

Date: January 14, 2007

To: T10 Committee (SCSI)

From: George Penokie (IBM)

Subject: SAS-2: SAM-4: Miscellaneous State Machine Fixes

## Overview

There have been a errors pointed out in several state machines. This proposal fixes those errors.

## Editorial change in ST\_ITS2

The following change would make the actions defined clearer:

### 9.2.6.2.3.3 ST\_ITS2:Initiator\_Send\_Frame state

If this state is entered from the ST\_ITS3:Prepare\_Command state for transmission of a COMMAND frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS6:Receive\_Data\_In state, and the vendor-specific number of retries has not been reached for the COMMAND frame requesting a read operation, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS4:Prepare\_Task state for transmission of an TASK frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS5:Prepare\_Data\_Out state for transmission of a write DATA frame, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer **#**:

- a) **after** this state has received an XFER\_RDY Arrived message; or
- b) **if** first burst is enabled ~~and~~ **then, after** this state has received a Transmission Status (Frame Transmitted) confirmation and a Transmission Status (ACK Received) confirmation for the COMMAND frame.

The following is an editorial change:

### 9.2.6.2.3.3.4 Transition ST\_ITS2:Initiator\_Send\_Frame to ST\_ITS5:Prepare\_Data\_Out

If first burst is enabled, this transition shall occur and include the First Burst argument after this state receives:

- a) a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) for a COMMAND frame requesting a write operation; or
- b) a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-interlocked) request if the Data-Out Buffer Offset state machine variable is less than the first burst size.

This transition shall occur after this state receives:

- a) an XFER\_RDY Arrived message; or
- b) a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-interlocked) request if the Data-Out Buffer Offset state machine variable is less than the Requested Offset Xfer\_Rdy state machine argument plus the Write Data Length Xfer\_Rdy state machine argument.

NOTE 1 - This transition occurs even if this state has not received a Transmission Status (ACK Received) for the write DATA frame.

This transition shall include a Retry argument and occur after:

- a) this state receives one of the following confirmations or arguments for a write DATA frame:
  - A) Transmission Status (NAK Received);
  - B) Transmission Status (ACK/NAK Timeout); or
  - C) Transmission Status (Connection Lost without ACK/NAK);

- b) the RETRY DATA FRAMES bit is set to one in the XFER\_RDY frame for the write operation;
- c) the Data-Out Buffer Offset state machine variable is set to the Requested Offset Xfer\_Rdy state machine argument; and
- d) the vendor-specific number of retries, if any, for the write DATA frame has not been reached.

### ST\_IFR state missing Transmission Complete messages from ST\_ITS2

The paragraphs below from 9.2.6.2.3.3 ST\_ITS2:Initiator\_Send\_Frame state indicate the two messages that are missing from 9.2.6.2.2.4 Processing Transmission Complete and Reception Complete messages.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I\_T Nexus Loss) (see table 131 in 8.2.2.3.4), and the Transmit Frame request was for a COMMAND frame or a DATA frame, then this state shall send a **Transmission Complete (Command Failed, Connection Failed)** message to the ST\_IFR state machine. The message shall include the tag.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I\_T Nexus Loss) (see table 131 in 8.2.2.3.4), and the Transmit Frame request was for a TASK frame, then this state shall send a **Transmission Complete (Task Failed, Connection Failed)** message to the ST\_IFR state machine. The message shall include the tag.

The correction is to add two rows to the “Confirmations sent to the SCSI application layer if a frame transmission or reception error occurs” table in section 9.2.6.2.2.4 Processing Transmission Complete and Reception Complete messages as shown below.

### ST\_IFR state missing Transmission Complete messages from ST\_ITS6

The paragraphs below from 9.2.6.2.3.7 ST\_ITS6:Receive\_Data\_In state indicate the message that is missing from 9.2.6.2.2.4 Processing Transmission Complete and Reception Complete messages.

If this state receives a Cancel message, then this state shall send a **Reception Complete (Cancel Acknowledged)** message to the ST\_IFR state machine. The Reception Complete message shall include the tag as an argument.

The correction is to add a row to the “Confirmations sent to the SCSI application layer if a frame transmission or reception error occurs” table in section 9.2.6.2.2.4 Processing Transmission Complete and Reception Complete messages as shown below.

In addition the a,b,c list in section 9.2.6.2.3.7.2 Transition ST\_ITS6:Receive\_Data\_In to ST\_ITS1:Initiator\_Start needs to be changed as indicated:

This transition shall occur after this state:

- e) sends one of the following to the ST\_IFR state machine:
  - A) a Reception Complete (Data Offset Error) message;
  - B) a Reception Complete (Too Much Read Data) message;
  - C) a Reception Complete (Incorrect Data Length) message; or
  - D) a **Transmission Reception Complete (Cancel Acknowledged)** message;
- or
- f) receives a Return To Start message.

**Table 1 — Confirmations sent to the SCSI application layer if a frame transmission or reception error occurs**

Message received from ST_ITS state machine	Protocol service confirmation and Delivery Result argument sent to the SCSI application layer
Transmission Complete (Command Failed, ACK/NAK Timeout)	Command Complete Received (Service Delivery or Target Failure - ACK/NAK Timeout)
Transmission Complete (Command Failed, NAK Received)	Command Complete Received (Service Delivery or Target Failure - NAK Received)
<a href="#">Transmission Complete (Command Failed, Connection Failed)</a>	<a href="#">Command Complete Received (Service Delivery or Target Failure - Connection Failed)</a>
Transmission Complete (Task Failed, ACK/NAK Timeout)	Received Task Management Function - Executed (Service Delivery or Target Failure - ACK/NAK Timeout)
Transmission Complete (Task Failed, NAK Received)	Received Task Management Function - Executed (Service Delivery or Target Failure - NAK Received)
<a href="#">Transmission Complete (Task Failed, Connection Failed)</a>	<a href="#">Received Task Management Function - Executed (Service Delivery or Target Failure - Connection Failed)</a>
Transmission Complete (XFER_RDY Incorrect Write Data Length)	Command Complete Received (Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length)
Transmission Complete (XFER_RDY Requested Offset Error)	Command Complete Received (Service Delivery or Target Failure - XFER_RDY Requested Offset Error)
Transmission Complete (Cancel Acknowledged)	Command Complete Received (Service Delivery or Target Failure - Cancel Acknowledged)
Reception Complete (Data Offset Error)	Command Complete Received (Service Delivery or Target Failure - DATA Offset Error)
Reception Complete (Too Much Read Data)	Command Complete Received (Service Delivery or Target Failure - DATA Too Much Read Data)
Reception Complete (Information Unit Too Short)	Command Complete Received (Service Delivery or Target Failure - DATA Information Unit Too Short)
Reception Complete (Command Failed, ACK/NAK Timeout)	Command Complete Received (Service Delivery or Target Failure - ACK/NAK Timeout)
<a href="#">Reception Complete (Cancel Acknowledged)</a>	<a href="#">Command Complete Received (Service Delivery or Target Failure - Cancel Acknowledged)</a>

### Mislabeled variable in ST\_ITS2 state description

#### 9.2.6.2.3 ST\_ITS (initiator transport server) state machine

##### 9.2.6.2.3.1 ST\_ITS state machine overview

The ST\_ITS state machine performs the following functions:

- a) receives and processes messages from the ST\_IFR state machine;
- b) sends messages to the ST\_IFR state machine;
- c) sends request to the port layer regarding frame transmission;

- d) receives confirmations from the port layer regarding frame transmission; and
- e) receives HARD\_RESET Received confirmations from the port layer.

