INTERNET DRAFT Mallikarjun Chadalapaka
draft-ietf-ips-iscsi-impl-guide-*06*.txt Hewlett-Packard Co.
Editor

Updates: RFC 3720
Intended status: Proposed Standard

Expires August 2007

iSCSI Corrections and Clarifications

Status of this Memo

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Abstract

iSCSI is a SCSI transport protocol and maps the SCSI family of application protocols onto TCP/IP. RFC 3720 defines the iSCSI protocol. This document compiles the clarifications to the original protocol definition in RFC 3720 to serve as a companion document for the iSCSI implementers. This document updates RFC 3720 and the text in this document supersedes the text in RFC 3720 when the two differ.

Table of Contents

1   Definitions and acronyms 5
1.1   Definitions 5
1.2   Acronyms 5
2   Introduction 7
3   iSCSI semantics for SCSI tasks 8
3.1   Residual handling 8
3.1.1 Overview..8
3.1.2 SCSI REPORT LUNS and Residual Overflow................9
3.2   R2T Ordering 10
3.3   Model Assumptions for Response Ordering 11
3.3.1 Model Description....................................11
3.3.2 iSCSI Semantics with the Interface Model.............12
3.3.3 Current List of Fenced Response Use Cases............12
4   Task Management 14
4.1   Requests Affecting Multiple Tasks 14
4.1.1 Scope of affected tasks..............................14
4.1.2 Clarified multi-task abort semantics.................14
4.1.3 Updated multi-task abort semantics...................16
4.1.4 Affected tasks shared across RFC3720 & FastAbort sessions...18
4.1.5 Implementation considerations........................19
4.1.6 Rationale behind the new semantics...................20
5   Discovery semantics 22
5.1   Error Recovery for Discovery Sessions 22
5.2   Reinstatement Semantics of Discovery Sessions 22
5.2.1 Unnamed Discovery Sessions...........................23
5.2.2 Named Discovery Sessions.............................23
5.3   Target PDUs during Discovery 24
6   Negotiation and Others 25
6.1   TPGT Values 25
6.2   SessionType Negotiation 25
6.3   Understanding NotUnderstood 25
6.4   Outstanding Negotiation Exchanges 26
7   iSCSI Error Handling and Recovery 27
7.1   ITT 27
7.2   Format Errors 27
7.3   Digest Errors 27
7.4   Message Error Checking 28
8   iSCSI PDUs 29
8.1   Asynchronous Message 29
8.2   Reject 29
9   Login/Text Operational Text Keys 30
9.1   TaskReporting 30
10   Security Considerations 32
11   IANA Considerations 33
12   References and Bibliography 34
12.1   Normative References.................................34
12.2   Informative References...............................34

13 Editor's Address 35
14 Acknowledgements 36
15 Full Copyright Statement 37
16 Intellectual Property Statement 38

1 Definitions and acronyms

1.1 Definitions

I/O Buffer - A buffer that is used in a SCSI Read or Write operation so SCSI data may be sent from or received into that buffer. For a read or write data transfer to take place for a task, an I/O Buffer is required on the initiator and at least one required on the target.

SCSI-Presented Data Transfer Length (SPDTL): SPDTL is the aggregate data length of the data that SCSI layer logically “presents” to iSCSI layer for a Data-in or Data-out transfer in the context of a SCSI task. For a bidirectional task, there are two SPDTL values - one for Data-in and one for Data-out. Note that the notion of “presenting” includes immediate data per the data transfer model in [SAM2], and excludes overlapping data transfers, if any, requested by the SCSI layer.

Third-party: A term used in this document to denote nexus objects (I_T or I_T_L) and iSCSI sessions which reap the side-effects of actions that take place in the context of a separate iSCSI session, while being third parties to the action that caused the side-effects. One example of a Third-party session is an iSCSI session hosting an I_T_L nexus to an LU that is reset with an LU Reset TMF via a separate I_T nexus.

1.2 Acronyms

Acronym	Definition
EDTL	Expected Data Transfer Length
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
I/O	Input - Output
IP	Internet Protocol
iSCSI	Internet SCSI
iSER	iSCSI Extensions for RDMA

ITT	Initiator Task Tag
LO	Leading Only
LU	Logical Unit
LUN	Logical Unit Number
PDU	Protocol Data Unit
RDMA	Remote Direct Memory Access
R2T	Ready To Transfer
R2TSN	Ready To Transfer Sequence Number
RFC	Request For Comments
SAM	SCSI Architecture Model
SCSI	Small Computer Systems Interface
SN	Sequence Number
SNACK	Selective Negative Acknowledgment - also Sequence Number Acknowledgement for data
TCP	Transmission Control Protocol
TMF	Task Management Function
TTT	Target Transfer Tag
UA	Unit Attention

2 Introduction

Several iSCSI implementations had been built after [RFC3720] was published and the iSCSI community is now richer by the resulting implementation expertise. The goal of this document is to leverage this expertise both to offer clarifications to the [RFC3720] semantics and to address defects in [RFC3720] as appropriate. This document intends to offer critical guidance to implementers with regard to non-obvious iSCSI implementation aspects so as to improve interoperability and accelerate iSCSI adoption. This document, however, does not purport to be an all-encompassing iSCSI how-to guide for implementers, nor a complete revision of [RFC3720]. This document instead is intended as a companion document to [RFC3720] for the iSCSI implementers.

iSCSI implementers are required to reference [RFC3722] and [RFC3723] in addition to [RFC3720] for mandatory requirements. In addition, [RFC3721] also contains useful information for iSCSI implementers. The text in this document, however, updates and supersedes the text in [RFC3720] whenever there is such a question.

