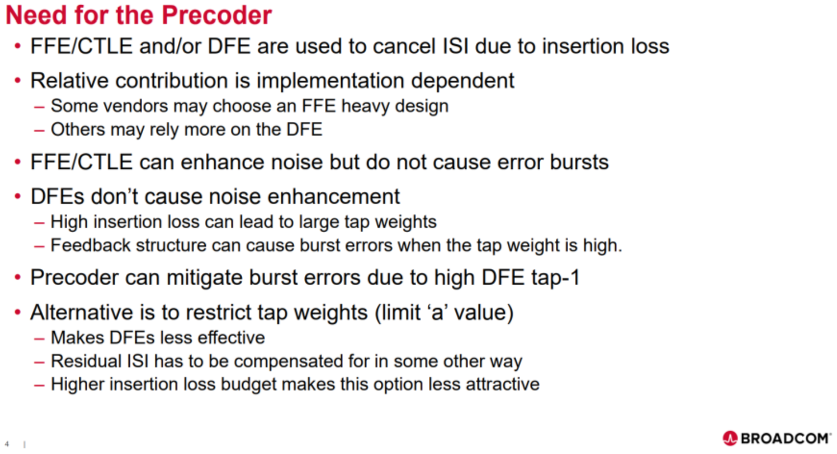
parthasarathy_01_0911 – slide 2



PAM4 DFE burst errors

- DFE's are well known to multiply errors in the feedback loop
  - A single error will become a burst error
- A single random error may consume multiple Reed Solomon symbols

hegde_3cd_01a_0516 – slide 4

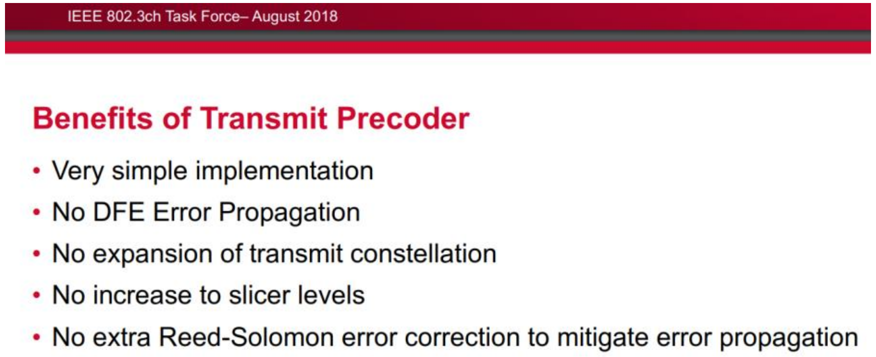


Need for the Precoder

- FFE/CTLE and/or DFE are used to cancel ISI due to insertion loss
- Relative contribution is implementation dependent
  - Some vendors may choose an FFE heavy design
  - Others may rely more on the DFE
- FFE/CTLE can enhance noise but do not cause error bursts
- DFEs don't cause noise enhancement
  - High insertion loss can lead to large tap weights
  - Feedback structure can cause burst errors when the tap weight is high.
- Precoder can mitigate burst errors due to high DFE tap-1
- Alternative is to restrict tap weights (limit 'a' value)
  - Makes DFEs less effective
  - Residual ISI has to be compensated for in some other way
  - Higher insertion loss budget makes this option less attractive

4 |

souvignier_3ch_01_0818– slide 2



IEEE 802.3ch Task Force– August 2018

Benefits of Transmit Precoder

- Very simple implementation
- No DFE Error Propagation
- No expansion of transmit constellation
- No increase to slicer levels
- No extra Reed-Solomon error correction to mitigate error propagation

Partial Response with Bounded Running Disparity

- curran_3dg_01_05132024 described the details of the PAM-3 8b6T code, its construction and properties
  - The 8b6T code with balanced running disparity is constructed using 6-tuples with zero disparity and pairs of 6-tuples, one with positive and one with negative disparity
  - For each 6-tuple there are two associated partial response (PR) sequences
    - One for each of the cases where the preceding transmitted ternary value is assumed to be -1 or +1
    - Each of these PR sequences is a 6-tuple and the elements are quinary
  - There is a memoryless one-to-one inverse mapping from the PR sequences to the 6-tuples
    - Thus, errors do not propagate
  - There is a $\sqrt{2}$ minimum distance between any two PR sequences to allow an effective SNR gain of up to 3 dB