This state machine consists of the following states:

- a) ST\_ITS1:Initiator\_Start state (see 9.2.6.2.3.2) (initial state);
- b) ST\_ITS2:Initiator\_Send\_Frame state (see 9.2.6.2.3.3);
- c) ST\_ITS3:Prepare\_Command state (see 9.2.6.2.3.4);
- d) ST\_ITS4:Prepare\_Task state (see 9.2.6.2.3.5);
- e) ST\_ITS5:Prepare\_Data\_Out state (see 9.2.6.2.3.6); and
- f) ST\_ITS6:Receive\_Data\_In state (see 9.2.6.2.3.7).

This state machine shall start in the ST\_ITS1:Initiator\_Start state after power on.

If this state machine receives a HARD\_RESET Received confirmation, then this state machine shall transition to the ST\_ITS1:Initiator\_Start state.

This state machine shall maintain the state machine variables defined in table 2.

**Table 2 — ST\_ITS state machine variables**

State machine variable	Description
Data-In Buffer Offset	Current offset in the data-in buffer for read data
Data-Out Buffer Offset	Current offset in the data-out buffer for write data
Previous Requested Offset	Data offset from the last XFER_RDY frame received
Previous Write Data Length	Write data length from the last XFER_RDY frame received

This state machine shall maintain the state machine arguments defined in table 3.

**Table 3 — ST\_ITS state machine arguments**

State machine argument	Description
Command	Consists of the Command arguments received in the Request (Send Command) message
Task	Consists of the arguments received in the Request (Send Task) message
Xfer_Rdy	Consists of the arguments received in the XFER_RDY Arrived message
Data-Out Buffer	The location of the write data buffer
Data-Out Buffer Size	The size in bytes of the write data buffer
Data-In Buffer Size	The size in bytes of the read data buffer

### 9.2.6.2.3.2 ST\_ITS1:Initiator\_Start state

#### 9.2.6.2.3.2.1 State description

This state is the initial state of the ST\_ITS state machine.

Upon entry into this state, this state shall set the Data-In Buffer Offset state machine variable to zero.

Upon entry into this state, this state shall set the Data-Out Buffer Offset state machine variable to zero.

#### 9.2.6.2.3.2.2 Transition ST\_ITS1:Initiator\_Start to ST\_ITS3:Prepare\_Command

This transition shall occur after this state receives a Request (Send Command) message.

### 9.2.6.2.3.2.3 Transition ST\_ITS1:Initiator\_Start to ST\_ITS4:Prepare\_Task

This transition shall occur after this state receives a Request (Send Task) message.

### 9.2.6.2.3.3 ST\_ITS2:Initiator\_Send\_Frame state

If this state is entered from the ST\_ITS3:Prepare\_Command state for transmission of a COMMAND frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS6:Receive\_Data\_In state, and the vendor-specific number of retries has not been reached for the COMMAND frame requesting a read operation, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS4:Prepare\_Task state for transmission of an TASK frame, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST\_ITS5:Prepare\_Data\_Out state for transmission of a write DATA frame, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer if:

- a) this state has received an XFER\_RDY Arrived message; or
- b) first burst is enabled and this state has received a Transmission Status (Frame Transmitted) confirmation and a Transmission Status (ACK Received) confirmation for the COMMAND frame.

A Transmit Frame request shall include the COMMAND frame from the ST\_ITS3:Prepare\_Command state or from the ST\_ITS6:Receive\_Data\_In state, the TASK frame from the ST\_ITS4:Prepare\_Task state, or the write DATA frame from the ST\_ITS5:Prepare\_Data\_Out state and the following arguments to be used for any OPEN address frame:

- a) initiator port bit set to one;
- b) protocol set to SSP;
- c) Connection Rate argument;
- d) Initiator Connection Tag argument;
- e) Destination SAS Address argument; and
- f) Source SAS Address argument.

After sending a Transmit Frame request this state shall wait to receive a Transmission Status confirmation.

If the confirmation is Transmission Status (I\_T Nexus Loss), then this state shall send a Transmission Complete (I\_T Nexus Loss) message to the ST\_IFR state machine. This Transmission Complete message shall include the tag as an argument.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I\_T Nexus Loss) (see table 131 in 8.2.2.3.4), and the Transmit Frame request was for a COMMAND frame or a DATA frame, then this state shall send a Transmission Complete (Command Failed, Connection Failed) message to the ST\_IFR state machine. The message shall include the tag.

If the confirmation is not Transmission Status (Frame Transmitted) or Transmission Status (I\_T Nexus Loss) (see table 131 in 8.2.2.3.4), and the Transmit Frame request was for a TASK frame, then this state shall send a Transmission Complete (Task Failed, Connection Failed) message to the ST\_IFR state machine. The message shall include the tag.

If the confirmation is Transmission Status (Frame Transmitted), and the Transmit Frame request was for a COMMAND frame not requesting a read operation, a COMMAND frame not requesting a write operation, a TASK frame, or a write DATA frame where the number of data bytes that have been transmitted equal the ~~request byte count~~ [Data-Out Buffer Size](#), then this state shall wait to receive one of the following confirmations:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

If the confirmation is Transmission Status (Frame Transmitted), and the Transmit Frame request was for a COMMAND frame requesting a write operation, or a write DATA frame where the number of data bytes that

have been transmitted is less than the ~~request byte count~~ [Data-Out Buffer Size](#) and the write data length from the previous XFER\_RDY frame, then this state shall wait to receive one of the following confirmations:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout);
- d) Transmission Status (Connection Lost Without ACK/NAK); or
- e) XFER\_RDY Arrived message.

### Confirmation sent to wrong layer in ST\_TFR state machine

In section 9.2.6.3.2.2 Processing Frame Received confirmations the following paragraph references the port layer when it should be referencing the SCSI application layer. See below for correction:

If the frame type is TASK, this state machine checks tags, the RETRANSMIT bit in the new TASK frame is set to one, and the tag for the new TASK frame is the same as the tag for a previous TASK frame where the task management function for the previous TASK frame is not complete, then this state machine shall discard the new TASK frame and not send a Task Management Request Received confirmation to the [SCSI application](#) ~~port~~-layer.

### SAS Source address not passed between link layer and port layer

The SAS source address that is received in an open address frame is not passed to the port layer. This is an error as the port layer needs to pass the SAS source address to the transport layer whenever a frame is received. The transport layer uses the SAS source address to determine if the frame came from the right source by comparing that address against the hashed SAS source address contained within the received frame, if that checking is enabled. To resolve this the following change needs to be made.

In addition I discovered that there was no message going to PL\_OC to indicate that an open was received. To resolve these the following changes need to be made.

#### 7.14.4.4.3 Transition SL\_CC2:Selected to SL\_CC3:Connected

This transition shall occur after sending a Connection Opened confirmation.