3 iSCSI semantics for SCSI tasks

3.1 Residual handling

Section 10.4.1 of [RFC3720] defines the notion of “residuals” and specifies how the residual information should be encoded into the SCSI Response PDU in Counts and Flags fields. Section 3.1.1 clarifies the intent of [RFC3720] and explains the general principles. Section 3.1.2 describes the residual handling in the REPORT LUNS scenario.

3.1.1 Overview

SCSI-Presented Data Transfer Length (SPDTL) is the term this document uses (see section 1.1 for definition) to represent the aggregate data length that the target SCSI layer attempts to transfer using the local iSCSI layer for a task. Expected Data Transfer Length (EDTL) is the iSCSI term that represents the length of data that the iSCSI layer expects to transfer for a task. EDTL is specified in the SCSI Command PDU.

When SPDTL = EDTL for a task, the target iSCSI layer completes the task with no residuals. Whenever SPDTL differs from EDTL for a task, that task is said to have a residual.

If SPDTL > EDTL for a task, iSCSI Overflow MUST be signaled in the SCSI Response PDU as specified in [RFC3720]. Residual Count MUST be set to the numerical value of (SPDTL - EDTL).

If SPDTL < EDTL for a task, iSCSI Underflow MUST be signaled in the SCSI Response PDU as specified in [RFC3720]. Residual Count MUST be set to the numerical value of (EDTL - SPDTL).

Note that the Overflow and Underflow scenarios are independent of Data-in and Data-out. Either scenario is logically possible in either direction of data transfer.

3.1.2 SCSI REPORT LUNS and Residual Overflow

This section discusses the residual overflow issues citing the example of SCSI REPORT LUNS command. Note however that there are several SCSI commands (e.g. INQUIRY) with ALLOCATION LENGTH fields following the same underlying rules. The semantics in the rest of the section apply to all such SCSI commands.

The specification of the SCSI REPORT LUNS command requires that the SCSI target limit the amount of data transferred to a maximum size (ALLOCATION LENGTH) provided by the initiator in the REPORT LUNS CDB. If the Expected Data Transfer Length (EDTL) in the iSCSI header of the SCSI Command PDU for a REPORT LUNS command is set to at least as large as that ALLOCATION LENGTH, the SCSI layer truncation prevents an iSCSI Residual Overflow from occurring. A SCSI initiator can detect that such truncation has occurred via other information at the SCSI layer. The rest of the section elaborates this required behavior.

iSCSI uses the (O) bit (bit 5) in the Flags field of the SCSI Response and the last SCSI Data-In PDUs to indicate that that an iSCSI target was unable to transfer all of the SCSI data for a command to the initiator because the amount of data to be transferred exceeded the EDTL in the corresponding SCSI Command PDU (see Section 10.4.1 of [RFC3720]).

The SCSI REPORT LUNS command requests a target SCSI layer to return a logical unit inventory (LUN list) to the initiator SCSI layer (see section 6.21 of SPC-3 [SPC3]). The size of this LUN list may not be known to the initiator SCSI layer when it issues the REPORT LUNS command; to avoid transfer of more LUN list data than the initiator is prepared for, the REPORT LUNS CDB contains an ALLOCATION LENGTH field to specify the maximum amount of data to be transferred to the initiator for this command. If the initiator SCSI layer has under-estimated the number of logical units at the target, it is possible that the complete logical unit inventory does not fit in the specified ALLOCATION LENGTH. In this situation, section 4.3.3.6 in [SPC3] requires that the target SCSI layer “shall terminate transfers to the Data-In Buffer” when the number of bytes specified by the ALLOCATION LENGTH field have been transferred.

Therefore, in response to a REPORT LUNS command, the SCSI layer at the target presents at most ALLOCATION LENGTH bytes of data (logical unit inventory) to iSCSI for transfer to the initiator. For a REPORT LUNS command, if the iSCSI EDTL is at least as large as the ALLOCATION LENGTH, the SCSI truncation ensures that the EDTL will accommodate all of the data to be transferred. If all of the logical unit inventory data presented to the iSCSI layer - i.e. the data remaining after any SCSI truncation - is transferred to the initiator by the iSCSI layer, an iSCSI Residual Overflow has not occurred and the iSCSI (O) bit MUST NOT be set in the SCSI Response or final SCSI Data-Out PDU. This is not a new requirement but is already required by the combination of [RFC3720] with the specification of the REPORT LUNS command in [SPC3]. If the iSCSI EDTL is larger than the ALLOCATION LENGTH however in this scenario, note that the iSCSI Underflow MUST be signaled in the SCSI Response PDU. An iSCSI Underflow MUST also be signaled when the iSCSI EDTL is equal to ALLOCATION LENGTH but the logical unit inventory data presented to the iSCSI layer is smaller than ALLOCATION LENGTH.

