From: Cris Simpson *cris.simpson@intel.com*
Date: March 26, 2002
Title: LOGOUT signals: Model concerns

References

T10/02-121r0	Draft minutes, SRP WG March 14, 2002
SAM-2	SCSI Architecture Model-2, revision 23
SBC-2	SCSI Block Commands-2, revision 05a
SPC-3	SCSI Primary Commands-3, revision 05
SRP	SCSI RDMA Protocol, revision 12, T10/01-328r4

Revision history

Revision 1: Removed Clause 3 (SAM-2). Added target port model, clarifications suggested in March 22 SRP call

Revision 0: Original version, March 21, 2002

1 Introduction

Recent discussions within the SRP WG have suggested the need for a “logout signal”, allowing certain resources to be released by targets and logical units. Defining this signal requires a model for what it does and does not do. Clause 2 introduces new concepts to support the signal, and highlights some apparent problems with the current direction.

2 Model and Issues

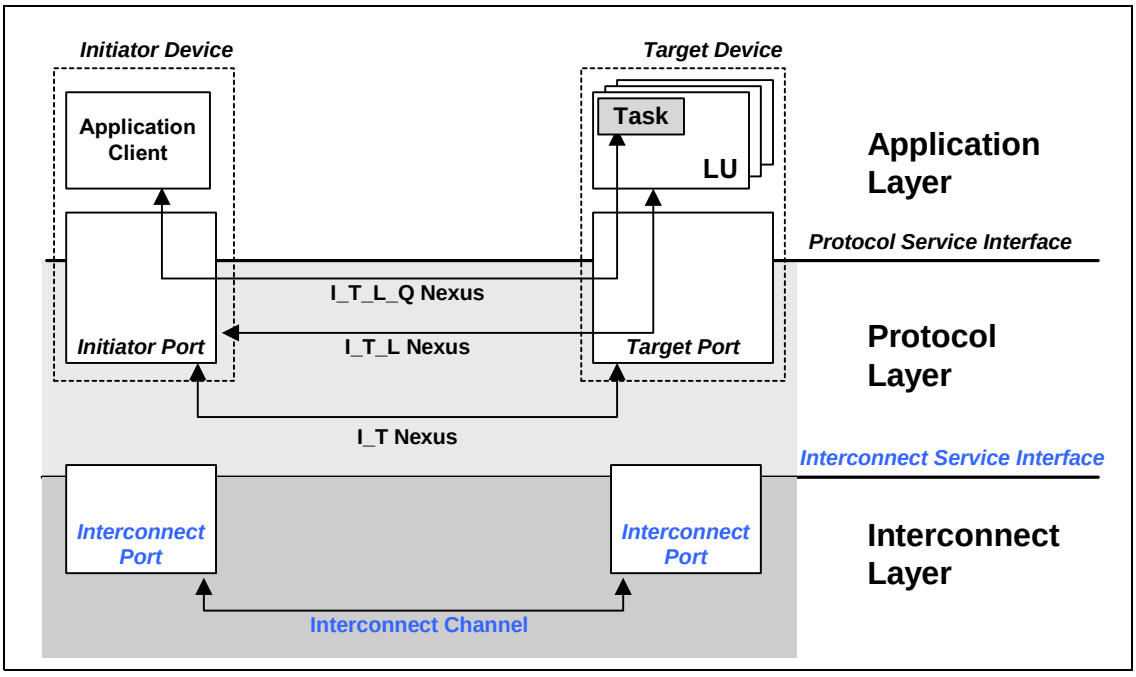

Figure 1: Interconnect Port Model

Under general agreement that ‘logout signal’ is a bad term, this document proposes ‘NEXUS LOST‘ as the name of the signal by which a logout/disconnect/channel error is reported.

A complete signalling mechanism must consider the interconnect layer, not currently discussed in SAM-2. The figure introduces *interconnect ports*, which serve as the connection between the protocol and interconnect layers. The association of two interconnect ports is called an *interconnect channel*, paralleling the SCSI I_T nexus at the protocol layer. In SRP, such an interconnect channel (IC) is called an *RDMA Channel*.

SRP supports a multi-channel mode in which multiple ICs form a single I_T nexus, but when one of the ICs is disestablished, all tasks delivered to the target over that IC are to be aborted. This suggests the need for another signal, INTERCONNECT CHANNEL LOST. This signal would be sent from the interconnect port to its associated SCSI (initiator or target) port. This is distinct from the proposed NEXUS LOST signal, which presumably would be generated when the last IC of an I_T nexus was disestablished.

