ANALYSIS OF SER DEGRADATION PARAMETERS

(COMMENT #40)

Adee Ran

Intel Corp.

Supporters:

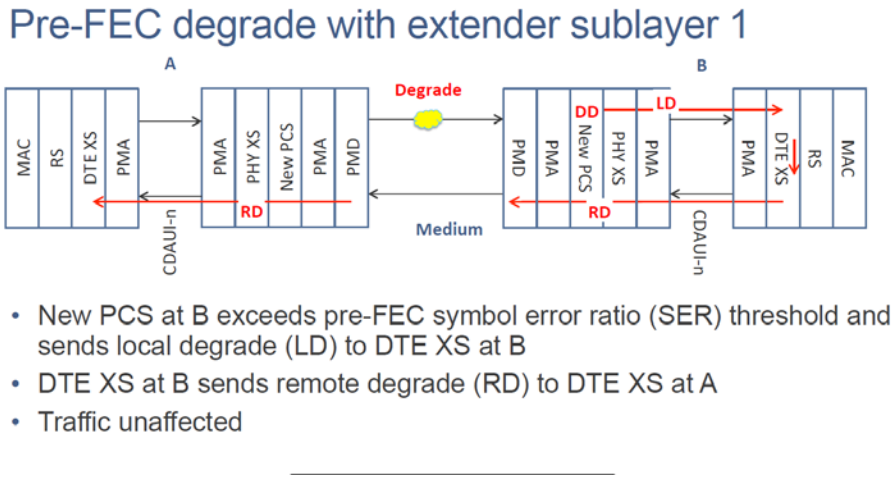
- Tongtong Wang, Huawei

Outline

- FEC SER degradation feature description
- Possible scenarios / use cases
- Analysis
- Further concerns

FEC SER degradation description

- Intent is to signal degradation condition to the link partner
- Described in several slides ofelt_3bs_01a_0516
- Example scenario shown
- The degradation condition is configurable with a combination of three 32-bit parameters, allowing great flexibility
- Which values make sense?



Possible scenarios / use cases

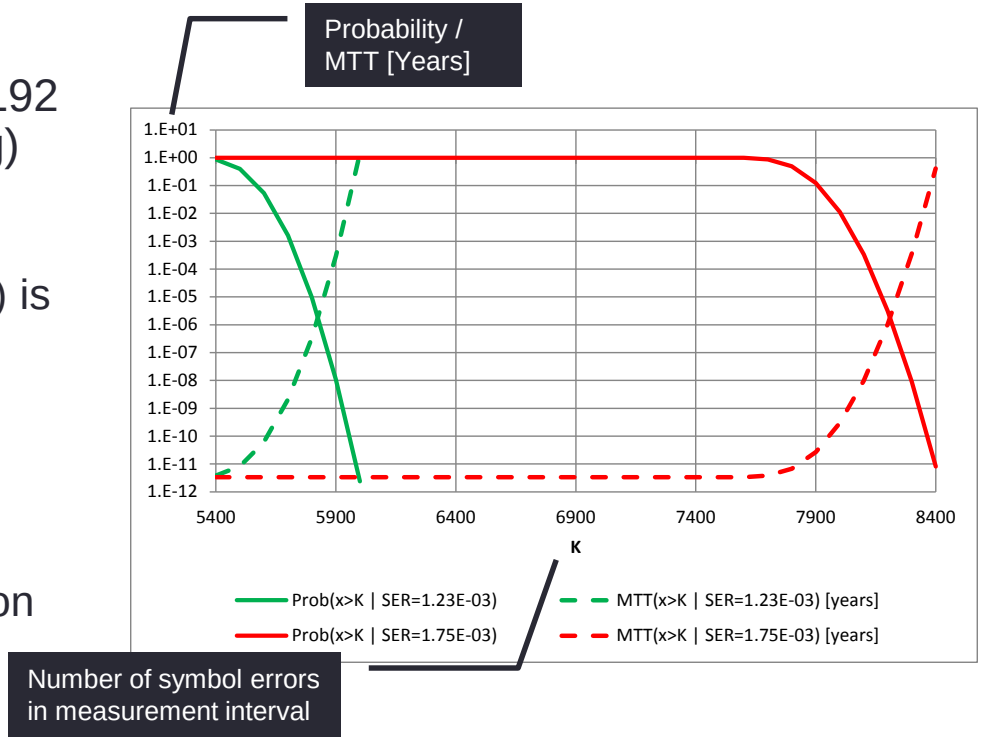
- To analyze the usability of this feature, let's consider as a "baseline":
  - A 400G link with stationary noise and small margins
  - Optical segment with BER=1.03e-4 (~0.24 dB margin to 2.4e-4)
  - Two C2C+C2M segments with total BER=2e-5
- And two scenarios
  1. A degradation of the optical SNR by 0.125 dB (reducing the margin)
  2. Non-stationary noise conditions: electrical segments create BER=2e-5 on average, but as 2e-4 in 10% of the time; cycle period is 1 μs