The LUN LIST LENGTH field in the logical unit inventory (first field in the inventory) is not affected by truncation of the inventory to fit in ALLOCATION LENGTH; this enables a SCSI initiator to determine that the received inventory is incomplete by noticing that the LUN LIST LENGTH in the inventory is larger than the ALLOCATION LENGTH that was sent in the REPORT LUNS CDB. A common initiator behavior in this situation is to re-issue the REPORT LUNS command with a larger ALLOCATION LENGTH.

3.2 R2T Ordering

Section 10.8 in [RFC3720] says the following:

> The target may send several R2T PDUs. It, therefore, can have a number of pending data transfers. The number of outstanding R2T PDUs are limited by the value of the negotiated key MaxOutstandingR2T. Within a connection, outstanding R2Ts MUST be fulfilled by the initiator in the order in which they were received.

The quoted [RFC3720] text was unclear on the scope of applicability - either per task, or across all tasks on a connection - and may be interpreted as either. This section is intended to clarify that the scope of applicability of the quoted text is a task. No R2T ordering relationship - either in generation at the target or in fulfilling at the initiator - across tasks is implied. I.e., outstanding R2Ts within a task

MUST be fulfilled by the initiator in the order in which they were received on a connection.

3.3 Model Assumptions for Response Ordering

Whenever an iSCSI session is composed of multiple connections, the Response PDUs (task responses or TMF responses) originating in the target SCSI layer are distributed onto the multiple connections by the target iSCSI layer according to iSCSI connection allegiance rules. This process generally may not preserve the ordering of the responses by the time they are delivered to the initiator SCSI layer. Since ordering is not expected across SCSI responses anyway, this approach works fine in the general case. However to address the special cases where some ordering is desired by the SCSI layer, the following "Response Fence" semantics are defined with respect to handling SCSI response messages as they are handed off from the SCSI protocol layer to the iSCSI layer.

3.3.1 Model Description

Target SCSI protocol layer hands off the SCSI response messages to the target iSCSI layer by invoking the "Send Command Complete" protocol data service ([SAM2], clause 5.4.2) and "Task Management Function Executed" ([SAM2], clause 6.9) service. On receiving the SCSI response message, iSCSI layer exhibits the Response Fence behavior for certain SCSI response messages (section 3.3.3 describes the specific instances where the semantics must be realized). Whenever the Response Fence behavior is required for a SCSI response message, the target iSCSI layer MUST ensure that the following conditions are met in delivering the response message to the initiator iSCSI layer:

(1) Response with Response Fence MUST chronologically be delivered after all the "preceding" responses on the I_T_L nexus, if the preceding responses are delivered at all, to the initiator iSCSI layer.

(2) Response with Response Fence MUST chronologically be delivered prior to all the "following" responses on the I_T_L nexus.

The “preceding” and “following” notions refer to the order of hand-off of a response message from the target SCSI protocol layer to the target iSCSI layer.

3.3.2 iSCSI Semantics with the Interface Model

Whenever the TaskReporting key (section 9.1) is negotiated to ResponseFence or FastAbort for an iSCSI session and the Response Fence behavior is required for a SCSI response message, the target iSCSI layer MUST perform the actions described in this section for that session.:

a) If it is a single-connection session, no special processing is required. Standard SCSI Response PDU build and dispatch process happens.

b) If it is a multi-connection session, target iSCSI layer takes note of last-sent and unacknowledged StatSN on each of the connections in the iSCSI session, and waits for acknowledgement (SHOULD solicit for acknowledgement by way of a Nop-In) of each such StatSN to clear the fence. SCSI response with the Response Fence flag must be sent to the initiator only after receiving acknowledgements for each of the unacknowledged StatSNs.

c) Target iSCSI layer must wait for an acknowledgement of the SCSI Response PDU that carried the response which the target SCSI layer marked with the Response Fence flag. The fence must be considered cleared after receiving the acknowledgement.

d) All further status processing for the LU is resumed only after clearing the fence. If any new responses for the I_T_L nexus are received from the SCSI layer before the fence is cleared, those Response PDUs must be held and queued at the iSCSI layer until the fence is cleared.

3.3.3 Current List of Fenced Response Use Cases

This section lists the fenced response use cases that iSCSI implementations must comply with. However, this is not an exhaustive enumeration. It is expected that as SCSI protocol specifications evolve, the specifications will specify when response fencing is required on a case-by-case basis.

Whenever the TaskReporting key (section 9.1) is negotiated to ResponseFence or FastAbort for an iSCSI session, target iSCSI layer MUST assume that Response Fence flag is set by the target SCSI layer on the following SCSI completion messages handed down to it:

1. The first completion message carrying the UA after the multi-task abort on issuing and third-party sessions.

2. The TMF Response carrying the multi-task TMF Response on the issuing session.

3. The completion message indicating ACA establishment on the issuing session.

4. The first completion message carrying the ACA ACTIVE status after ACA establishment on issuing and third-party sessions.

5. The TMF Response carrying the Clear ACA response on the issuing session.

6. The response to a PERSISTENT RESERVE OUT/PREEMPT AND ABORT command

Note: Due to the absence of ACA-related fencing requirements in [RFC3720], initiator implementations SHOULD NOT use ACA on multi-connection iSCSI sessions to targets complying only with [RFC3720]. Initiators which want to employ ACA on multi-connection iSCSI sessions SHOULD first assess response fencing behavior via negotiating for ResponseFence or FastAbort values for the TaskReporting (section 9.1) key.

