Accredited Standards Committee
NCITS, National Committee for Information Technology Standards

Doc: FCW/98w119r0
Date: January 30, 1998
Project:
Ref Doc.:
Reply to: Jim Coomes

To: T11 Membership
From: Jim Coomes

Subject: New link error reporting

A new method for reporting link errors has been proposed for FC and particularly FC-AL. This proposal is for a method to report these errors while running test or ULP protocol in a test mode. The proposal calls for a new primitive signal with an error count field.

First the problems:

The definition of a new primitive is loaded with complexity:

- for ports in Monitoring state, these ports would have to detect frame boundaries and process the frame content. This is not required today and was met with stiff opposition in the FCL Error Recovery SSWG;
- to be even partly reliable, the primitives could not be deleted for clock skew management. If different ports on a loop detect different errors within the same frame, more than one error primitive would follow the frame. If these primitives replace fill words, the ability to perform clock skew management would be broken. If the error primitives are inserted with the elasticity FIFO expanded, the depth of FIFOs would have to increase;
- the delivery of the new error primitive would be unreliable with any present FC-AL devices on the loop. Ports in the "Open" states would remove the primitives;
- the requirement that a port scrub the primitives it originates needs a fail-over to cover the cases of the originating port being removed or the primitive being corrupted; and
- the complexity of future devices would increase if they are required to buffer received error primitives while they are in an Open state and retransmit them.

There is no need for a new way to report link errors. FC-PH already defines a method, the Link Error Status Block for ports to log and report link errors. The problem to date is that FC-PH allows the implementation and degree of support to be optional. If the goal of this new proposal is to provide a link/loop monitor the capability to capture link error information, the LESB function should be enhanced.

Sure a link monitor that wants to gather error information has to initiate the Read Link Status ELS to recover the LESBs. But, the monitor is the device that should bare the burden of the function. The complexity should not be forced on all devices.

Where the real problem is:

- Implementations of the LESB have varied in the implementation due to the lack of consensus on what errors should be counted. A clear definition is needed of what errors may be meaningfully and reasonably counted.
- FC-PH does not include a method for resetting the count fields. They are just allowed to wrap. This means the link monitor must keep the previous readings to determine if the count is increasing.
- Bit error rate testers have a big advantage in detecting and quantifying link errors. They know exactly what was transferred and can compare this expected data to what was received. This allows them to isolate the length of errors.

- When functional information is being transferred on the link, transmitted and received information are quite different. A receiving port does not know what information to expect. It may detect an error in the first character of a frame. This error could be a single bit error and cause running disparity to be wrong for all the remaining
- characters in the frame as running disparity is only recovered between frames. This single bit error may cause 536 Invalid -Transmission-Words for a full size frame before running disparity is restored. But after four consecutive Invalid -Transmission-Words a Loss-of Synchronization occurs.

Thus, determining a bit error rate in a functional system is just not possible. Determining the relative quality of the links is an achievable goal.

The suggested approach:

Support of a clearly defined LESB should required. This would allow link monitors to obtain what information is presently available in LESBs and take advantage of more comprehensive error data when it becomes available.

Jim Coomes
Jim_Coomes@notes.seagate.com