Baseline vs. scenario 1

- The baseline: Optical BER: 1e-4; Electrical BER: 2e-5; total BER 1.2e-4
- With no error propagation, SER is 1.23e-3, expected FLR is 3.75e-17; mean time to uncorrectable codeword (MTTUC) is 12 years ➔ seems OK
  - No MTTFPA issue if uncorrectable errors are marked
  - With error marking bypassed, this would lead to MTTFPA less than 6 billion years; the error monitoring would trigger errors constantly (not enough margin for bypassing error marking)
- We want to detect scenario 1 (optical SNR degradation of 0.125 dB)
  - This would bring total BER to 1.76e-4, SER to 1.75e-3, and FLR to 8.22e-15; MTTUC=20 days ➔ maybe unacceptable
- What parameters should be used?

Baseline vs. scenario 1 (#2)

- Let's use FEC_degraded_SER_interval = 8192 (as in bypass indication monitoring)
  - With SER=1.23e-3, the expected number of symbol errors in 8192 codewords (denoted <SE(8192)>) is ~5480
  - With SER=1.75e-3, we get <SE(8192)> ≈7802
  - Large expected numbers create steep curves for exceeding them
  - Significant difference in expectation enables good distinction (many orders of magnitude in MTTA)



Baseline vs. scenario 1 (#3)

- Based on the curves, we can choose thresholds between 6000 to 8000
  - For example, FEC_degraded_SER_assert_threshold=8000 and FEC_degraded_SER_deassert_threshold=6000
  - A large distance between thresholds creates hysteresis and prevent noisy alerts (not analyzed here)
- With assert threshold at 8000:
  - SER=1.23e-3 or lower would “never“ create (false) alert
  - SER=1.75e-3 or higher would cause immediate (true) alert

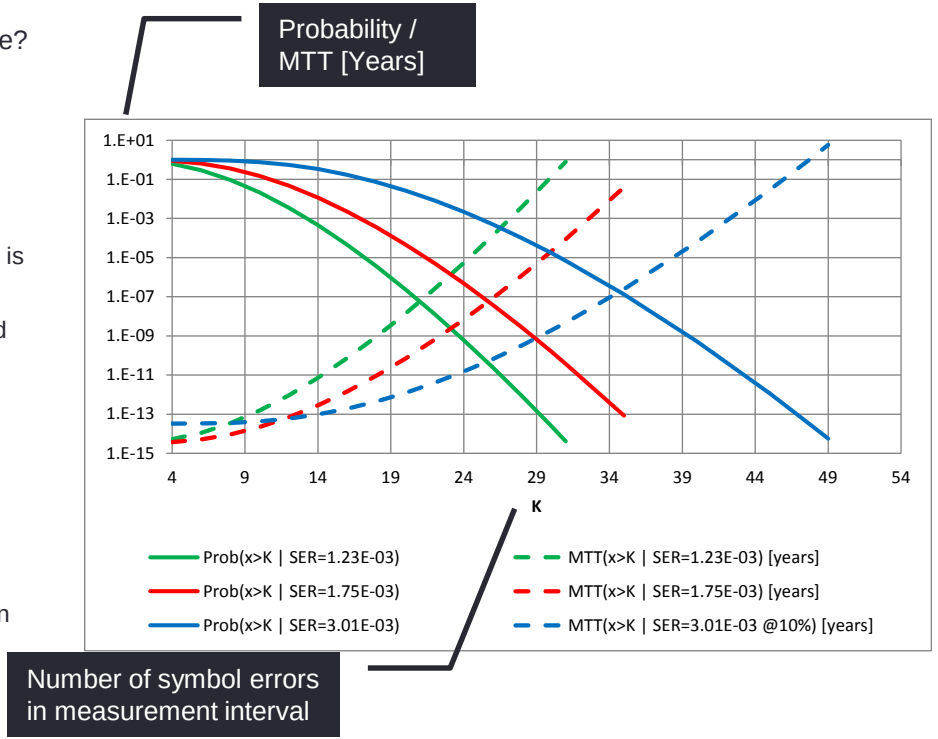
➔ Looks like a good combination of parameters

Baseline vs. scenario 2

- In scenario 2, we get electrical BER=2e-5 on average (as in the baseline), but as 2e-4 10% of the time and a negligible BER otherwise; cycle period is 1 μs
- Optical segment BER is stationary 1e-4; total BER is 1e-4 90% of the time, and 3e-4 10% of the time
- This results in
  - SER is 1.03e-3/3.01e-3 respectively, effective (weighted sum): 1.03e-3*0.9+3.01e-3*0.1=1.23e-3
  - The high-SER period is 100 ns, at 400G, codeword duration is 12.8 ns, so ~8 codewords fit into this period
  - FLR is 2.46e-18/2.55e-11 respectively, effective (weighted average): 2.46e-18*0.9+2.55e-11*0.1=2.55e-12
  - Mean time to uncorrectable codeword (MTTUC) is 1.7 hours (accounting for the 10% factor)
- Much worse than scenario 1
  - But average SER is the same as baseline; with the SER degradation parameters we “chose” for scenario 1, this would pass unnoticed