This transition shall include:

- a) an Open SSP Connection, Open STP Connection, or Open SMP Connection argument based on the requested protocol; [and](#)
- b) [the received OPEN address frame.](#)

#### 8.2.3.2.3 Transition PL\_PM1:Idle to PL\_PM3:Connected

This transition shall occur after a Connection Opened confirmation is received. This transition shall include [the received OPEN address frame.](#)

#### 8.2.3.4.1 PL\_PM3:Connected state description

[If this state was entered from the PL\\_PM1:Idle state, then this state shall send a Connection Opened message to the PL\\_OC state machine that includes as an argument the received OPEN address frame.](#)

---



---

[Editor's Note 1: This message also needs to be added into the PL\\_PM state machine figure.](#)

---



---

### Missing responses to Transmission status confirmations in ST\_TTS2

None of the Transmission status confirmations relating to a abandoned open attempts or to a break are defined in the target transport layer. The following changes needs to be added to fix that.

## 9.2.6.3.2.3 Processing transport protocol service requests and responses

Table 4 — Confirmations sent to the SCSI application layer

Message received from ST_TTS state machine	Protocol service confirmation sent to SCSI application layer
Transmission Complete (Xfer_Rdy Delivered)	None
Transmission Complete (Response Delivered)	None
Transmission Complete (Response Failed) <sup>a</sup>	None
<a href="#">Transmission Complete (Connection Failed)</a>	<a href="#">Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED</a>
<a href="#">Transmission Complete (Connection Failed)</a>	<a href="#">Data-Out Delivered with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED</a>
<a href="#">Transmission Complete (Data Transfer Terminated)</a>	<a href="#">Data Transfer Terminated</a>
Transmission Complete (Data-In Delivered)	Data-In Delivered with the Delivery Result argument set to DELIVERY SUCCESSFUL
Transmission Complete (Xfer_Rdy Failed, NAK Received)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - NAK RECEIVED
Transmission Complete (Xfer_Rdy Failed, ACK/NAK Timeout)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED
Transmission Complete (Data-In Failed, NAK Received)	Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - NAK RECEIVED
Transmission Complete (Data-In Failed, ACK/NAK Timeout)	Data-In Delivered with the Delivery Result argument set to DELIVERY FAILURE - CONNECTION FAILED
Reception Complete (Data-Out Received)	Data-Out Received with the Delivery Result argument set to DELIVERY SUCCESSFUL
Reception Complete (Data Offset Error)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - DATA OFFSET ERROR
Reception Complete (Too Much Write Data)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - TOO MUCH WRITE DATA
Reception Complete (Information Unit Too Short)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - INFORMATION UNIT TOO SHORT.
Reception Complete (Initiator Response Timeout)	Data-Out Received with the Delivery Result argument set to DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT
Reception Complete (Data Transfer Terminated)	Data Transfer Terminated
<sup>a</sup> SAM-4 does not define a mechanism for the device server to determine the result of its Send Command Complete and Task Management Function Executed transport protocol service response calls.	

### 9.2.6.3.3.3 ST\_TTS2:Target\_Send\_Frame state

....

If the confirmation or argument is not Transmission Status (Frame Transmitted) or Transmission Status (I\_T Nexus Loss), then this state shall send the Transmission Complete message defined in table 158 [or table 5](#) to the ST\_TFR state machine. The message shall include the following arguments:

- a) tag; and
- b) arguments received with the Transmission Status confirmation.

....

[Table 5 defines the messages that this state shall send to the ST\\_TFR state machine upon receipt of the listed confirmations.](#)

**Table 5 — Additional messages sent to ST\_TFR state machine**

<a href="#">Confirmation received</a>	<a href="#">Message to be sent to ST_TFR</a>
<a href="#">Transmission Status (Bad Destination)</a>	<a href="#">Transmission Complete (Connection Failed)</a>
<a href="#">Transmission Status (Connection Rate Not Supported)</a>	<a href="#">Transmission Complete (Connection Failed)</a>
<a href="#">Transmission Status (Protocol Not Supported)</a>	<a href="#">Transmission Complete (Connection Failed)</a>
<a href="#">Transmission Status (STP Resources Busy)</a>	<a href="#">Transmission Complete (Connection Failed)</a>
<a href="#">Transmission Status (Wrong Destination)</a>	<a href="#">Transmission Complete (Connection Failed)</a>
<a href="#">Transmission Status (Break Received)</a>	<a href="#">Transmission Complete (Data Transfer Terminated)</a>

#### 9.2.6.3.3.6.1 State description

...

[If this state receives Transmission Status \(Break Received\) confirmation, then this state shall send a Reception Complete \(Data Transfer Terminated\) to the ST\\_TFR state machine.](#)

---

Editor's Note 2: There needs to be a Transmission Status confirmation arrow added to the ST\_TTS5: Receive\_Data\_Out state in the ST\_T state machine figure.

---

### Making it clear in PL\_OC2 that TASK frames are interlocked

The current wording on the PL\_OC2:Overall\_Control state frame transmission on the sequencing of the TASK frames could be interpreted to allow multiple TASK frames to be outstanding on a wide port which should not be allowed. The following wording change will make this clear.

#### 8.2.2.3.6 PL\_OC2:Overall\_Control state frame transmission

....

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I\_T\_L\_Q nexus (e.g., an ABORT TASK or QUERY TASK task management function (see SAM-4)) [to any](#)



[PL\\_PM state machine](#) until this state has received one of the following messages for each Tx Frame message with the same I\_T\_L\_Q nexus:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I\_T\_L nexus (e.g., an ABORT TASK SET, CLEAR TASK SET, CLEAR ACA, or LOGICAL UNIT RESET task management function (see SAM-4)) [to any PL\\_PM state machine](#) until this state has received one of the following messages for each Tx Frame message with the same I\_T\_L nexus:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

This state shall not send a Tx Frame message containing a TASK frame for a task that only affects an I\_T nexus (e.g., an I\_T NEXUS RESET task management function (see SAM-4)) [to any PL\\_PM state machine](#) until this state has received one of the following messages for each Tx Frame message with the same I\_T nexus:

- a) Transmission Status (ACK Received);
- b) Transmission Status (NAK Received);
- c) Transmission Status (ACK/NAK Timeout); or
- d) Transmission Status (Connection Lost Without ACK/NAK).

## Missing responses NOTIFY (POWER LOSS EXPECTED)

None of the state machines cover what happens when a NOTIFY (POWER LOSS EXPECTED) is received. The following changes needs to be added to fix that.

## SAS-2 changes

### 7.2.5.10.3 NOTIFY (POWER LOSS EXPECTED)

NOTIFY (POWER LOSS EXPECTED) is transmitted by a SAS initiator port or expander port and is used to specify to a SAS target device that power loss may occur within the time specified in the POWER LOSS TIMEOUT field in the Protocol-Specific Port mode page Shared Port Control subpage (see 10.2.7.2.4).

NOTIFY (POWER LOSS EXPECTED) shall be transmitted at least three times by the SAS initiator port or expander port.

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then the device server for each logical unit to which the SSP target port has access shall:

- 1) stop writing data to the media on a block boundary (e.g., all write activity shall continue until a block boundary is reached then all writing shall stop); ~~and~~
- 2) clear all task sets [as defined in SAM-4](#) ~~(i.e., the device server acts as if it has received a CLEAR-TASK-SET task management function (see SAM-4) for each task set); and~~
- 3) establish a unit attention condition [as defined in SAM-4](#) ~~for the initiator port associated with every I\_T nexus with the additional sense code set to COMMANDS-CLEARED-BY-POWER-LOSS-NOTIFICATION.~~

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then each SAS phy within the target device shall, within 1 ms:

- a) ~~if there is an SSP connection, then transmit a BREAK on that connection; and~~
- a) on the phy that receives the NOTIFY (POWER LOSS EXPECTED), if there is an SSP connection, then transmit a BREAK on that connection; and
- b) on the phy that does not receive the NOTIFY (POWER LOSS EXPECTED), if there is an SSP connection, then cancel that connection (e.g., transmit a BREAK or a CLOSE on that connection).