4 Task Management

4.1 Requests Affecting Multiple Tasks

This section clarifies and updates the original text in section 10.6.2 of [RFC3720]. The clarified semantics (section 4.1.2) are a superset of the protocol behavior required in the original text and all iSCSI implementations MUST support the new behavior. The updated semantics (section 4.1.3) on the other hand are mandatory only when the new key TaskReporting (section 9.1) is negotiated to “FastAbort”.

4.1.1 Scope of affected tasks

This section defines the notion of “affected tasks” in multi-task abort scenarios. Scope definitions in this section apply to both the clarified protocol behavior (section 4.1.2) and the updated protocol behavior (section 4.1.3).

- ABORT TASK SET: All outstanding tasks for the I_T_L nexus identified by the LUN field in the ABORT TASK SET TMF Request PDU.

- CLEAR TASK SET: All outstanding tasks in the task set for the LU identified by the LUN field in the CLEAR TASK SET TMF Request PDU. See [SPC3] for the definition of a “task set”.

- LOGICAL UNIT RESET: All outstanding tasks from all initiators for the LU identified by the LUN field in the LOGICAL UNIT RESET Request PDU.

- TARGET WARM RESET/TARGET COLD RESET: All outstanding tasks from all initiators across all LUs to which the TMF-issuing session has access to on the SCSI target device hosting the iSCSI session.

Usage: an “ABORT TASK SET TMF Request PDU” in the preceding text is an iSCSI TMF Request PDU with the “Function” field set to “ABORT TASK SET” as defined in [RFC3720]. Similar usage is employed for other scope descriptions.

4.1.2 Clarified multi-task abort semantics

All iSCSI implementations MUST support the protocol behavior defined in this section as the default behavior. The execution of ABORT TASK SET, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, and TARGET COLD RESET TMF Requests consists of the following sequence of actions in the specified order on the specified party.

The initiator iSCSI layer:

a. MUST continue to respond to each TTT received for the affected tasks.

b. Should receive any responses that the target may provide for some tasks among the affected tasks (may process them as usual because they are guaranteed to have chronologically originated prior to the TMF response).

c. Should receive the TMF Response concluding all the tasks in the set of affected tasks.

The target iSCSI layer:

a. MUST wait for responses on currently valid target transfer tags of the affected tasks from the issuing initiator. MAY wait for responses on currently valid target transfer tags of the affected tasks from third-party initiators.

b. MUST wait (concurrent with the wait in Step.a) for all commands of the affected tasks to be received based on the CmdSN ordering. SHOULD NOT wait for new commands on third-party affected sessions - only the instantiated tasks have to be considered for the purpose of determining the affected tasks. In the case of target-scoped requests (i.e. TARGET WARM RESET and TARGET COLD RESET), all the commands that are not yet received on the issuing session in the command stream however can be considered to have been received with no command waiting period - i.e. the entire CmdSN space up to the CmdSN of the task management function can be "plugged".

c. MUST propagate the TMF request to and receive the response from the target SCSI layer.

d. MUST address the Response Fence flag on the TMF Response on issuing session as defined in 3.3.2.

e. MUST address the Response Fence flag on the first post-TMF Response on third-party sessions as defined in 3.3.2. If some tasks originate from non-iSCSI I_T_L nexuses then the means by which the target ensures that all affected tasks have returned their status to the initiator are defined by the specific non-iSCSI transport protocol(s).

Implementation note: Technically, the TMF servicing is complete in Step.d. Data transfers corresponding to terminated tasks may however still be in progress on third-party iSCSI sessions even at the end of Step.e. TMF Response MUST NOT be sent by the target iSCSI layer before the end of Step.d, and MAY be sent at the end of Step.d despite these outstanding data transfers until after Step.e.

4.1.3 Updated multi-task abort semantics

Protocol behavior defined in this section MUST be implemented by all iSCSI implementations complying with this document. Protocol behavior defined in this section MUST be exhibited by iSCSI implementations on an iSCSI session when they negotiate the TaskReporting (section 9.1) key to “FastAbort” on that session. The execution of ABORT TASK SET, CLEAR TASK SET, LOGICAL UNIT RESET, TARGET WARM RESET, and TARGET COLD RESET TMF Requests consists of the following sequence of actions in the specified order on the specified party.

The initiator iSCSI layer:

a. MUST NOT send any more Data-Out PDUs for affected tasks on the issuing connection of the issuing iSCSI session once the TMF is sent to the target.

b. Should receive any responses that the target may provide for some tasks among the affected tasks (may process them as usual because they are guaranteed to have chronologically originated prior to the TMF response).

c. MUST respond to each Async Message PDU with AsyncEvent=5 as defined in section 8.1.

d. Should receive the TMF Response concluding all the tasks in the set of affected tasks.

