ENDL
TEXAS

Date: 6 July 2006
To: T10 Technical Committee
From: Ralph O. Weber
Subject: How to send SMP over SSP?

Two reasons have been given for defining a mechanism to tunnel SMP requests/responses through an SSP transport:

- Requests and/or responses are growing larger and even sometimes needing to grow larger than a single frame; and
- Processing time for some requests is pushing the limits of the time which SMP allows between receipt of request and return of response.

However, before an SSP tunnel for SMP can be defined, Expander and HBA designers need to select a basic SCSI command/status mechanism for the tunnel. The assumptions and constraints of SCSI commands and status are different from those of SMP requests and responses. These differences must be accounted for in any tunnel design proposed.

I can think of four alternatives but none of them are perfect and each of them places different demands on the tunnel endpoints.

- **Bi-directional Agnostic Command** — This design maps an SMP request and its associated response to a single bi-directional SCSI command in a one-to-one fashion.

- **Unqueued Agnostic Commands** — This is the easiest design to explain because an example already exists (see the SECURITY PROTOCOL IN/OUT commands in SPC-4).

- **Queued Agnostic Commands** — This design has much in common with the Unqueued Agnostic Commands, but it attempts to address an unqueued deficiency by asking targets to support a queue depth of at least two.

- **Knowing Commands** — This design relies on careful matching of SCSI command with SMP requests and responses to eliminate the deficiencies of Agnostic Commands.

Agnostic Commands (unqueued or queued) posit no special relationship between SMP requests, responses, and SCSI commands. The SMP request and response are nothing more than data bits to be transferred one way or the other by the command.

In the agnostic bi-directional variation, a single SCSI command delivers the SMP request in the data-out transfer and retrieves the SMP response during the data-in transfer. SMP requests and responses are matched to each other 'naturally' because both exist within the context of a single SCSI command. If more implementations of bi-directional commands existed, the Bi-directional Agnostic Command would be the hands-down choice for the tunnel.

The remaining agnostic command variations replace the bi-directional command with two SCSI commands. All SMP requests are sent using an SMP OUT SCSI command. All SMP responses are retrieved using an SMP IN command. The issue with these variations of the Agnostic Commands is matching SMP requests to SMP responses and that is were the unqueued versus queued choice comes in.

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

Unqueued Agnostic Commands require the target to 'remember' the SMP response until an SMP IN command arrives to retrieve it. Any requirement for a target to 'remember' something entails rules about how long the 'remembered' stuff is 'remembered'. A sense of how bad (or good) these rules can get may be found in the SPC-4 definition of the SECURITY PROTOCOL IN command.

Queued Agnostic Commands address the 'remembering' problem by requiring the target's command queue to contain an SMP IN command for the response before an SMP OUT command will be processed. The 'remembering' requirement is moved from the target to the HBA in that the HBA is required to allocate a buffer for the response prior to sending the request. (Note: Bi-directional Commands and Knowing Commands have the same requirement, but it is better hidden.)

Knowing Commands attempt to have the benefits of Bi-directional Commands without going over the bi-directional precipice, but doing so requires intimate knowledge of both SMP requests/responses and SCSI commands. This document hints at, but does not fully address, possible issues. (The goal of this document may be viewed as attempting to determine whether the work required to specify the Knowing Commands model is needed.)

SCSI commands (except for bi-directional) require data transfers be in only one direction. Effectively, the designer is allowed a large SMP request with one byte of status (i.e., response) or a limited number of SMP request bytes (about 14 for fixed length CDBs and 180 for variable length CDBs) and as many bytes response as is desired.

Based on these constraints it will be necessary to pick and choose which SMP requests match to which SCSI commands. A few examples are given here to demonstrate in general terms how the Knowing Commands design would work.

The REPORT GENERAL SMP request could be mapped to an SMP IN(16) fixed-length CDB with the relevant portions of the SMP request being transferred in the CDB and the SMP response being transferred as data-in data. This technique works equally well for the REPORT MANUFACTURER INFORMATION, DISCOVER, REPORT PHY ERROR LOG, REPORT PHY SATA, REPORT ROUTE INFORMATION, REPORT PHY EVENT INFORMATION, and REPORT PHY BROADCAST COUNTS SMP requests.

The CONFIGURE GENERAL SMP request could be mapped to an SMP OUT(16) fixed-length CDB with the relevant portions of the SMP request being transferred in the data-out data, GOOD status being mapped to the SMP FUNCTION ACCEPTED function result, and all other function results being mapped to CHECK CONDITION status with the correct function result being placed in the additional sense data. This technique works equally well for the ZONED BROADCAST, CONFIGURE ROUTE INFORMATION, PHY CONTROL, PHY TEST FUNCTION, CONFIGURE PHY EVENT INFORMATION SMP requests.

The DISCOVER LIST request could be mapped to an SMP IN(64) variable-length CDB with the relevant portions of the SMP request being transferred in the CDB and the SMP response being transferred as data-in data. This technique works equally well for the REPORT EXPANDER ROUTE TABLE SMP request.

The headache in the Knowing Commands design is that every SMP request/response pair must be matched to the right SCSI command. Any changes in the length of an SMP request or response must be examined to verify that they do not force a change in the matched SCSI command. From time to time, new SCSI commands may be required to match new SMP requests and responses that do not match any existing SCSI command. Proposal writing (and/or SAS editing) for new or updated SMP functions will become noticeably more challenging because the Knowing Command issues will always need to be considered.