The SCSI Application layer that receives a Power Loss Expected event shall:

- a) start the power loss timer;
- b) send an Accept\_Reject OPENs (Reject SSP) request to all ST\_T state machines (i.e., all SSP connection requests result in OPEN\_REJECT (RETRY));
- c) if a SCSI Command Received transport protocol service indication is received, then the device server shall abort that command and send an Accept\_Reject OPENs (Reject SSP) request to the ST\_T state machine on which the SCSI Command Received transport protocol service indication was received; and
- d) ~~respond to SSP connection requests with OPEN\_REJECT (RETRY) until~~ if the power loss timeout timer expires, then the SCSI Application layer shall send an Accept\_Reject OPENs (Accept SSP) request to all ST\_T state machines ~~or power is lost.~~

If any frames are received by the SAS target device after receiving NOTIFY (POWER LOSS EXPECTED) before a connection is closed, then the SAS target device shall discard the received frames.

After power on, the power loss timeout timer shall be initialized and stopped until a NOTIFY (POWER LOSS EXPECTED) is received.

## 7.14 SL (link layer for SAS phys) state machines

### 7.14.1 SL state machines overview

The SL (link layer for SAS phys) state machines controls connections, handling both connection requests (OPEN address frames), CLOSEs, and BREAKs. The SL state machines are as follows:

- a) SL\_RA (receive OPEN address frame) state machine (see 7.14.3); and
- b) SL\_CC (connection control) state machine (see 7.14.4).

All the SL state machines shall begin after receiving an Enable Disable SAS Link (Enable) message from the SL\_IR state machines.

If a state machine consists of multiple states the initial state is as indicated in the state machine description.

Figure 1 shows the SL state machines.

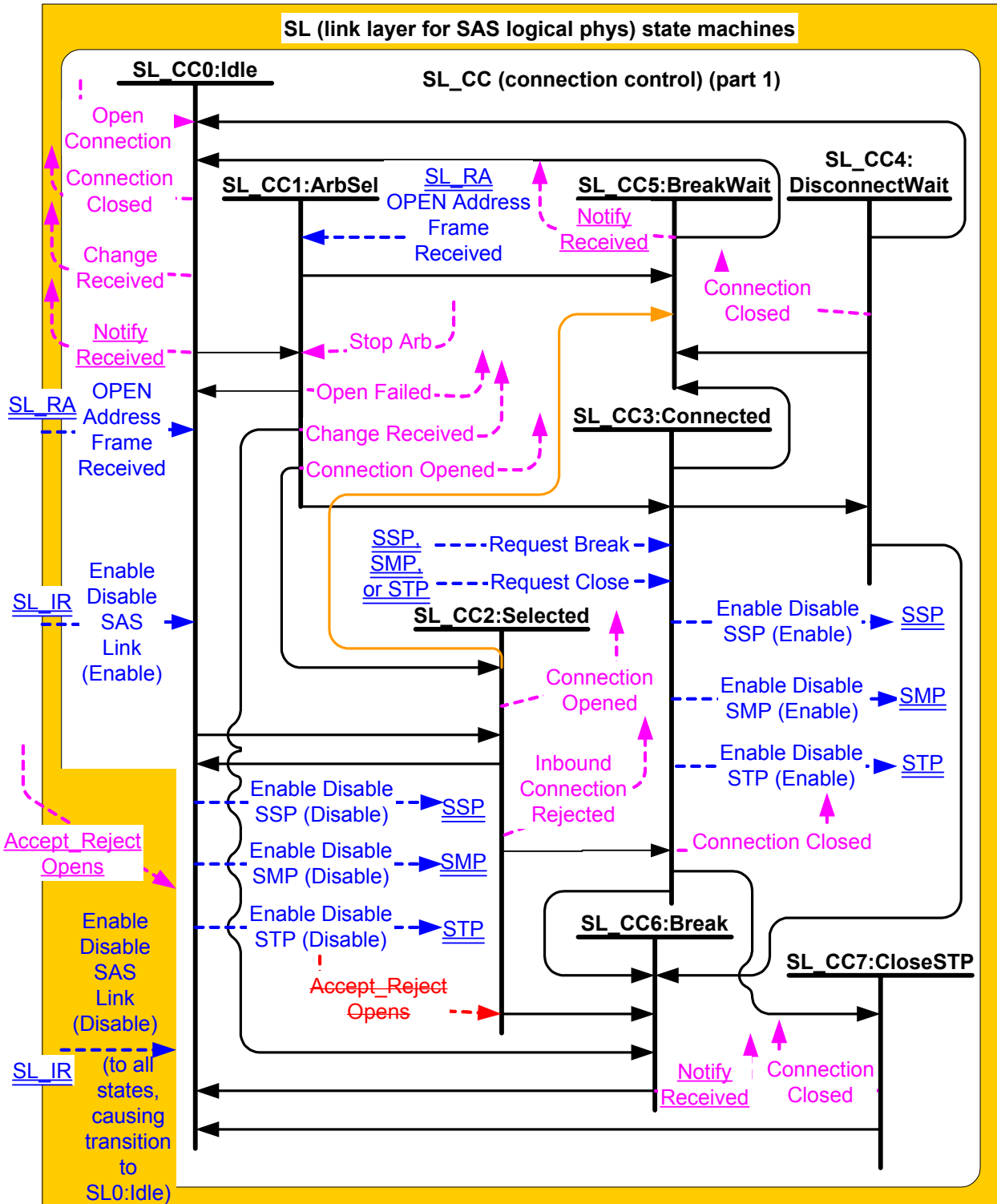


Figure 1 — SL (link layer for SAS phys) state machines (part 1)

Editor's Note 3: Figure 1 has changed. The changes are indicated by the orange arrow and by underlines on the confirmations, requests, and messages.

Figure 2 shows the messages sent to the SL transmitter and received from the SL receiver.