8b6T Code-groups

- The proposal curran_3dg_02_05132024 reserved some 6-tuples for control codes and for idle codes to achieve the lowest possible latency
  - In this proposal an 8N/8N+1 block code is used to encode data and idle and thus only 256 6-tuples are required for the 8-bit data values
- As there are too many code-groups to list them all here, the following file is provided

  ctl_data_code_groups_07242024.txt

  - This file has 256 lines. Each line has 7 columns
  - The first column is the 8-bit binary selection
  - The remaining 6 columns provide the ternary values for the code-group
- Each of the code-groups has non-negative disparity (NND)
  - For each code-group with positive disparity we can create a corresponding code-group with negative disparity by negating the positive code-group element wise

Application to 8b6T at 80 MBaud

- The following is the list of non-negative disparity (NND) 6-tuples
  - Total of 256 NND 6-tuples
  - 86 6-tuples with disparity 0
  - 81 6-tuples with disparity 1
  - 60 6-tuples with disparity 2
  - 29 6-tuples with disparity 3
  - Note we have removed the following two 6-tuples with disparity 0,as these increase the running disparity bound

```
(+1, +1, +1, -1, -1, -1)
(-1, -1, -1, +1, +1, +1)
```

- We associate each of the 256 8-bit values from the encoder with one of these NND 6-tuples

Control of Running Disparity

- The running disparity (RD) at the transmitter is controlled as follows
  - If RD is positive, and the 8-bit value from the encoder is associated with a 6-tuple with positive disparity, then the 6-tuple should be negated before transmission
  - If RD is zero, and the 8-bit value from the encoder is associated with a 6-tuple with positive disparity, then a pseudo random Boolean value will determine whether to negate the 6-tuple before transmission
  - The pseudo random Boolean value will be determined by a variable derived from the scrambler
  - RD is recomputed after transmission of each 6-tuple

Running Disparity for 8b6T

- The 8b6T code consists of 425 6-tuples with disparities 0, ±1 , ±2 & ±3
  - 425 = 86 + 2 x (81 + 60 + 29)
- Using a pseudo random variable to select a positive or negative 6-tuple reduces the data correlation which is very important for adaptive systems
- The RD controls ensures that the maximum RD at the 6T symbol boundary is ±3 and the maximum RD within the 6T symbol is ±5
- In 10BASE-T1L using 4B3T the code has 3-tuples which also have disparities of 0, ±1 , ±2 & ±3
  - A scheme is used with a table lookup to choose a positive or negative 3-tuple depending on the value of RD and does not use a random variable, the maximum RD at the symbol boundary is ±3
  - However, this results in greater correlation in the data after RD control which is not good for an adaptive system
  - Using a pseudo random variable to select a positive or negative 3-tuple with 4B3T would result in a maximum RD at the 3T symbol boundary of ±3, which is the same as 8b6T

Running Disparity Checking

- It is proposed not to support RD checking in the receiver
  - A disturbance such as an EFT event would be likely to cause an error in such an RD check
  - The RD checking process would take time to resynchronize after such an error
  - This would result in error propagation
- We do not see a benefit in including RD checking in the receiver
  - Such checking is not required to detect frame errors
  - The benefit of RD control is on the transmit side

Baseline SNR Assumptions

- We would like to establish a baseline SNR requirement for 8b6T with PR equalization
  - For this we assume that we slice the equalizer outputs one by one using a quinary slicer
    - We will look at improvements later using Maximum Likelihood detection
  - To compare our PR equalizer to a conventional DFE we need a common reference for expressing noise power
    - With the proposed encoding we see a transmitted ternary symbol power of 0.7057
    - We will use this level as the reference for expressing the noise power in dB
  - We assume that the system noise can be represented by an AWGN signal, ω , adding at the output of the equalizer
  - In Ethernet, the bit error ratio is normally inferred from the frame error ratio. We use the following equation for the probability of a bit error with 8b6T line coding

$$P_b^e = \frac{P_f^e}{N_b} \cong \frac{N_s}{N_b} P_s^e = 0.75 \times P_s^e$$

Here P_b^e and P_s^e represent the probabilities of a bit error and a symbol error respectively

