Cl **45** *SC* **45.2.7.6** *P* **42** *L* **26** # 3

Dawe, Piers Avago Technologies

Comment Type **T** *Comment Status* **X**

.3ap text contradicts and/or repeats .3an text; also sentences need re-ordering to make the whole read properly

SuggestedRemedy

Wordsmith it. Also the following three subclauses.

Proposed Response *Response Status* **O**

Cl **45** *SC* **45.2.7.6** *P* **42** *L* **43** # 5

McClellan, Brett Solarflare

Comment Type **E** *Comment Status* **X**

It is unclear which parts of this subclause apply only to backplane and which apply to non-backplane devices. For example, does the text on lines 32 to 35 apply to all devices? Do lines 45 to 49 apply to backplane devices? Page 45 line 43 and page 46 lines 6-7 separately describe the use of bit 7.16.12.

SuggestedRemedy

Break 45.2.7.6 into two subclauses, one describing the use of registers 7.16 to 7.18 for backplane and one for non-backplane devices.

Proposed Response *Response Status* **O**

Cl **72** *SC* **10.4** *P* **131** *L* **32** # 1

Spagna, Fulvio INTEL

Comment Type **E** *Comment Status* **X**

This is out of scope but could be fixed now.

CF28 Value/Comment does not match text in 72.6.10.2.6

SuggestedRemedy

Change CF28 Value/Comment field to match 72.6.10.2.6

Proposed Response *Response Status* **O**

Cl **72** *SC* **6.10** *P* **111** *L* **31** # 2

Spagna, Fulvio INTEL

Comment Type **E** *Comment Status* **X**

Typo from draft 2.4 to 2.5

SuggestedRemedy

left barnch exiting VALID_MARKER state should read

good_markers < 2 *
frame_offset

Proposed Response *Response Status* **O**

Cl **74** *SC* **74.10.3** *P* **192** *L* **31** # 4

Dawe, Piers Avago Technologies

Comment Type **TR** *Comment Status* **X**

Four problems with this state machine: 1. It throws away lock unnecessarily in transient error conditions e.g. lighting strikes (or plugging a neighbouring card in?) hence taking MUCH longer than needed to recover a good link. What it should do is keep lock and de-assert FEC_SIGNAL.indication while BER >10^-4 but lock is OK. 2. At a BER 10^-4, the machine could gain and lose ""lock"" repeatedly - I understand that network management systems really hate anything that can cause unnecessary multiple alarms. Compare Clause 49 64B/66B PCS sync which uses hi_ber to shield the system from this (does it provide hysteresis?). 3. Text says ""Evaluate parity for the potential block i) If the parity does not match (i.e., the received parity does not match the computed parity), shift candidate start by one bit position and try again."" While this state machine tests every hopeless bit position m=8 times before trying the next one. As the probability of a false match is very low, slipping as soon as a parity mismatch is found (like the text says) is the right thing to do. But state diagram takes precedence, therefore a change is needed. 4. All this reminds us why over-prescriptive viewgraph engineering is not so clever. Do we really need a state diagram for this, or can we do the job with words?

SuggestedRemedy

Try to define the lock requirements in words. If we can't, give reason in response, and change state machine so that when seeking lock, a single imperfect block causes a slip, and when in lock, m consecutive correctable or uncorrectable blocks cause FEC_SIGNAL.indication to be false yet not necessarily cause a slip; recovery by n perfect blocks as for initial block lock; m consecutive uncorrectable blocks cause slip as at present.

Proposed Response *Response Status* **O**