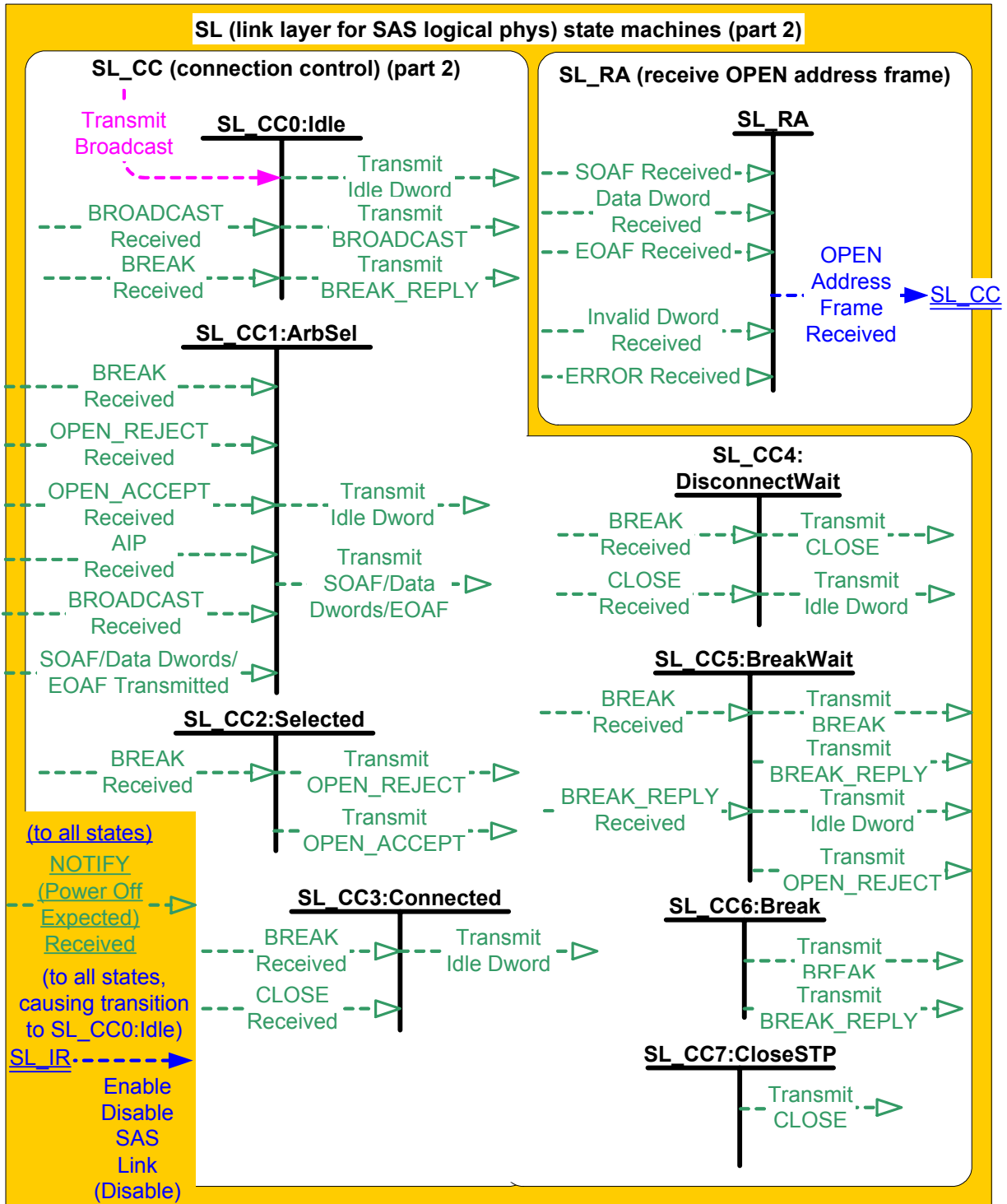


Figure 2 — SL (link layer for SAS phys) state machines (part 2)

Editor's Note 4: Figure 2 has changed. The changes are indicated by underlines on the confirmations, requests, and messages.

### 7.14.2 SL transmitter and receiver

The SL transmitter receives the following messages from the SL state machines specifying primitive sequences, frames, and dwords to transmit:

- a) Transmit Idle Dword;
- b) Transmit SOAF/Data Dwords/EOAF;
- c) Transmit OPEN\_ACCEPT;
- d) Transmit OPEN\_REJECT with an argument indicating the specific type (e.g., Transmit OPEN\_REJECT (Retry));
- e) Transmit BREAK;
- f) Transmit BREAK\_REPLY;
- g) Transmit BROADCAST; and
- h) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal)).

When the SL transmitter is requested to transmit a dword from any state within any of the SL state machines, it shall transmit that dword. If there are multiple requests to transmit, the following priority should be followed when selecting the dword to transmit:

- 1) BREAK\_REPLY;
- 2) BREAK;
- 3) CLOSE;
- 4) OPEN\_ACCEPT or OPEN\_REJECT;
- 5) SOAF or data dword or EOAF; then
- 6) idle dword.

When there is no outstanding message specifying a dword to transmit, the SL transmitter shall transmit idle dwords.

The SL transmitter sends the following messages to the SL state machines based on dwords that have been transmitted:

- a) SOAF/Data Dwords/EOAF Transmitted.

The SL receiver sends the following messages to the SL state machines indicating primitive sequences and dwords received from the SP\_DWS receiver (see 6.9.2):

- a) SOAF Received;
- b) Data Dword Received;
- c) EOAF Received;
- d) BROADCAST Received with an argument indicating the specific type (e.g., BROADCAST Received (Change));
- e) BREAK Received;
- f) BREAK\_REPLY Received;
- g) OPEN\_ACCEPT Received;
- h) OPEN\_REJECT Received with an argument indicating the specific type (e.g., OPEN\_REJECT Received (No Destination));
- i) AIP Received;
- j) CLOSE Received with an argument indicating the specific type (e.g., CLOSE Received (Normal));
- k) ERROR Received;
- l) [NOTIFY \(Power Loss Expected\) Received](#); and
- m) Invalid Dword Received.

The SL receiver shall ignore all other dwords.

### 7.14.3 SL\_RA (receive OPEN address frame) state machine

### 7.14.4 SL\_CC (connection control) state machine

#### 7.14.4.1 SL\_CC state machine overview

The state machine consists of the following states:

- a) SL\_CC0:Idle (see 7.14.4.2)(initial state);
- b) SL\_CC1:ArbSel (see 7.14.4.3);
- c) SL\_CC2:Selected (see 7.14.4.4);
- d) SL\_CC3:Connected (see 7.14.4.5);
- e) SL\_CC4:DisconnectWait (see 7.14.4.6);
- f) SL\_CC5:BreakWait (see 7.14.4.7);
- g) SL\_CC6:Break (see 7.14.4.8); and
- h) SL\_CC7:CloseSTP (see 7.14.4.9).

The state machine shall start in the SL\_CC0:Idle state. The state machine shall transition to the SL\_CC0:Idle state from any other state after receiving an Enable Disable SAS Link (Disable) message from the SL\_IR state machines (see 7.9.4).

The SL\_CC state machine receives the following messages from the SSP link layer state machine (see 7.16.8), the STP link layer state machine, and SMP link layer state machine (see 7.18.5):

- a) Request Break; and
- b) Request Close.

The SL\_CC state machine sends the following messages to the SSP link layer state machine, the STP link layer state machine, and SMP link layer state machine:

- a) Enable Disable SSP (Enable);
- b) Enable Disable SSP (Disable);
- c) Enable Disable STP (Enable);
- d) Enable Disable STP (Disable);
- e) Enable Disable SMP (Enable); and
- f) Enable Disable SMP (Disable).

The SL\_CC state machine receives the following messages from the SL\_IR state machines (see 7.9.4):

- a) Enable Disable SAS Link (Enable); and
- b) Enable Disable SAS Link (Disable).

Any message received by a state that is not referred to in the description of that state [or this subclause](#) shall be ignored.

