LINK HEALTH REPORTING

(COMMENT #44)

Adee Ran

Intel Corp.

Supporters:

- Tongtong Wang, Huawei

Outline

- Motivation
- Problems with the current approach
- Proposed alternative
- Pros and cons

Motivation

- Clause 91 defines error counters, but they are not reported to partner
  - May be possible to access by network management protocol, but does not immediately alert
- Quotes from ofelt_3bs_01a_0116, maki_3bs_01a_1115:
  - “Pre-FEC BER can show link health before packet errors are even seen on the link”
  - Possible reaction: “pre-emptively move traffic away from a link”
  - “Adding these features to the standard allows for interoperability and a consistent feature set”
- Essentially the information that network management needs is “Mean time to uncorrectable codeword (MTTUC) is too small”

Problems with the current approach

- Defining “degradation” in terms of SER
  - Works well when error statistics are stationary
  - May miss short periods with more errors than the average
  - May not predict “MTTUC too small”
- Setting the threshold values correctly
  - Depends on what kind of BER you want to catch
  - Should be done on each link, or with XS, each segment of the link
- Alert is binary
  - Exceeding a threshold is a random event, may happen at any link
  - “SER degradation” may be asserted and de-asserted “randomly”
  - In a large network this may happen much more often than actual uncorrectable codewords
  - When does it indicate a real problem?

What do you prefer?

Proposed alternative

- An uncorrectable codeword is one that has more than t=15 symbol errors
  - Let's denote the event of having exactly k symbol errors in a codeword as "Ek", k or more as "Ek+"
  - Probability of a specific codeword to be uncorrectable is p(E16+)
- Assumption: codewords with smaller numbers of errors are more likely
  - So p(E16+)<p(E15)<p(E14) … <p(E1)
  - This holds for most channel models (error statistics), including non-stationary and bursty channels
  - Exception: at high BER (allowed in 802.3bs), it may happen that p(E1)>p(E0)
- The number of corrected symbol errors in a codeword is readily available from the RS decoder
  - The SER degradation feature uses it
  - We can easily track each of the events above in separate counters, C1 to C15
  - When read periodically, these counters can be use to assess the probability of each event up to p(E15), and extrapolate to p(E16+)

Proposed alternative (cont.)

- If we want to report back error statistics in-band, we have the alignment markers
  - In 400G: 16 PCS lanes, 2056 bits (8 × 257) once every 8192 codewords
  - In 200G: 8 PCS lanes, 1028 bits (4 × 257) once every 4096 codewords
- The AMs can include the event counts measured since previous AM
  - Current AM definition includes six unique octets per PCS lane and pad bits
  - We can utilize some of these to send the k-symbol-error event counters
- Counters defined as
  - Count of E1 to E16+ events in received codewords, summed from both interleaved decoders
  - Reset after sending the AM block
  - Non-rollover

Proposed alternative (cont.)

- At the receiving side, these counters can be accumulated over time, to create a “graph” of long-term event frequency
- This “graph” is a soft metric – various policies can be used to decide on alert
  - Example: calculate a fitted probability and estimate MTTUC, alert if shorter than threshold
  - This metric captures all the required information, even if the channel is not stationary – unlike the current SER threshold
- Accumulated counters can be mapped to MDIO registers at the receiver
- For a PCS adjacent to a PHY XS, the PHY XS effect can be added by taking the maximum of each counter and the corresponding counter received by the PHY XS PCS
  - The maximum represents the worst of the two segments, which will dominate the MTTUC
  - This assumes both segments use the same FEC – which is what we have now

Bit allocation for counters in AM block

- Assuming decreasing probabilities, some of these counters will advance faster than others
- How to allocate bits for each counter in the AM block?
- One possible allocation is according to the expected counts between AMs in a minimally-compliant link
  - Set each counter width so that reaching the maximum count is a very rare event (less than once a day)
- This “maximum allocation” turns out to require many bits, and is different for 400G and 200G
  - Detailed in next slides

“maximum” counter bit allocation for 400G

k	Expected count in 8192 CWs (minimally compliant 400G link)	# bits for Ck	Max count
1	2773	12	4095
2	2037	12	4095
3	996	11	2047
4	364	9	511
5	106	8	255
6	26	6	63
7	5.4	5	31
8	0.98	4	15
9	0.16	3	7
10	2.3E-2	3	7
11	3.0E-3	2	3
12	3.6E-4	2	3
13	4.0E-05	2	3
14	4.1E-06	1	1
15	3.9E-07	1	1
16 or more (UC)	3.8E-08	1	1
Total		82	