Baseline SNR Requirement

- We would like to know the SNR requirement for 10^{-10} BER when using PR equalization
  - The probability of an error at the quinary slicer is as follows

$$P_s^e = \big(1 + P(0) + 2 \times P(+1)\big) \times P(\omega > 0.5) = \big(1 + P(0) + 2 \times P(+1)\big) \times 0.5 \times \mathrm{erfc}\left(\frac{1}{2\sqrt{2}\sigma}\right)$$

  - We require the noise power to be below about -20.7 dB with reference to the ternary symbol power

SNR Requirement without PR

- Let us also calculate the SNR requirement for 10^{-10} BER when using a conventional DFE
  - We use the same set of ternary 6-tuples but this time the output of the equalizer is a ternary value observed with additive noise
  - We assume that there is no error propagation. This assumption will not generally be true at long lengths.
  - The probability of an error at the ternary slicer is as follows

$$P_s^e = \left(1 + P(0)\right) \times P(\omega > 0.5) = \left(1 + P(0)\right) \times 0.5 \ \times \mathrm{erfc}\left(\frac{1}{2\sqrt{2}\sigma}\right)$$

  - We require the noise power to be below about -20.6 dB with reference to the ternary symbol power
  - There is no real difference between the acceptable noise levels at the slicer in the cases with and without PR equalization
    - The argument of the complementary error function is the same in each case and will hugely dominate the calculation of the probability of an error

Mean Square Error for the DFE and for Partial Response

- A DFE estimates the received symbol from the values sampled at the ADC and the previous decisions where yd_n are the decisions

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k} - \sum_{k=2}^{M} a_k yd_{n-k} - a_1 yd_{n-1}$$

  - Note, I have shown the 1st feedback term separately
  - The MSE (σ^2) at the slicer is calculated from the average of the error at the slicer from a large number of values (L)

$$MSE = \frac{1}{L}\sum_{n=0}^{L-1}(y_n - yd_n)^2$$

$$\sigma = \sqrt{MSE}$$

- Now a DFE in a partial response system assumes a 1 + D channel and does not have a 1st tap

$$w_n = \sum_{k=0}^{N-1} h_k x_{n-k} - \sum_{k=2}^{M} a_k yd_{n-k} \qquad \text{and also} \qquad w_n = y_n + yd_{n-1}$$

$$MSE = \frac{1}{L}\sum_{n=0}^{L-1}(w_n - wd_n)^2$$

$$MSE = \frac{1}{L}\sum_{n=0}^{L-1}\left((y_n + yd_{n-1}) - (yd_n + yd_{n-1})\right)^2$$

$$MSE = \frac{1}{L}\sum_{n=0}^{L-1}(y_n - yd_n)^2$$

- With a 1 + D channel the MSE at the output of a conventional DFE **is identical** to the MSE at the output of a partial response DFE

Maximum Likelihood Detection

- The receiver may use a maximum likelihood (ML) detector operating on 6 PR samples at a time to determine the most likely ternary 6-tuple
  - The detector can take advantage of the fact that the last ternary value in each 6-tuple is non-zero
  - The ML detector may be reinitialized every 6 cycles of the symbol clock so that errors do not propagate from the detection of one 6-tuple to the next
  - As ML detection may be formulated as a shortest-path problem, the complexity is limited
  - Any ML detector operates by computing a measure of likelihood

Maximum Likelihood Detection and Effective SNR Gain

- A simulation has been run using the proposed 8b6T line code and an ideal 1 + D channel with AWGN and MLD
  - In this simulation the ML detector was reinitialized at the start of each 6-tuple and a decision was made at the end of each 6-tuple
  - There is no error propagation, and the latency is low
  - The simulation was run at an SNR of about 17.0 dB until at least 100 ML errors were observed and in this time over 40,000 Ternary Hard Decision errors were observed
    - At least 100 ML errors was deemed sufficient to estimate the effective SNR gain
    - Obviously, this was a very long simulations, a number of seconds of real time data
- An effective SNR gain of about 2.8 dB has been observed when simulating the proposed 8b6T line code