[If this state machine receives an Accept\\_Reject Opens \(Accept SSP\) request, then this state shall set the Accept SSP Opens state machine variable to YES. If this state machine receives an Accept\\_Reject Opens \(Reject SSP\) request, then this state shall set the Accept SSP Opens state machine variable to NO.](#)

[If this state machine receives an Accept\\_Reject Opens \(Accept SMP\) request, then this state shall set the Accept SMP Opens state machine variable to YES. If this state machine receives an Accept\\_Reject Opens \(Reject SMP\) request, then this state shall set the Accept SMP Opens state machine variable to NO.](#)

[If this state machine receives an Accept\\_Reject Opens \(Accept STP\) request, then this state shall set the Accept STP Opens state machine variable to YES. If this state machine receives an Accept\\_Reject Opens \(Reject STP\) request, then this state shall set the Accept STP Opens state machine variable to NO.](#)

Any detection of an internal error shall cause the SL\_CC state machine to transition to the SL\_CC5:BreakWait state.

The SL\_CC state machine shall maintain the timers listed in table 6.

**Table 6 — SL\_CC timers**

Timer	Initial value
Open Timeout timer	1 ms
Close Timeout timer	1 ms
Break Timeout timer	1 ms

[The state machine shall maintain the state machine variables defined in table 7.](#)

**Table 7 — SL\_CC state machine variables**

State machine variable	Description
<a href="#">Accept SSP Opens</a>	<a href="#">Used to determine if the SCSI application layer is permitting SSP opens to be accepted on this phy.</a>
<a href="#">Accept SMP Opens</a>	<a href="#">Used to determine if the Management application layer is permitting SMP opens to be accepted on this phy.</a>
<a href="#">Accept STP Opens</a>	<a href="#">Used to determine if the ATA application layer is permitting STP opens to be accepted on this phy.</a>

**7.14.4.2 SL\_CC0:Idle state**

**7.14.4.2.1 State description**

This state is the initial state and is the state that is used when there is no connection pending or established.

Upon entry into this state, this state shall send:

- a) an Enable Disable SSP (Disable) message to the SSP link layer state machines;
- b) an Enable Disable SMP (Disable) message to the SMP link layer state machines;
- c) an Enable Disable STP (Disable) message to the STP link layer state machines; and
- d) a Connection Closed (Transition to Idle) confirmation to the port layer.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter (see 7.4).

If a BROADCAST Received (Change) message, BROADCAST Received (Reserved Change 0) message, or BROADCAST Received (Reserved Change 1) message is received, this state shall send a Change Received confirmation to the management layer.

If a Transmit Broadcast request is received with any argument, this state shall send a Transmit BROADCAST message with the same argument to the SL transmitter.

If a BREAK Received message is received and the BREAK\_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.12.5), then this state shall send a Transmit BREAK\_REPLY message to the SL transmitter.

[After this state receives an Enable Disable SAS Link \(Enable\) confirmation this state shall:](#)

- a) [set the Accept SSP Opens state machine variable to YES;](#)
- b) [set the Accept SMP Opens state machine variable to YES; and](#)
- c) [set the Accept STP Opens state machine variable to YES.](#)

[If a NOTIFY \(Power Loss Expected\) Received message is received then this state shall send a Notify Received \(Power Loss Expected\) confirmation to the port layer.](#)

**7.14.4.2.2 Transition SL\_CC0:Idle to SL\_CC1:ArbSel****7.14.4.2.3 Transition SL\_CC0:Idle to SL\_CC2:Selected****7.14.4.3 SL\_CC1:ArbSel state****7.14.4.3.1 State description****7.14.4.3.2 Transition SL\_CC1:ArbSel to SL\_CC0:Idle****7.14.4.3.3 Transition SL\_CC1:ArbSel to SL\_CC2:Selected****7.14.4.3.4 Transition SL\_CC1:ArbSel to SL\_CC3:Connected****7.14.4.3.5 Transition SL\_CC1:ArbSel to SL\_CC5:BreakWait**

This transition shall occur after receiving a SOAF/Data Dwords/EOAF Transmitted message if a BREAK Received message has not been received and after:

- a) a Stop Arb request is received and after sending an Open Failed (Port Layer Request) confirmation to the port layer; ~~or~~
- b) there is no response to the OPEN address frame before the Open Timeout timer expires and after sending an Open Failed (Open Timeout Occurred) confirmation to the port layer; or
- c) a NOTIFY (Power Loss Expected) Received message is received.

If a NOTIFY (Power Loss Expected) Received message was received this state shall include a Power Loss Expected argument.

**7.14.4.3.6 Transition SL\_CC1:ArbSel to SL\_CC6:Break****7.14.4.4 SL\_CC2:Selected state****7.14.4.4.1 State description**

This state completes the establishment of an SSP, SMP, or STP connection when an incoming connection request has won arbitration by sending a Transmit OPEN\_ACCEPT message, or rejects opening a connection by sending a Transmit OPEN\_REJECT message to the SL transmitter.

This state shall respond to an incoming OPEN address frame using the following rules:

- 1) If the OPEN address frame DESTINATION SAS ADDRESS field does not match the SAS address of this port, this state shall send a Transmit OPEN\_REJECT (Wrong Destination) message to the SL transmitter (see 7.14.4.4.2);
- 2) If the OPEN address frame INITIATOR PORT bit, PROTOCOL field, FEATURES field, and/or INITIATOR CONNECTION TAG field are set to values that are not supported (e.g., a connection request from an SMP target port), this state shall send a Transmit OPEN\_REJECT (Protocol Not Supported) message to the SL transmitter (see 7.14.4.4.2);
- 3) If the OPEN address frame CONNECTION RATE field is set to a connection rate that is not supported, this state shall send a Transmit OPEN\_REJECT (Connection Rate Not Supported) message to the SL transmitter (see 7.14.4.4.2);
- 4) If the OPEN address frame PROTOCOL field is set to STP, the source SAS address is not that of the STP initiator port with an affiliation established or the source SAS address is not that of an STP initiator port with task file register set resources (see 7.17.4), this state shall send a Transmit OPEN\_REJECT (STP Resources Busy) message to the SL transmitter (see 7.14.4.4.2);
- 5) If ~~an Accept\_Reject Opens (Reject SSP) request, Accept\_Reject Opens (Reject SMP) request, or Accept\_Reject Opens (Reject STP) request is received~~ the Accept SSP Opens state machine variable is set to NO, the Accept SMP Opens state machine variable is set to NO, or the Accept SMP Opens state machine variable is set to NO and the requested protocol is the corresponding protocol, this state shall send a Transmit OPEN\_REJECT (Retry) message to the SL transmitter (see 7.14.4.4.2);



- 6) If the requested protocol is SSP and ~~this state has not received an Accept\_Reject Opens (Reject-SSP) request~~ the Accept SSP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN\_ACCEPT message to the SL transmitter and send a Connection Opened (SSP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3);
- 7) If the requested protocol is SMP and ~~this state has not received an Accept\_Reject Opens (Reject-SMP) request~~ the Accept SMP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN\_ACCEPT message to the SL transmitter and send a Connection Opened (SMP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3); or
- 8) If the requested protocol is STP and ~~this state has not received an Accept\_Reject Opens (Reject-STP) request~~ the Accept STP Opens state machine variable is set to YES, then this state shall send a Transmit OPEN\_ACCEPT message to the SL transmitter and send a Connection Opened (STP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3).

If this state sends a Transmit OPEN\_REJECT message to the SL transmitter, it shall also send an Inbound Connection Rejected confirmation to the port layer.