“maximum” counter bit allocation for 200G

k	Expected count in 4096 CWs (minimally compliant 200G link)	# bits for Ck	Max count
1	1386	11	2047
2	1018	11	2047
3	498	10	1023
4	182	9	511
5	53	7	127
6	13	6	63
7	2.7	5	31
8	0.49	4	15
9	0.079	3	7
10	1.1E-2	3	7
11	1.5E-3	2	3
12	1.8E-4	2	3
13	2.0E-05	2	3
14	2.0E-06	1	1
15	2.0E-07	1	1
16 or more (UC)	1.9E-08	1	1
Total		78	

Alternative allocation

- Encode counters with variable width, but squeeze into a 64-bit field (suitable for both 400G and 200G, either pad or per-lane octet)
- 64 bits are not enough to guarantee no saturation
  - Workaround: for the large counters, report only the most significant bits (scale down), round upwards
  - Effect of scaling on MTTUC assessment (based on the accumulated counters in the partner) should be insignificant
  - Also, occasional saturation has only a small effect on accumulated counts, so shrink some counters

“minimum” counter bit allocation

k	Expected count in 8192 CWs (minimally compliant 400G link)	# bits for Ck	LSBs truncated	Max count
1	2773	9	3	4088
2	2037	9	3	4088
3	996	8	3	2040
4	364	8	1	510
5	106	7	1	254
6	26	5	1	62
7	5.4	4	1	30
8	0.98	3	1	14
9	0.16	2	1	6
10	2.3E-2	2	0	3
11	3.0E-3	2	0	3
12	3.6E-4	1	0	1
13	4.0E-05	1	0	1
14	4.1E-06	1	0	1
15	3.9E-07	1	0	1
16 or more (UC)	3.8E-08	1	0	1
Total		64		

“Flat” counter bit allocation

- Similar to “minimum” but allocate 4 bits to each of the counters to form a 64-bit field (suitable for both 400G and 200G, either pad or per-lane octet)
  - Simpler to describe, encode and decode
  - More loss of accuracy in the lower k counters, but should still be OK for MTTUC assessment
  - Can be extended to 50G (which has a higher distance between AMs)
- This is the author’s preferred option

“flat” counter bit allocation

k	Expected count in 8192 CWs (minimally compliant 400G link)	# bits for Ck	LSBs truncated	Max count
1	2773	4	8	3840
2	2037	4	8	3840
3	996	4	7	1920
4	364	4	5	480
5	106	4	4	240
6	26	4	2	60
7	5.4	4	1	30
8	0.98	4	0	15
9	0.16	4	0	15
10	2.3E-2	4	0	15
11	3.0E-3	4	0	15
12	3.6E-4	4	0	15
13	4.0E-05	4	0	15
14	4.1E-06	4	0	15
15	3.9E-07	4	0	15
16 or more (UC)	3.8E-08	4	0	15
Total		64		

Comparison to SER degradation

Pros

- Captures required statistics to estimate MTTUC, even in non-stationary channels (no averaging across codewords)
- Soft metric, prevents false alerts, useful in good channels too
- No parameters need to be set in the remote (reporting) receiver

Cons

- Larger gate count (probably negligible)
- More bits consumed in the AM block
- Requires more registers in the “statistics collecting” receiver
- Requires unspecified logic to deduce alert
- Assumes same FEC on main and XS segments

Proposal summary

- Define 16 non-rollover, 12-bit counters Ck (k=1 to 16) in the RS decoder, to count received codewords with k symbol-errors
  - k=16 used to count any uncorrectable codeword
- Encode the counters into a 64-bit block
  - Use the “flat” bit allocation, 4 bits per counter, with scaling as shown above, ordered from C1 to C16
  - If not implemented, encode all-ones instead
  - To prevent long runs in good links, XOR with PRBS9 initialized to all-ones
- Place this block into the pad bits of the AM block transmitted to the partner, am_mapped<1023:960> or am_mapped<1983:1920>
  - Instead of the current pad contents
  - Reset the counters after AM block is transmitted
- If implemented, then when receiving the AM block from the partner, decode and accumulate the counters into 16-bit variables, mapped to MDIO registers, cleared on read
- For a PCS with an adjacent PHY XS, report the maximum of the each of the local counters and the corresponding counter received at the PHY XS PCS

QUESTIONS/COMMENTS?

Thank you