The target iSCSI layer:

a. MUST wait for all commands of the affected tasks to be received based on the CmdSN ordering on the issuing session. SHOULD NOT wait for new commands on third-party affected sessions - only the instantiated tasks have to be considered for the purpose of determining the affected tasks. In the case of target-scoped requests (i.e. TARGET WARM RESET and TARGET COLD RESET), all the commands that are not yet received on the issuing session in the command stream can be considered to have been received with no command waiting period - i.e. the entire CmdSN space up to the CmdSN of the task management function can be "plugged".

b. MUST propagate the TMF request to and receive the response from the target SCSI layer.

c. MUST leave all active "affected TTTs" (i.e. active TTTs associated with affected tasks) valid.

d. MUST send an Asynchronous Message PDU with AsyncEvent=5 (section 8.1) on:
   i) each connection of each third-party session to which at least one affected task is allegiant if TaskReporting=FastAbort is operational on that third-party session, and
   ii) each connection except the issuing connection of the issuing session that has at least one allegiant affected task.

   If there are multiple affected LUs (say due to a target reset), then one Async Message PDU MUST be sent for each such LU on each connection that has at least one allegiant affected task. The LUN field in the Asynchronous Message PDU MUST be set to match the LUN for each such LU.

e. MUST address the Response Fence flag on the TMF Response on issuing session as defined in 3.3.2.

f. MUST address the Response Fence flag on the first post-TMF Response on third-party sessions as defined in 3.3.2. If some tasks originate from non-iSCSI I_T_L nexuses then the means by which the target ensures that all affected tasks have returned their status to the initiator are defined by the specific non-iSCSI transport protocol(s).

g. MUST free up the affected TTTs (and STags, if applicable) and the corresponding buffers, if any, once it receives each associated Nop-Out acknowledgement that the initiator generated in response to each Async Message.

Implementation note: Technically, the TMF servicing is complete in Step.e. Data transfers corresponding to terminated tasks may however still be in progress even at the end of Step.f. TMF Response MUST NOT be sent by the target iSCSI layer before the end of Step.e, and MAY be sent at the end of Step.e despite these outstanding Data transfers until Step.g. Step.g specifies an event to free up any such resources that may have been reserved to support outstanding data transfers.

4.1.3.1 Clearing effects update

Appendix F.1 of [RFC3720] specifies the clearing effects of target and LU resets on “Incomplete TTTs” as “Y”. This meant that a target warm reset or a target cold reset or an LU reset would clear the active TTTs upon completion. The TaskReporting=FastAbort (section 9.1) semantics defined by this section however do not guarantee that the active TTTs are cleared by the end of the reset operations. In fact, the new semantics are designed to allow clearing the TTTs in a “lazy” fashion after the TMF Response is delivered. Thus, when TaskReporting=FastAbort is operational on a session, the clearing effects of reset operations on “Incomplete TTTs” is “N”.

4.1.4 Affected tasks shared across RFC3720 & FastAbort sessions

If an iSCSI target implementation is capable of supporting TaskReporting=FastAbort functionality (section 9.1), it may end up in a situation where some sessions have TaskReporting=RFC3720 operational (RFC3720 sessions) while some other sessions have TaskReporting=FastAbort operational (FastAbort sessions) even while accessing a shared set of affected tasks (section 4.1.1).

If the issuing session is a RFC3720 session, iSCSI target implementation is FastAbort-capable and third-party affected session is a FastAbort session, the following behavior SHOULD be exhibited by the iSCSI target layer:

a. Between steps c and d of target behavior in section 4.1.2, send an Asynchronous Message PDU with AsyncEvent=5 (section 8.1) on each connection of each third-party session to which at least one affected task is allegiant. If there are multiple affected LUs, then send one Async Message PDU for each such LU on each connection that has at least one allegiant affected task. When sent, the LUN field in the Asynchronous Message PDU MUST be set to match the LUN for each such LU.

b. After step e of target behavior in section 4.1.2, free up the affected TTTs (and STags, if applicable) and the corresponding buffers, if any, once each associated Nop-Out acknowledgement is received that the third-party initiator generated in response to each Async Message sent in step a.

If the issuing session is a FastAbort session, iSCSI target implementation is FastAbort-capable and third-party affected session is a RFC3720 session, the following behavior MUST be exhibited by the iSCSI target layer: Asynchronous Message PDUs MUST NOT be sent on the third-party session to prompt the FastAbort behavior.

If the third-party affected session is a FastAbort session and issuing session is a FastAbort session, initiator in the third-party role MUST respond to each Async Message PDU with AsyncEvent=5 as defined in section 8.1. Note that an initiator MAY thus receive these Async Messages on a third-party affected session even if the session is a single-connection session.

4.1.5 Implementation considerations

Both in clarified semantics (section 4.1.2) and updated semantics (section 4.1.3), there may be outstanding data transfers even after the TMF completion is reported on the issuing session. In the case of iSCSI/iSER [iSER], these would be tagged data transfers for STags not owned by any active tasks. Whether or not real buffers support these data transfers is implementation-dependent. However, the data transfers logically MUST be silently discarded by the target iSCSI layer in all cases. A target MAY, on an implementation-defined internal timeout, also choose to drop the connections on which it did not receive the expected Data-out sequences (section 4.1.2) or Nop-Out acknowledgements (section 4.1.3) so as to reclaim the associated buffer, STag and TTT resources as appropriate.

4.1.6 Rationale behind the new semantics

There are fundamentally three basic objectives behind the semantics specified in section 4.1.2 and section 4.1.3.

