Date: November 13, 2001
To: T10 Committee (SCSI)
From: Julian Satran (IBM) (julian_satran@il.ibm.com), Jim Hafner (IBM) (hafner@almaden.ibm-com)

Subject: Interlocking for exceptions - BUSY, RESERVATION CONFLICT and TASK SET FULL

1. Introduction

This proposal picks up where T10/01-219r0 (proposed by Ed Gardner) left off. Revision0 of this proposal amended the (pending?) proposal T10/00-359r2 (also proposed by Ed Gardner). Revision1 (and beyond if necessary) adds additional function built on those of T10/00-359r7 (the last revision) but is not strictly an amendment to that proposal. Revision 2 simplifies the defined behavior. In Revision 1, the UA was only established as a consequence of going out of the exceptional state. In Revision 2, the UA is established as a consequence of the failure of the first command to get an exception status (see below). Revision 3 just cleans up the change bars and editing history of the document. The “solution” proposed in 00-219 introduced some additional problems and also didn’t fully address the stated problem:

> The various serial protocols (FCP,SBP, SRP, iSCSI) share a possible problem regarding BUSY, RESERVATION CONFLICT and TASK SET FULL. If a system environment where both command queuing and command ordering are important, there is at present no satisfactory interlock between a command being rejected with one of these status codes and subsequent commands.
>
> This is particularly a concern for iSCSI, which contemplates operating in environments with larger communication latencies than traditional SCSI environments.

[To be fair, the author of that proposal was not 100% supportive, but was performing a public service by bringing the issue and the proposal to T10.]

What complicates the issue is the fact that while we would like all the commands in flight after a command is rejected, to be rejected as well (until explicit repair action is performed at the initiator), we would like to see all the commands already queued make as much progress as possible during this type of interlock.

This proposal defines the following model as a solution to this problem.

After the logical unit encounters one of these conditions (busy, task set full or a reservation is established), all commands already in the task set are handled by the current rules. Note that in the reservation state, new commands may enter the task set; for the other two states, no new command can enter the task set until the condition changes. If any command is rejected by one of these statuses, a Unit Attention condition is established for that initiator. If no command has been rejected, no other changes are effected.

In short, as viewed by an initiator the condition BUSY, RESERVATION CONFLICT or TASK SET FULL, when cleared, is replaced by a Unit Attention only if the condition caused at least one command from that initiator to terminate.

This new behavior only has value when the unit attention interlock is enabled (see T10/00-359r7). So, this proposal requires T10/00-359r7 as a prerequisite. It adds additional function on top of those in that proposal; these new functions are optionally enabled only when the function of T10/00-359 are enabled.

[Aside: strictly speaking, the device server may set up a Unit Attention condition as specified here without this proposal, because the device server has the “right” to set up UAs when it feels a need to. However, we feel that it is important to make this behavior standard and predictable.]

The Unit Attention condition is cleared by the mechanism defined in 00-359r7, namely, by a REQUEST SENSE.

2. Proposed changes to SAM-2 (rev 20) and SPC-3

[AUTHOR’S NOTE: The proposed changes here add an additional bit to the Control Mode Page next to the UAINTLCK bit proposed in T10/00-359r7 and allows this bit to be one only if the UAINTLCK bit is one. An alternative would be to join the two bits into a field (perhaps called the UAINTLCK field) with values 00b for current behavior, 10b for behavior specified in T10/00-359r7, 11b for 00-359 behavior plus that specified here and 01b as reserved. We consider this only editorial and leave that to the editor.]

2.1 In SPC-3 (rev 01), Clause 8.3.6, Control mode page:

Add a bit field called UAEXCPT next to the UAINTLCK bit (we suggest UAEXCPT in bit 4 of byte 4 and UAINTLCK in bit 5 of byte 4).

Add the following additional text after the paragraph on pg 203 (pdfpg 224) that will be added for UAINTLCK bit (which should probably follow the paragraph on the RAC bit).

> A unit attention on exception condition (UAEXCPT) bit of one specifies a unit attention condition shall be generated for an initiator when a command from that initiator is terminated with a status of BUSY, TASK SET FULL or RESERVATION CONFLICT (see SAM-2). The device server shall set the additional sense code to PREVIOUS BUSY STATUS, PREVIOUS TASK SET FULL STATUS or PREVIOUS RESERVATION CONFLICT STATUS. The unit attention condition is generated only once regardless to the number of commands terminated with a status of BUSY, TASK SET FULL or RESERVATION CONFLICT.
>
> A UAEXCPT bit of zero specifies that the device server should not generate such a unit attention condition.
>
> If the UAINTLCK bit is zero, the UAEXCPT bit shall be zero.

2.2 In SAM-2 (rev 20, clause 5.8.5)

Insert a new item in the list of events that cause Unit Attention between the current h) and i) to read:

> x) when the UAEXCPT bit in the control mode page (see SPC3) is set to one and at least one command was terminated with a status of BUSY, RESERVATION CONFLICT or TASK SET FULL; or

2.3 In SPC-3, Appendix C, Table C.1

Add three entries for new ASC/ASCQ values (as suggested by the SPC-3 editor):

2C 07	D T L P W R S O M C A E B K	PREVIOUS BUSY STATUS
2C 08	D T L P W R S O M C A E B K	PREVIOUS TASK SET FULL STATUS
2C 09	D T L P W R S O M C A E B K	PREVIOUS RESERVATION CONFLICT STATUS