NOTE 2 - Possible livelock scenarios occur if the BREAK\_REPLY method of responding to BREAK primitive sequences is disabled and a SAS phy transmits BREAK to abort a connection request (e.g., if its Open Timeout timer expires). SAS phys should respond to OPEN Address frames faster than 1 ms to reduce susceptibility to this problem.

#### 7.14.4.4.2 Transition SL\_CC2:Selected to SL\_CC0:Idle

#### 7.14.4.4.3 Transition SL\_CC2:Selected to SL\_CC3:Connected

#### 7.14.4.4.4 Transition SL\_CC2:Selected to SL\_CC6:Break

This transition shall occur after a BREAK Received message is received.

#### 7.14.4.4.5 Transition SL\_CC2:Selected to SL\_CC5:BreakWait

This transition shall occur after receiving a NOTIFY (Power Loss Expected) Received message and shall include a Power Loss Expected argument.

#### 7.14.4.5 SL\_CC3:Connected state

##### 7.14.4.5.1 State description

##### 7.14.4.5.2 Transition SL\_CC3:Connected to SL\_CC4:DisconnectWait

##### 7.14.4.5.3 Transition SL\_CC3:Connected to SL\_CC5:BreakWait

This transition shall occur after sending a Connection Closed (Break Requested) confirmation to the port layer if:

- a) a Request Break message is received; and a BREAK Received message has not been received; or
- b) a NOTIFY (Power Loss Expected) Received message is received.

If a NOTIFY (Power Loss Expected) Received message was received this state shall include a Power Loss Expected argument.

**7.14.4.5.4 Transition SL\_CC3:Connected to SL\_CC6:Break****7.14.4.5.5 Transition SL\_CC3:Connected to SL\_CC7:CloseSTP****7.14.4.6 SL\_CC4:DisconnectWait state****7.14.4.6.1 State description****7.14.4.6.2 Transition SL\_CC4:DisconnectWait to SL\_CC0:Idle****7.14.4.6.3 Transition SL\_CC4:DisconnectWait to SL\_CC5:BreakWait**

This transition shall occur if: [a NOTIFY \(Power Loss Expected\) Received message is received or if:](#)

- a) a BREAK Received message has not been received;
- b) no CLOSE Received message is received in response to a Transmit CLOSE message before the Close Timeout timer expires; and
- c) after sending a Connection Closed (Close Timeout) confirmation to the port layer.

[If a NOTIFY \(Power Loss Expected\) Received message was received this state shall include a Power Loss Expected argument.](#)

**7.14.4.6.4 Transition SL\_CC4:DisconnectWait to SL\_CC6:Break****7.14.4.7 SL\_CC5:BreakWait state****7.14.4.7.1 State description**

This state closes the connection if one is established and releases all resources associated with the connection.

Upon entry into this state, this state shall:

- 1) send a Transmit BREAK message to the SL transmitter; and
- 2) initialize and start the Break Timeout timer.

[If this state is entered with a Power Loss Expected message or this state receives a NOTIFY \(Power Loss Expected\) Received message then this state shall send a Notify Received \(Power Loss Expected\) confirmation to the port layer.](#)

If a BREAK Received message is received and the BREAK\_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.12.5), then this state shall send a Transmit BREAK\_REPLY message to the SL transmitter.

NOTE 3 - Some SAS phys compliant with previous versions of this standard send a Transmit OPEN\_REJECT (Retry) message to the SL transmitter in response to each OPEN Address Frame Received message received while in this state.

**7.14.4.7.2 Transition SL\_CC5:BreakWait to SL\_CC0:Idle****7.14.4.8 SL\_CC6:Break state****7.14.4.8.1 State description**

This state closes any connection and releases all resources associated with this connection.

Upon entry into this state:

- a) if the BREAK\_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.12.5), this state shall send a Transmit BREAK\_REPLY message to the SL transmitter (see 7.14.4.8.2); and
- b) if the BREAK\_REPLY method of responding to received BREAK primitive sequences is disabled (see 7.12.5), this state shall send a Transmit BREAK message to the SL transmitter.

[If this state receives a NOTIFY \(Power Loss Expected\) Received message then this state shall send a Notify Received \(Power Loss Expected\) confirmation to the port layer.](#)

**7.14.4.8.2 Transition SL\_CC6:Break to SL\_CC0:Idle**

**7.14.4.9 SL\_CC7:CloseSTP state**

**7.14.4.9.1 State description**

**7.14.4.9.2 Transition SL\_CC7:CloseSTP to SL\_CC0:Idle**

**7.16.8 SSP (link layer for SSP phys) state machines**

**7.16.8.1 SSP state machines overview**

The SSP link layer contains several state machines that run in parallel to control the flow of dwords on the physical link during an SSP connection. The SSP state machines are as follows:

- a) SSP\_TIM (transmit interlocked frame monitor) state machine (see 7.16.8.3);
- b) SSP\_TCM (transmit frame credit monitor) state machine (see 7.16.8.4);
- c) SSP\_D (DONE control) state machine (see 7.16.8.5);
- d) SSP\_TF (transmit frame control) state machine (see 7.16.8.6);
- e) SSP\_RF (receive frame control) state machine (see 7.16.8.7);
- f) SSP\_RCM (receive frame credit monitor) state machine (see 7.16.8.8);
- g) SSP\_RIM (receive interlocked frame monitor) state machine (see 7.16.8.9);
- h) SSP\_TC (transmit credit control) state machine (see 7.16.8.10); and
- i) SSP\_TAN (transmit ACK/NAK control) state machine (see 7.16.8.11).

All the SSP state machines shall start after receiving an Enable Disable SSP (Enable) message from the SL state machines (see 7.14).

All the SSP state machines shall terminate after:

- a) receiving an Enable Disable SSP (Disable) message from the SL state machines;
- b) receiving a Request Close message from the SSP\_D state machine indicating that the connection has been closed;
- c) [receiving a NOTIFY \(Power Loss Expected\) Received message from the SP\\_DWS receiver \(see 6.9.2\)](#); or
- d) receiving a Request Break message from the SSP\_D state machine indicating that a BREAK has been transmitted.

If a state machine consists of multiple states the initial state is as indicated in the state machine description in this subclause.

The SSP state machines shall maintain the timers listed in table 8.

**Table 8 — SSP link layer timers**

Timer	Initial value	State machine	Reference
ACK/NAK Timeout timer	1 ms	SSP_TIM	7.16.8.3
DONE Timeout timer	1 ms	SSP_D	7.16.8.5
Credit Timeout timer	1 ms	SSP_TF	7.16.8.6

Figure 3 shows the SSP state machines and states related to frame transmission.