1. Maintaining an ordered command flow I_T nexus abstraction to the target SCSI layer even with multi-connection sessions.

   - Target iSCSI processing of a TMF request must maintain the single flow illusion. Target behavior in Step.b of section 4.1.2 and Step.a of section 4.1.3 correspond to this objective.

2. Maintaining a single ordered response flow I_T nexus abstraction to the initiator SCSI layer even with multi-connection sessions when one response (i.e. TMF response) could imply the status of other unfinished tasks from the initiator’s perspective.

   - Target must ensure that the initiator does not see "old" task responses (that were placed on the wire chronologically earlier than the TMF Response) after seeing the TMF response. Target behavior in Step.d of section 4.1.2 and Step.e of section 4.1.3 correspond to this objective.

   - Whenever the result of a TMF action is visible across multiple I_T_L nexuses, [SAM2] requires the SCSI device server to trigger a UA on each of the other I_T_L nexuses. Once an initiator is notified of such an UA, the application client on the receiving initiator is required to clear its task state (clause 5.5 in [SAM2]) for the affected tasks. It would thus be inappropriate to deliver a SCSI Response for a task after the task state is cleared on the initiator, i.e. after the UA is notified. The UA notification contained in the first SCSI Response PDU on each affected Third-party I_T_L nexus after the TMF action thus MUST NOT pass the affected task responses on any of the iSCSI sessions accessing the LU. Target behavior in Step.e of section 4.1.2 and Step.f of section 4.1.3 correspond to this objective.

3. Draining all active TTTs corresponding to affected tasks in a deterministic fashion.

   - Data-out PDUs with stale TTTs arriving after the tasks are terminated can create a buffer management problem even for traditional iSCSI implementations, and is fatal for the connection for iSCSI/iSER implementations. Either the termination of affected tasks should be postponed until the TTTs are retired (as in Step.a of section 4.1.2), or the TTTs and the buffers should stay allocated beyond task termination to be deterministically freed up later (as in Step.c and Step.g of section 4.1.3).

The only other notable optimization is the plugging. If all tasks on an I_T nexus will be aborted anyway (as with a target reset), there is no need to wait to receive all commands to plug the CmdSN holes. Target iSCSI layer can simply plug all missing CmdSN slots and move on with TMF processing. The first objective (maintaining a single ordered command flow) is still met with this optimization because target SCSI layer only sees ordered commands.

5 Discovery semantics

5.1 Error Recovery for Discovery Sessions

The negotiation of the key ErrorRecoveryLevel is not required for Discovery sessions - i.e. for sessions that negotiated “SessionType=Discovery” - because the default value of 0 is necessary and sufficient for Discovery sessions. It is however possible that some legacy iSCSI implementations might attempt to negotiate the ErrorRecoveryLevel key on Discovery sessions. When such a negotiation attempt is made by the remote side, a compliant iSCSI implementation MUST propose a value of 0 (zero) in response. The operational ErrorRecoveryLevel for Discovery sessions thus MUST be 0. This naturally follows from the functionality constraints [RFC3720] imposes on Discovery sessions.

5.2 Reinstatement Semantics of Discovery Sessions

Discovery sessions are intended to be relatively short-lived. Initiators are not expected to establish multiple Discovery sessions to the same iSCSI Network Portal (see [RFC3720]). An initiator may use the same iSCSI Initiator Name and ISID when establishing different unique sessions with different targets and/or different portal groups. This behavior is discussed in Section 9.1.1 of [RFC3720] and is, in fact, encouraged as conservative reuse of ISIDs. ISID RULE in [RFC3720] states that there must not be more than one session with a matching 4-tuple: <InitiatorName, ISID, TargetName, TargetPortalGroupTag>. While the spirit of the ISID RULE applies to Discovery sessions the same as it does for Normal sessions, note that some Discovery sessions differ from the Normal sessions in two important aspects:

* Because [RFC3720] allows a Discovery session to be established without specifying a TargetName key in the Login Request PDU (let us call such a session an “Unnamed” Discovery session), there is no Target Node context to enforce the ISID RULE.

* Portal Groups are defined only in the context of a Target Node. When the TargetName key is NULL-valued (i.e. not specified), the TargetPortalGroupTag thus cannot be ascertained to enforce the ISID RULE.

The following sections describe the two scenarios - Named Discovery sessions and Unnamed Discovery sessions - separately.

5.2.1 Unnamed Discovery Sessions

For Unnamed Discovery sessions, neither the TargetName nor the TargetPortalGroupTag is available to the targets in order to enforce the ISID RULE. So the following rule applies.

UNNAMED ISID RULE: Targets MUST enforce the uniqueness of the following 4-tuple for Unnamed Discovery sessions: <InitiatorName, ISID, NULL, TargetAddress>. The following semantics are implied by this uniqueness requirement.

Targets SHOULD allow concurrent establishment of one Discovery session with each of its Network Portals by the same initiator port with a given iSCSI Node Name and an ISID. Each of the concurrent Discovery sessions, if established by the same initiator port to other Network Portals, MUST be treated as independent sessions - i.e. one session MUST NOT reinstate the other.