Several issues discussed at the March 14 SRP WG were resolved by tying them to the NEXUS LOST signal, with the intention that the signal concept would be introduced into other relevant specifications(e.g., SAM, SPC, SBC). While valuable, NEXUS LOST does not solve the problems completely.

(Section numbers based on T10/02-121r0):

(02-121r0) 4.3.1.3 Existing tasks at logout

The WG decided to make logout an SRP protocol-specific event that caused all the initiator's tasks to be aborted. Tasks are typically aborted by an application client issuing an ABORT TASK or ABORT TASK SET task management function request to a logical unit's task manager. The whole point of this discussion being that the application client cannot send a task management request, it must be generated elsehere, namely within the target port.

Although there is no support in SAM-2 for a target port generating task management requests, Rob Elliott pointed out that there is precedent for defining internally-executed task management - a logical unit is required to abort all tasks upon receipt of a LOGICAL UNIT RESET. It should not be a great stretch to specify that the logical unit abort all tasks associated with the specified initiator upon receipt of a NEXUS LOST signal.

The current definition of SRP's multiple independent channel operation (SRP §5.1.4) specifies that all outstanding tasks delivered to the target over a particular channel are to be aborted when that channel is disestablished. Since channels are not visible to logical units, and there is likely to be little support for changing that model, it appears that the target port must maintain a *per-channel task list* (PCTL) so that the target port can issue ABORT TASK task management requests when the associated channel is disestablished. A well-behaved initiator would ensure that the PCTL was empty before issuing SRP_I_LOGOUT, but there are also other ways that channels are disestablished. (*Should the target port issue the task management requests as the initiator?*) The union of PCTLs associated with a particular initiator is called a *per-initiator task list* (See Figure 2).

Neither SRP nor SAM-2 define the order in which tasks are to be aborted. Initiators that depend on the ORDERED task attribute to control execution order could suffer data corruption if an ORDERED task was aborted before the the task it was blocking. Strong execution ordering is a Good Thing, in that it allows initiators to launch several commands at once, eliminating the need to take a context switch for every commands that completes. I see two options for enabling strong ordering: we could either specify an order for aborting tasks (youngest to oldest) , or a means to prevent tasks from a specified initiator from entering the enabled state (e.g., FREEZE_INITIATOR_TASKS, RELEASE_INITIATOR_TASKS).