Low Latency Mode 100BASE-T1L PHY

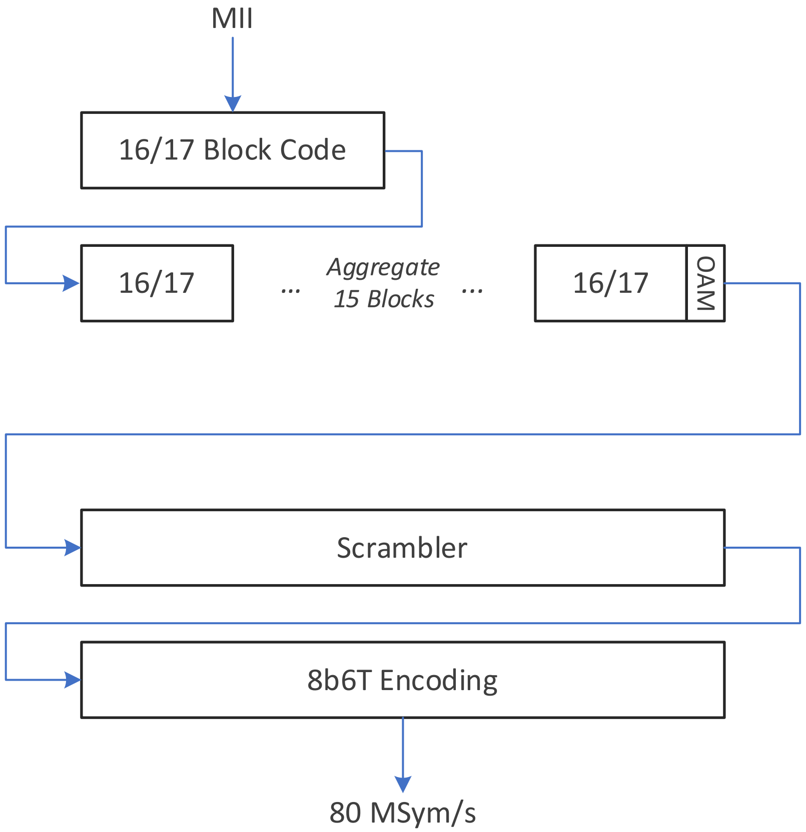
- Use PAM-3 modulation with an 8b6T code at 80 MSym/s
  - Use an 8N/8N+1 block code with N = 2: hence a 16B/17B block code
  - With L = 15 and a data block size of 15 x 16 = 240 bits
  - With L = 15 and 1 x OAM bit we have 15 x 17 + 1 = 256 bits after the block code
  - Transmitted as 16 x 8b6T symbols
  - The symbol rate is (256/240) x (6/8) x 100 = 80 MSym/s
- Tx + Rx latency of < 1 μs in this mode, implementation dependent

Burst Error Protection Mode 100BASE-T1L PHY

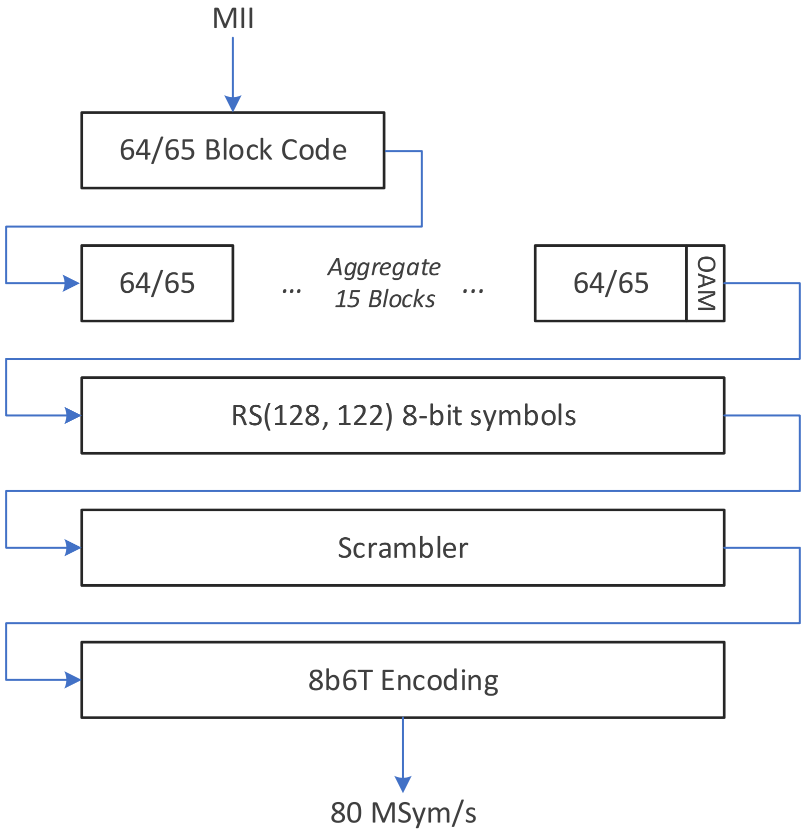
- Use an RS(128,122, 3, 8) code with a block size of 9.6 μs and 225 ns of burst error protection
  - This is the same RS FEC proposed in Tingting_3dg_14_05_2024
- Use PAM-3 modulation with an 8b6T code at 80 MSym/s
  - Use an 8N/8N+1 block code with N = 8: hence a 64B/65B block code
  - Use a Reed Solomon FEC code with a Galois Field of 8 and RS(128, 122, 3, 8)
  - With 3 correctable symbols for 225 ns of burst error protection
  - With L = 15 and a data block size of 15 x 64 = 960 bits and thus a block length of 9.6 μs
  - With L = 15 and 1 x OAM bit we have 15 x 65 + 1 = 122 x 8 = 976 bits after the block code
  - And a total RS block size of 128 x 8 = 1024 bits
  - Transmitted as 128 x 8b6T symbols
  - The symbol rate is (1024/960) x (6/8) x 100 = 80 MSym/s
- Tx + Rx latency of ~12 to 15 μs in this mode, implementation dependent

