Document:	T10/02-323r0	Date:	August 22, 2002
To:	T10 Committee Membership		
From:	Edward A. Gardner, Basil Networks		
Subject:	SAS Data Corruption Problem		

Consider the situation where a transmission error occurs when an initiator is sending write data to a target. For the sake of argument assume that an SOF is corrupted, so that a frame is completely lost. However the result is substantially similar with other errors, e.g. other coding violations or a CRC error.

Since data transfers are not interlocked, the initiator may not become aware of the problem until many frames after the fact. The target will not be aware of the problem until after it times out the data's arrival or receives a TASK ABORT. Meanwhile the data frame in error is dropped from the middle of the data stream received at the target.

Many or all present day disk drives stream data from SCSI or Fibre Channel onto the media. If SAS drives were to use similar designs, the corrupted data stream will typically have already been written to media by the time the error is discovered. This creates a substantial time window where a power failure, system crash or similar problem will leave the corrupted data on the media permanently. That is unacceptable for most enterprise applications.

Fibre Channel avoids this problem by defining two header fields for error checking. The one most commonly implemented is SEQ_CNT, a sequence count used to check that all data frames arrived. Less often used is RELATIVE OFFSET, which allows the recipient to check that all data bytes arrived.

With SAS all the headers in a sequence of data frames are identical. There is no way for a recipient to distinguish them to detect that any are missing, except by waiting until all have been received.

I see three possible solutions to this problem:

1. Require that SAS drives receive all data requested by a XFER_RDY before any of it is written to media. If this is the expected solution it needs to be documented.
2. Add an equivalent of the Fibre Channel sequence count to SAS data frames.
3. Add an equivalent of the Fibre Channel relative offset to SAS data frames.

I advocate solution 3, as that reuses an offset counter that I expect many implementations will already use internally for other purposes. It also simplifies future (optional) use of random access to data buffers.