

Date: August 27, 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 and amends the (pending?) proposal T10/00-359r2 (also proposed by Ed Gardner). 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 commands can enter the task set until the condition changes. After the state changes, if any command has been rejected with the rejecting status, a Unit Attention condition is established for that initiator. If no command has been rejected, no other changes are effected. For RESERVATION CONFLICT the end of the condition is when any reservation is released. For BUSY and TASK SET FULL, the end of the condition is when the device server's resources become available.

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 00-359r2). So, this proposal amends 00-359r2 by adding additional rules that the events noted above trigger a Unit Attention condition.

[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 00-359r2, namely, by a REQUEST SENSE.

## 2. Proposed changes to SAM-2 and SPC-3

In SAM-2 add a clause 5.3.x as follows:

### 5.3.x UA generated when an exception condition disappears

When the UAINTLCK bit in the control mode page (see SPC3) is set to one, then the device server shall generate a Unit Attention condition for each initiator that sent a command that was terminated with a status of BUSY, RESERVATION CONFLICT or TASK SET FULL. The Unit Attention condition is generated only once regardless to the number of commands terminated with a status of BUSY, RESERVATION CONFLICT or TASK SET FULL.

When the UAINTLCK bit in the control mode page is set to zero, the device server may generate the Unit Attention condition.

In SAM-2 (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 UAINTLCK 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 (see 5.3.x); or

In SPC-3 modify clause 8.3.6 as proposed in 00-359r2 in the definition of the UAINTLCK in the Control Mode Page as follows:

A unit attention interlock (uaintlck) bit of zero specifies that the logical unit shall clear any unit attention condition reported with autosense or asynchronous event reporting (see SAM-2).

A unit attention interlock (uaintlck) bit of one specifies that the logical unit shall not clear any unit attention condition reported with autosense or asynchronous event reporting. In addition, if any command is terminated with a BUSY, RESERVATION CONFLICT or TASK SET FULL status, then a unit attention condition shall be established for the initiator receiving that status (see SAM-2) and the additional sense data shall be set to BUSY CONDITION CLEARED, RESERVATIONS RELEASED, or TASK SET FULL CONDITION CLEARED, respectively.

In SPC-3, Appendix C, Table C.1, add two entries for new ASC/ASCQ values:

2A 08	D T L P W R S O M C A E B K	BUSY CONDITION CLEARED
2A 09	D T L P W R S O M C A E B K	TASK SET FULL CONDITION CLEARED

[Editor's note: RESERVATIONS RELEASED is already defined (2A/04). The specific values above are just a suggestion, and are subject to the SPC-3 editor's approval or revision.]

[Editor's note: is there a need to add anything more to the reservation clause concerning this behavior?]