Block Diagram of Transmit Path for each Mode

Low Latency Mode



Burst Error Protection Mode

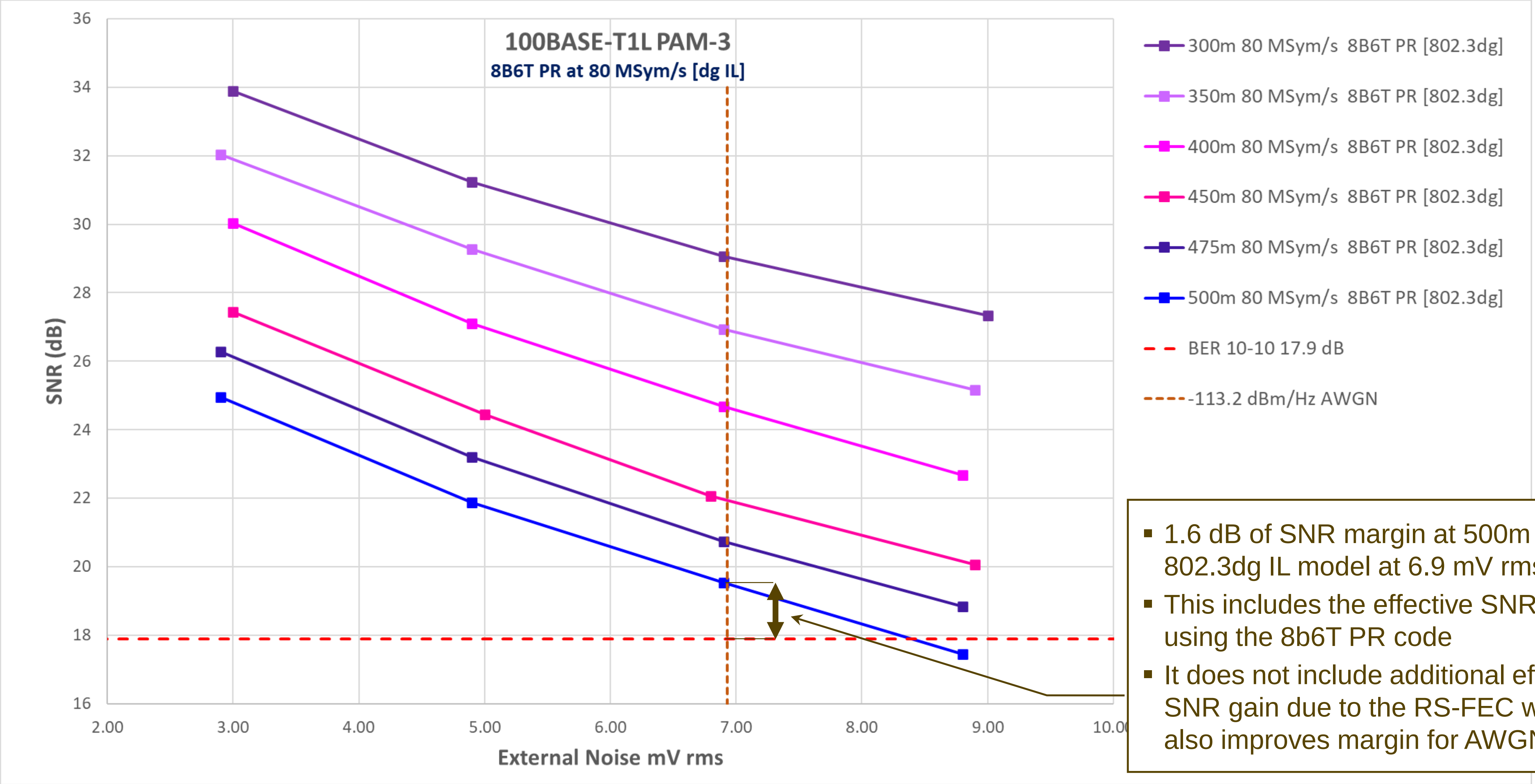


100BASE-T1L SNR vs Ext Noise at 80 MSym/s

- Plot SNR versus external Gaussian noise
  - For values of 3, 5, 7 and 9 mV rms
  - For Insertion Loss model proposed for 802.3dg
    - For cable lengths 300, 350, 400, 450, 475 and 500 m derived from a scaled IL model
    - Scaling relative to the Insertion Loss model will be worse than a typical cable model
  - At 2.4V transmit level
  - After 1536K symbols of start-up, idle and data (~ 20000 μs)
    - Enter data after 300K symbols
  - The plots are really MSE expressed as SNR calculated relative to a fixed transmit symbol power
- The decision after the partial Response DFE is done using 3 methods
  - A conventional Ternary Hard decision to the 6-tuple, 1T at a time
  - A Quinary Hard decision and inverse mapping to the 6-tuple
  - A Maximum Likelihood Detection to select the 6-tuple

