Date: November 5, 1996
To: Membership of X3T10 and X3T11
From: David Cressman, Quantum Corp., DLT™ tape engineering

Subject: Proposed General Link Error Recovery Model for Tape Drives

Brief History

After the SCSI Streaming Commands/Medium Changer Commands (SSC/MMC) working group meeting at Natick in Sept., there was a discussion about the issues of error recovery on tape drives, especially for link errors. (On today's tape drives, medium errors are not much of an issue with respect to error recovery, since the drive usually reports a hard write or read error only after exhausting all retry methods at its disposal.) After that, Ed Gardner sent out a reflector email suggesting several positions and approaches, which Charles Monia replied to. Also, there have been a number of discussions about, or related to, this topic at earlier X3T10/11 meetings and on the reflector. This proposal follows up on these previous discussions.

The Need

There seem to be two main things driving the debate: serial SCSI link error rates (e.g. FC-AL) possibly being a not-so-uncommon occurrence, and the need for a generally accepted model for recovery from transport level errors on tape drives.

In looking at the FC-AL link error rate spec, running some calculations, talking to several people more knowledgeable than myself (about actual link error rates), and looking at new developments (copper, >1Gb/s rates, etc.), it seems to me that a link error is likely to occur one or more times when doing a full write or read of even a single high capacity tape cartridge. I calculate that at a 10^-12 link bit error rate, over 100 links, 1KB packets, 1 in 10^7 packets may fail. A 70GB backup has 70GB/1KB = 70M or 7x10^7 packets, which implies relatively frequent backup failures. This is much higher than the observed error rate with parallel SCSI, where in practice we virtually never see bus parity errors on a properly configured bus. This is supported by the fact that no customer has required full automatic recovery from bus parity errors on our (parallel SCSI) tape products. (We are not, in principle, against using Restore Pointers, etc., to do bus level retries of parity errors on data transfers—in fact our products have this capability in some cases—but we have found bus retries of tape data transfers difficult to do without sacrificing cost and/or performance due to the pipelined nature of the data path on our products. In the mid-range tape market we get pushed very hard on both points-- as is everyone else.)

The other driving force for an error recovery model, is that this area is not comprehensively addressed in the SCSI specifications today. I think it would be valuable to have such a model in the SSC document, for the obvious reasons.

The Model

This proposal defines three main cases. For a multi-initiator environment, one initiator (host) should Reserve the Logical Unit. Note that the Read Position and Locate commands are required by the SSC specification now.

1. The host driver keeps track of expected tape position, by keeping track of what it wrote, or by periodically issuing Read Position commands, or both.

   a. When a data transmission error is detected while writing, the tape drive returns an error status, and the host issues a Read Position command. (The read case is similar, but the host detects the error and allows the command to complete or aborts it and issues a Read Position.)
   b. If the position reported indicates the tape drive is not logically positioned just before the block in question, the host must issue a Locate or Space command to logically position there. Otherwise the host proceeds to the next step.
   c. The host reissues the SCSI write or read command, if using variable block mode. When using fixed block mode, the host needs to reset the block count in the CDB to the residue from the Sense Data, and then reissue the command.

2. The tape drive maintains the logical position at the end of the last successfully transfered block. A Mode page bit would be defined and used to indicate support for this-- which allows the host to disable it if desired.

3. The host driver and the tape drive don't support either of the above: then the system integrator must be cognizant of the issue, and only purchase tape drives which solve this problem in a manner consistent with the host tape drivers. This is the "hole" we are trying to solve.