Baseline vs. scenario 2… and 1

- How can we detect scenario 2 with the SER degradation feature?
- FEC_degraded_SER_interval must be shorter than 80 codewords
  - Otherwise, the cycle is averaged out and we only see the average SER
- Let's try an interval of 8 codewords (minimum averaging):
  - With stationary noise creating SER=1.23e-3 (baseline), <SE(8)> is 5.35
  - In the high-SER period of scenario 2 we have SER=3.01e-3, and <SE(8)> changes to 13.1
  - Graphs are shallow for small numbers – no "clear cut"
- With assert threshold set to 34…
  - Scenario 2 would trigger an alert after ~3 seconds
  - Baseline would take more than 10 years create an alert (but not "never")
  - But Scenario 1 changes <SE(8)> to only 7.6 and would create an alert only every 3-4 days
- Lower thresholds would increase false alert rate in good scenarios, higher thresholds would miss scenario 1 completely



Summary of analysis

- Detecting different scenarios require different parameters…
  - Non-stationary noise requires very small intervals
  - Small intervals ➔ difference between healthy and bad links is not enough to place robust thresholds
  - Too small thresholds ➔ frequent false alerts, may result in a requirement of higher than necessary margins!
  - Too large thresholds ➔ late or missing alerts in some scenarios
- Real life is probably more complex than these simple scenarios…
  - Do we expect network engineers to examine multiple scenarios and analyze each one as done here, in order to set the parameters?
  - Maybe we need another approach
- Large intervals and thresholds, which 32-bit variables enable, are not required
  - Averaging many codewords could miss even long-cycle non-stationary conditions
  - Bursts of errors and even uncorrectable codewords may go unnoticed

Comment #40 – updated remedy

Cl **119** *SC* **119.2.5.3** *P* **162** *L* **17** # 40

Ran, Adee Intel

Comment Type **TR** *Comment Status* **D**

FEC_degraded_SER_interval, FEC_degraded_SER_assert_threshold and FEC_degraded_SER_deassert_threshold defined here do not have default values. In addition, all three are 32-bit long.

This enables a huge number of combinations of interval and threshold values. Only a small part of these combinations makes sense; for example, any threshold larger than 544*FEC_degraded_SER_interval would be inherently invalid. Additionally, both threshold values should be less than 15*FEC_degraded_SER_interval, otherwise the indication of degradation would only occur after at least one complete codeword in the period is uncorrectable; and the assert threshold should be higher than the deassert threshold.

There should be default values for all three variables, and a recommendation for setting them together.

Also, the parameters and scenarios should be analyzed to show the mean time to assert/deassert, and check whether this feature is useful or not. I am planning a presentation for that.

SuggestedRemedy

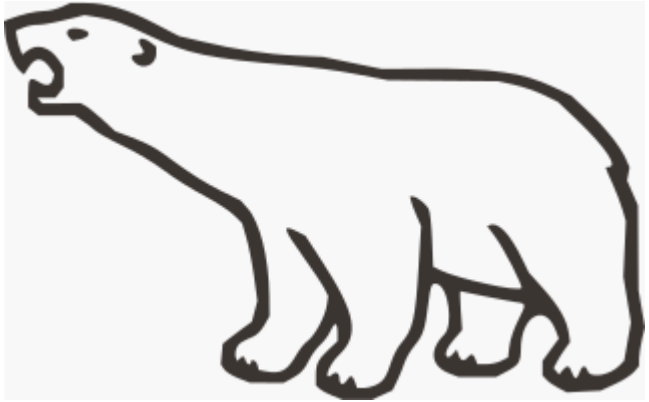
Specify default values as follows:
- FEC_degraded_SER_interval: default 8192 (as when indication is bypass)
- FEC_degraded_SER_assert_threshold: default 5560 (MTTFPA or uncorrectable codeword concern).
- FEC_degraded_SER_deassert_threshold: default 5000 (very healthy link)

Add text to indicate that unless the threshold values are set such that the assert threshold is higher than the deassert threshold, the behavior is unspecified (or degradation always asserted - see other comment)

Add as a note (informative) that in typical use, both values should be lower than the interval value.

- Based on the analysis here, the threshold values in the suggested remedy seem to require higher margins (would trigger an alert for links with very long MTTUC)
- Instead, it is suggested to use default values suitable for identifying degradation in stationary-noise conditions (e.g. baseline vs. scenario 1):
  - FEC_degraded_SER_interval: default 8192
  - FEC_degraded_SER_assert_threshold: default 8000
  - FEC_degraded_SER_deassert_threshold: default 6000
- The rest of the suggested remedy still holds.
- In addition, consider changing all 3 parameter definitions to a single MDIO register (16 bit) each.

What kind of BER do we want to catch?



QUESTIONS/COMMENTS?

Thank you