100BASE-T1L SNR vs Ext Noise – PAM-3 dg IL Model

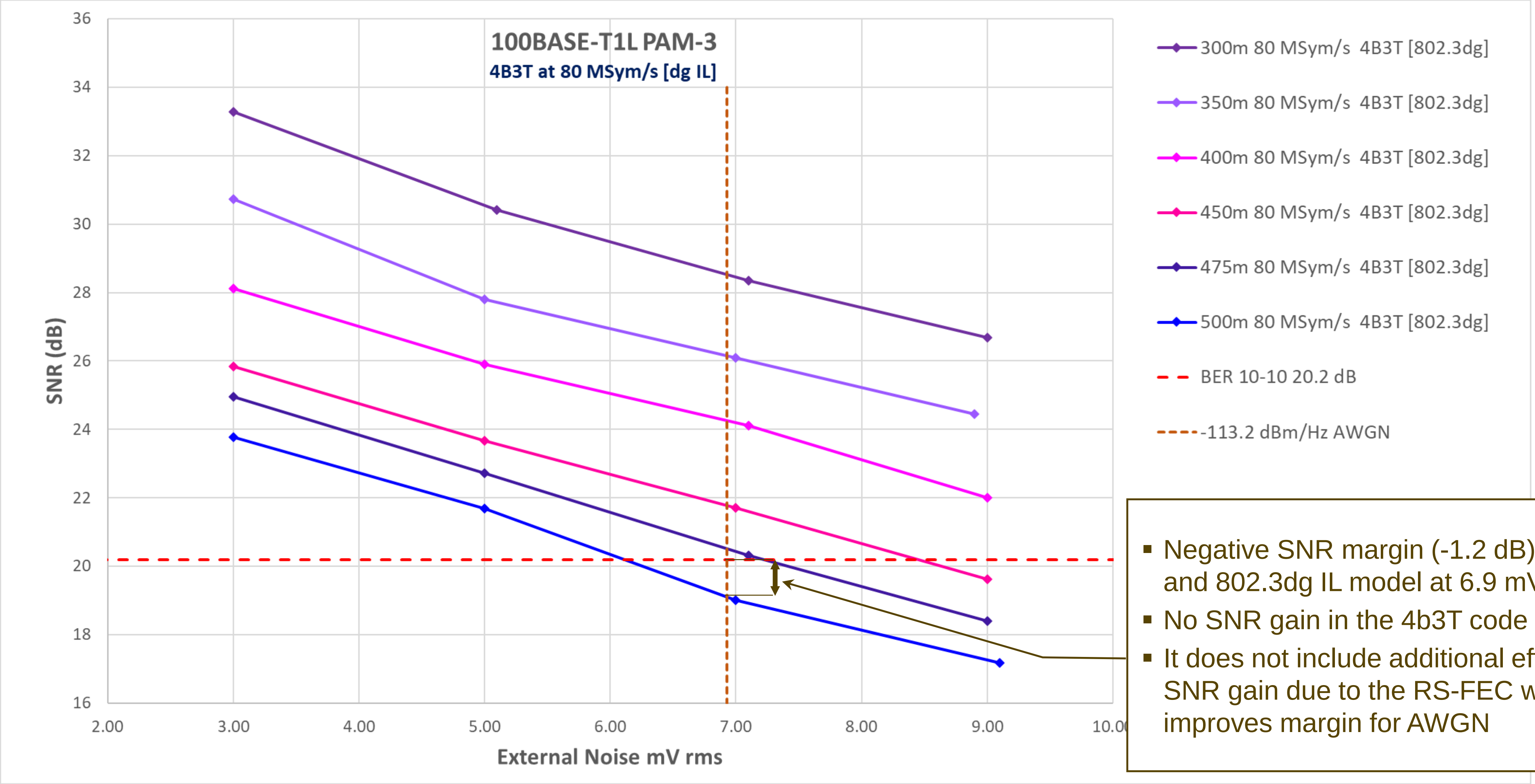
100BASE-T1L 80 MSym/s 8b6T PR: SNR versus External Noise – 2.4V Tx Amplitude



- 1.6 dB of SNR margin at 500m and 802.3dg IL model at 6.9 mV rms
- This includes the effective SNR gain using the 8b6T PR code
- It does not include additional effective SNR gain due to the RS-FEC which also improves margin for AWGN

100BASE-T1L SNR vs Ext Noise – PAM-3 dg IL Model

100BASE-T1L 80 MSym/s 4b3T: SNR versus External Noise – 2.4V Tx Amplitude



- Negative SNR margin (-1.2 dB) at 500m and 802.3dg IL model at 6.9 mV rms
- No SNR gain in the 4b3T code
- It does not include additional effective SNR gain due to the RS-FEC which also improves margin for AWGN

8b6T Versus 4B3T

- 8b6T coding with a 1 + D channel can be decoded using any of the following 3 methods without any change on the transmit side
  - A conventional Ternary Hard decision to the 6-tuple, 1T at a time
  - A Quinary Hard decision and inverse mapping to the 6-tuple
  - A Maximum Likelihood Detection to select the 6-tuple
- This is entirely up to the receiver designer
- 4B3T coding only supports the first option, a conventional Ternary Hard decision and suffers from error propagation
  - This is ignoring many years of established precedent of Reed Solomon codes with difficult channels
  - In 4b3T all 26 3-tuples are separated by a distance of 1, and there is no means to achieve any effective SNR gain

Summary

- PAM-3 coding using 8b6T with partial response at 80 MSym/s meets the reach requirements of 500 m on the proposed link segment specifications with SNR margin and low latency
- PAM-3 8b6T with partial response has balanced running disparity to support intrinsic safety and lowest component cost for single-pair power over Ethernet (SPoE)
- PAM-3 8b6T with partial response has the advantage of up to 3 dB of effective SNR gain due to the $\sqrt{2}$ minimum distance between any two partial response sequences
- PAM-3 has the advantage of wider spacing of decision thresholds which gives the greatest immunity to impulse noise
- Operating at 80 MSym/s allows a single PHY to support a low latency mode and a higher latency mode with RS FEC for burst error protection
- PAM-3 8b6T with partial response mitigates error propagation which is particularily important for a RS FEC to ensure a burst of errors does not propagate for longer than the number of correctable errors in the RS block

Questions ?