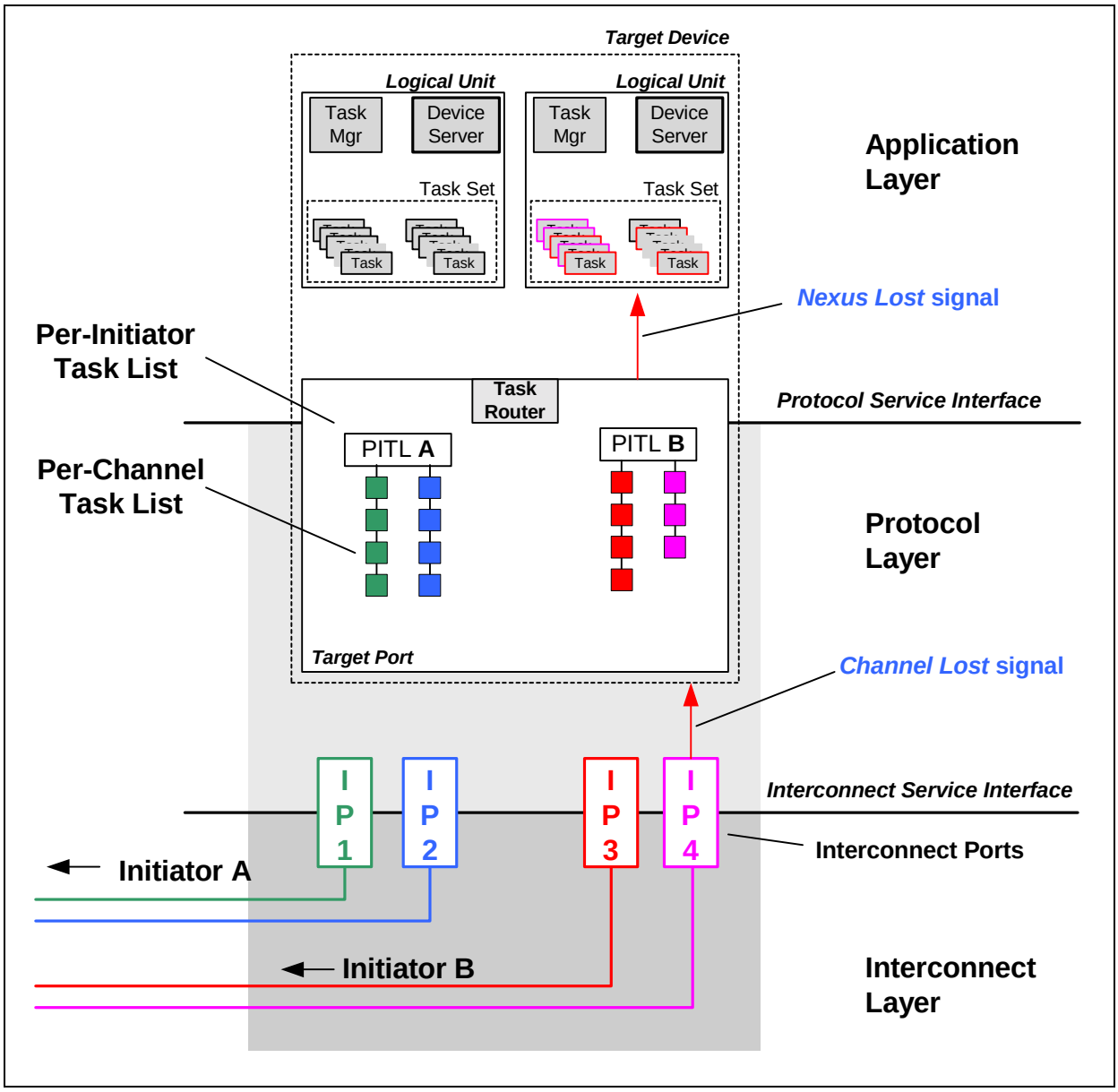

Figure 2: Target port model

4.3.1.6 Buffered data for EXTENDED COPY, 4.3.1.7 Buffered data for XOR commands

At the 3/14 meeting, the recommendation was that SPC3 and SBC2 would add a “protocol-specific behavior” statement, and that SRP’s behavior was to clear the buffered data at logout. It is unclear whether this action occurs upon NEXUS LOST or upon INTERCONNECT CHANNEL LOST. *(I assume NEXUS LOST.)*

Ignoring that question for now, how *does* the target port effect the clearing of that data? For example, SPC3r05 says the following for EXTENDED COPY (7.16.2):

> The copy manager shall discard the COPY STATUS data when:

a) A RECIEVE COPY RESULTS command with COPY STATUS service action is received from the same initiator with a matching list identifier;

b) When another EXTENDED COPY command is received from the same initiator and the list identifier matches the list identifier associated with the data preserved for the COPY STATUS service action;

c) When the copy manager detects a hard reset condition; or

d) When the copy manager requires the resources used to preserve the data.

Which method should a target port use to clear the data? Option *c)* is too dramatic, option *d)* is too non-deterministic, leaving *a)* and *b)* (or a new action TBD). Both options appear to require that the PICTL entry contain much more than just a 'pending' flag.

March 22 SRP WG call decided that we can't/don't want to clear this on INTERCONNECT CHANNEL LOST, as this data is related to the initiator, not the channel. Since the copy manager can discard the data when it needs the space, it appears that we do not need a way to clear it on NEXUS LOST.

SCBC2r05a says:

4.2.3.7 XOR data retention requirements

The target shall retain XOR data while awaiting retrieval by an XDREAD command until performing one of the following events: a matching XDREAD command, logical unit reset, CLEAR TASK SET, ABORT TASK if the task matches the pending XDREAD, or ABORT TASK SET.

It appears that the method to clear buffered data is command-specific.

March 22 SRP WG call decided that we can't/don't want to clear this on INTERCONNECT CHANNEL LOST, as this data is related to the initiator, not the channel.

However, this does seem like something that should be cleared on NEXUS LOST.

4.3.1.9. Preexisting ACA,unit attention,and deferred error conditions

On March 14, the WG decided that SRP should treat these the same as clearing tasks (for that initiator), but worried whether SAM allows this behavior. SAM-2 suggests *not* for ACA (at least not via ABORT TASK):

(SAM-2) 6.2 ABORT TASK

[...] Previously established conditions, including MODE SELECT parameters, reservations, ACA, and CA shall not be changed by the ABORT TASK function.

March 22 SRP WG agreed that ACA is per-initiator, not per-channel.

Deferred Errors -

(SPC-3) 7.25.4 Deferred errors

The deferred error indication may be sent at a time selected by the device server through use of the asynchronous event reporting mechanism (see SAM-2), if AER is supported by both the application client and device server.

I'm unable to find any way to tell a device server to discard unsent deferred error indications. In any case, the LU would associate the indication with an initiator, not an IC.

(SRP has no requirement that such an AER be sent on the IC that delivered the original command.)