A new Unnamed Discovery session that has a matching <InitiatorName, ISID, NULL, TargetAddress> to an existing discovery session MUST reinstate the existing Unnamed Discovery session. Note thus that only an Unnamed Discovery session may reinstate an Unnamed Discovery session.

5.2.2 Named Discovery Sessions

For a Named Discovery session, the TargetName key is specified by the initiator and thus the target can unambiguously ascertain the TargetPortalGroupTag as well. Since all the four elements of the 4-tuple are known, the ISID RULE MUST be enforced by targets with no changes from [RFC3720] semantics. A new session with a matching <InitiatorName, ISID, TargetName, TargetPortalGroupTag> thus will reinstate an existing session. Note in this case that any new iSCSI session (Discovery or Normal) with the matching 4-tuple may reinstate an existing Named Discovery iSCSI session.

5.3 Target PDUs during Discovery

Targets SHOULD NOT send any responses other than a Text Response and Logout Response on a Discovery session, once in full feature phase.

Implementation Note: A target may simply drop the connection in a Discovery session when it would have requested a Logout via an Async Message on Normal sessions.

6 Negotiation and Others

6.1 TPGT Values

[SAM2] and [SAM3] specifications incorrectly note in their informative text that TPGT value should be non-zero, although [RFC3720] allows the value of zero for TPGT. This section is to clarify that zero value is expressly allowed as a legal value for TPGT. This discrepancy currently stands corrected in [SAM4].

6.2 SessionType Negotiation

During the Login phase, the SessionType key is offered by the initiator to choose the type of session it wants to create with the target. The target may accept or reject the offer. Depending on the type of the session, a target may decide on resources to allocate and the security to enforce etc. for the session. If the SessionType key is thus going to be offered as "Discovery", it SHOULD be offered in the initial Login request by the initiator.

6.3 Understanding NotUnderstood

[RFC3720] defines NotUnderstood as a valid answer during a negotiation text key exchange between two iSCSI nodes. NotUnderstood has the reserved meaning that the sending side did not understand the key semantics. This section seeks to clarify that NotUnderstood is a valid answer for both declarative and negotiated keys. The general iSCSI philosophy is that comprehension precedes processing for any iSCSI key. A proposer of an iSCSI key, negotiated or declarative, in a text key exchange MUST thus be able to properly handle a NotUnderstood response.

The proper way to handle a NotUnderstood response varies depending on the lineage and type of the key. All keys defined in [RFC3720] MUST be supported by all compliant implementations; a NotUnderstood answer on any of the [RFC3720] keys therefore MUST be considered a protocol error and handled accordingly. For all other later keys, a NotUnderstood answer concludes the negotiation for a negotiated key whereas for a declarative key, a NotUnderstood answer simply informs the declarer of lack of comprehension by the receiver. In either case, a NotUnderstood answer always requires that the protocol behavior associated with that key be not used within the scope of the key (connection/session) by either side.

6.4 Outstanding Negotiation Exchanges

There was some uncertainty around the number of outstanding Login Response PDUs on a connection. [RFC3720] offers the analogy of SCSI linked commands to Login and Text negotiations in sections 5.3 and 10.10.3 respectively, but does not make it fully explicit. This section is to offer a clarification in this regard.

There MUST NOT be more than one outstanding Login Request or Login Response or Text Request or Text Response PDU on an iSCSI connection. An outstanding PDU in this context is one that has not been acknowledged by the remote iSCSI side.

7 iSCSI Error Handling and Recovery

7.1 ITT

Section 10.19 in [RFC3720] mentions this in passing but noted here again for making it obvious since the semantics apply to the initiators in general. An ITT value of 0xffffffff is reserved and MUST NOT be assigned for a task by the initiator. The only instance it may be seen on the wire is in a target-initiated NOP-In PDU (and in the initiator response to that PDU if necessary).

7.2 Format Errors

Section 6.6 of [RFC3720] discusses format error handling. This section elaborates on the “inconsistent” PDU field contents noted in [RFC3720].

All initiator-detected PDU construction errors MUST be considered as format errors. Some examples of such errors are:

- NOP-In with a valid TTT but an invalid LUN

- NOP-In with a valid ITT (i.e. a NOP-In response) and also a valid TTT

- SCSI Response PDU with Status=CHECK CONDITION, but DataSegmentLength = 0

7.3 Digest Errors

Section 6.7 of [RFC3720] discusses digest error handling. It states that “No further action is necessary for initiators if the discarded PDU is an unsolicited PDU (e.g., Async, Reject)” on detecting a payload digest error. This is incorrect.

An Asynchronous Message PDU or a Reject PDU carries the next StatSN value on an iSCSI connection, advancing the StatSN. When an initiator discards one of these PDUs due to a payload digest error, the entire PDU including the header MUST be discarded. Consequently, the initiator MUST treat the exception like a loss of any other solicited response PDU - i.e. it MUST use one of the following options noted in [RFC3720]:

a) Request PDU retransmission with a status SNACK.

b) Logout the connection for recovery and continue the tasks on a different connection instance.

c) Logout to close the connection (abort all the commands associated with the connection).

7.4 Message Error Checking

There has been some uncertainty on the extent to which incoming messages have to be checked for protocol errors, beyond what is strictly required for processing the inbound message. This section addresses that question.

Unless [RFC3720] or this draft requires it, an iSCSI implementation is not required to do an exhaustive protocol conformance checking on an incoming iSCSI PDU. The iSCSI implementation especially is not required to double-check the remote iSCSI implementation's conformance to protocol requirements.

8 iSCSI PDUs

8.1 Asynchronous Message

This section defines additional semantics for the Asynchronous Message PDU defined in section 10.9 of [RFC3720] using the same conventions.

The following new legal value for AsyncEvent is defined:

5: all active tasks for LU with matching LUN field in the Async Message PDU are being terminated.

The receiving initiator iSCSI layer MUST respond to this Message by taking the following steps in order.

i) Stop Data-Out transfers on that connection for all active TTTs for the affected LUN quoted in the Async Message PDU.
ii) Acknowledge the StatSN of the Async Message PDU via a Nop-Out PDU with ITT=0xffffffff (i.e. non-ping flavor), while copying the LUN field from Async Message to Nop-Out.

8.2 Reject

Section 10.17.1 of [RFC3720] specifies the Reject reason code of 0x0b with an explanation of "Negotiation Reset". At this point, we do not see any legitimate iSCSI protocol use case for using this reason code. Thus reason code 0x0b MUST be considered as deprecated and MUST NOT be used by any new implementations.

9 Login/Text Operational Text Keys

This section follows the same conventions as section 12 of [RFC3720].

9.1 TaskReporting

Use: LO
Senders: Initiator and Target
Scope: SW

Irrelevant when: SessionType=Discovery
TaskReporting=<list-of-values>

Default is RFC3720.
Result function is AND.

This key is used to negotiate the task completion reporting semantics from the SCSI target. Following table describes the semantics an iSCSI target MUST support for respective negotiated key values. Whenever this key is negotiated, at least the RFC3720 and ResponseFence values MUST be offered as options by the negotiation originator.

Name	Description
RFC3720	RFC 3720-compliant semantics. Response fencing is not guaranteed and fast completion of multi-task aborting is not supported
ResponseFence	Response Fence (section 3.3.1) semantics MUST be supported in reporting task completions
FastAbort	Updated fast multi-task abort semantics defined in section 4.1.3 MUST be supported. Support for Response Fence is implied - i.e. section 3.3.1 semantics MUST be supported as well

When TaskReporting is not negotiated to FastAbort, the default behavior is to use the [RFC3720] TMF semantics as clarified in section 4.1.2.

10 Security Considerations

This document does not introduce any new security considerations other than those already noted in [RFC3720].   Consequently, all the iSCSI-related security text in [RFC3723] is also directly applicable to this document.

11 IANA Considerations

This draft does not have any specific IANA considerations.

12 References and Bibliography

12.1 Normative References

[RFC3720] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

[SPC3] T10/1416-D, SCSI Primary Commands-3.

[RFC2119] S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels”, March 1997.

12.2 Informative References

[RFC3721] Bakke, M., Hafner, J., Hufferd, J., Voruganti, K., and M. Krueger, "Internet Small Computer Systems Interface (iSCSI) Naming and Discovery", RFC 3721, April 2004.

[RFC3723] Aboba, B., Tseng, J., Walker, J., Rangan, V., and F. Travostino, "Securing Block Storage Protocols over IP", RFC 3723, April 2004.

[RFC3722] Bakke, M., "String Profile for Internet Small Computer Systems Interface (iSCSI) Names", RFC 3722, April 2004.

[iSER] Ko, M., Chadalapaka, M., Elzur, U., Shah, H., Thaler, P., J. Hufferd, “iSCSI Extensions for RDMA”, IETF Internet Draft draft-ietf-ips-iser-04.txt (work in progress), June 2005.

[RFC2119] Bradner, S. "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[SAM2] ANSI INCITS 366-2003, SCSI Architecture Model-2 (SAM-2).

[SAM3] ANSI INCITS *402-2005*, SCSI Architecture Model-3 (SAM-3).

[SAM4] T10 Project: 1683-D, SCSI Architecture Model-4 (SAM-4), Work in Progress.

13 Editor’s Address

Mallikarjun Chadalapaka
Hewlett-Packard Company
8000 Foothills Blvd.
Roseville, CA 95747-5668, USA
Phone: +1-916-785-5621
E-mail: cbm@rose.hp.com

14 Acknowledgements

The IP Storage (ips) Working Group in the Transport Area of IETF has been responsible for defining the iSCSI protocol (apart from a host of other relevant IP Storage protocols). The editor acknowledges the contributions of the entire working group.

The following individuals directly contributed to identifying [RFC3720] issues and/or suggesting resolutions to the issues clarified in this document: David Black (REPORT LUNS/overflow semantics, ACA semantics, TMF semantics), Gwendal Grignou (TMF scope), Mike Ko (digest error handling for Asynchronous Message), Dmitry Fomichev (reserved ITT), Bill Studenmund (residual handling, discovery semantics), Ken Sandars (discovery semantics), Bob Russell (discovery semantics), Julian Satran (discovery semantics, TMF semantics), Rob Elliott (T10 liaison, R2T ordering), Joseph Pittman(TMF scope), Somesh Gupta (multi-task abort semantics), Eddy Quicksall (message error checking), Paul Koning (message error checking). This document benefited from all these contributions.

15 Full Copyright Statement

16 Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.