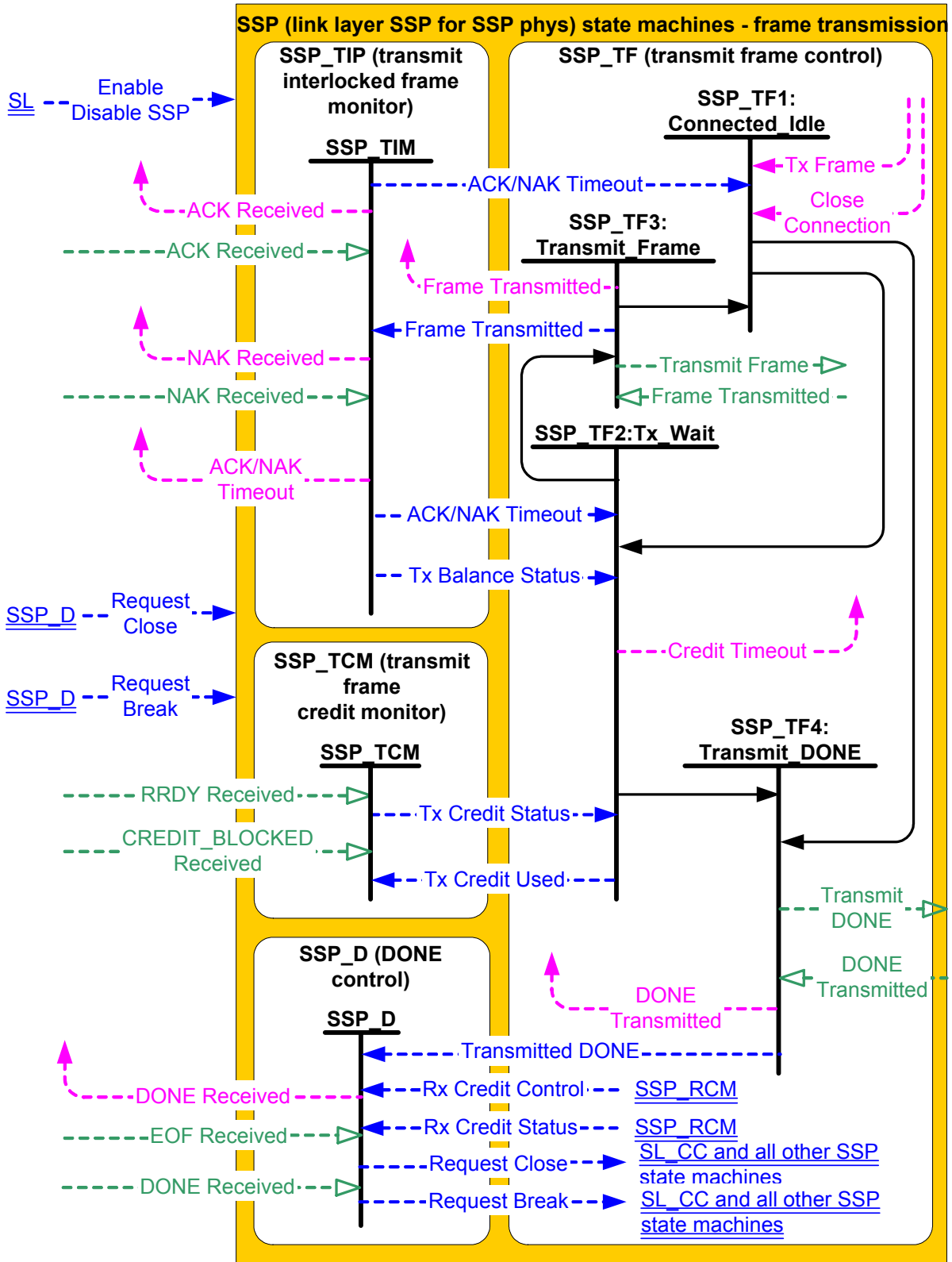


Figure 3 — SSP (link layer for SSP phys) state machines (part 1 - frame transmission)

Figure 4 shows the SSP state machines and states related to frame reception.

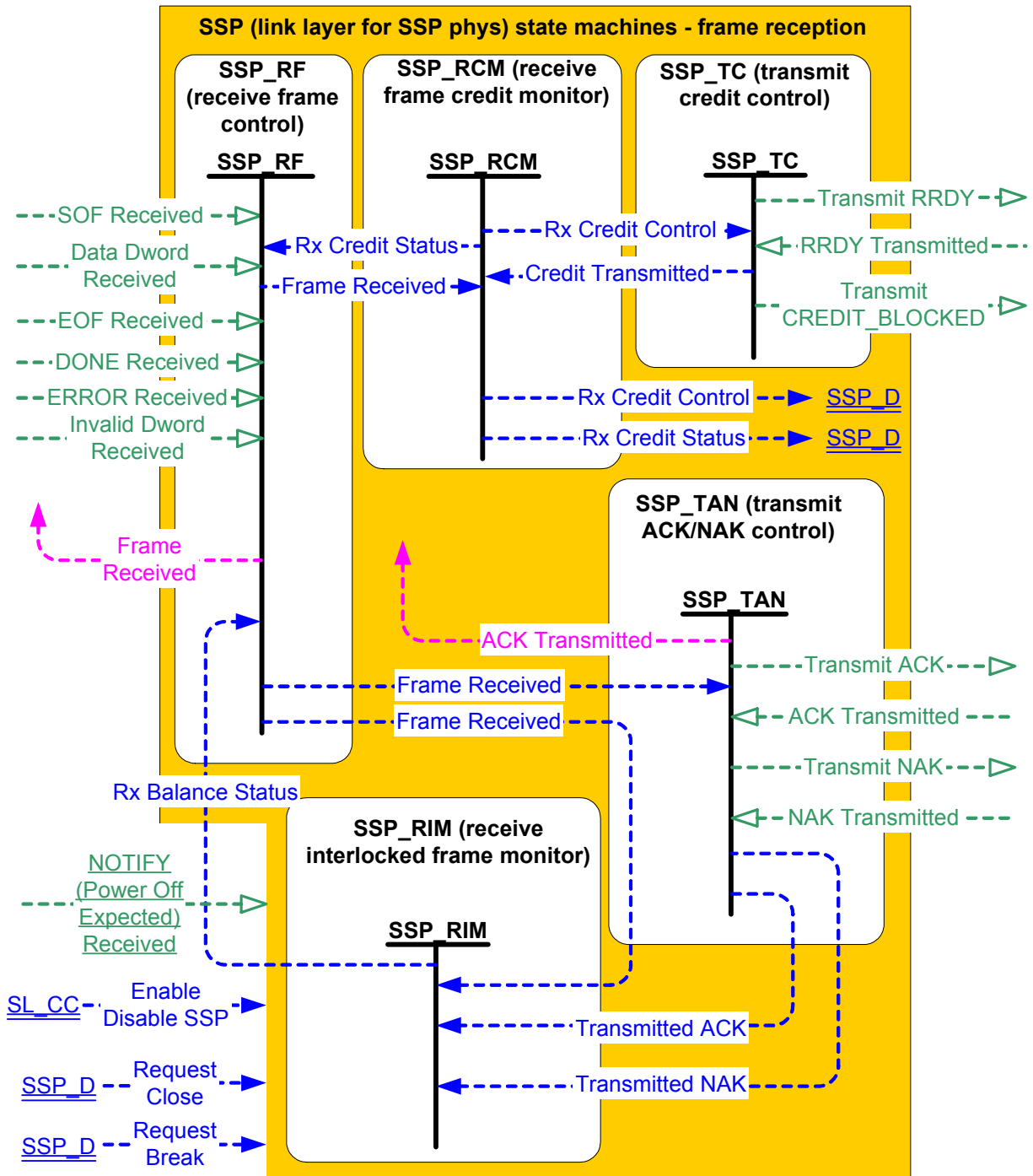


Figure 4 — SSP (link layer for SSP phys) state machines (part 2 - frame reception)

---



---

Editor's Note 5: Figure 4 has changed. The changes are indicated by underlines on the confirmations, requests, and messages.

---



---

### 7.16.8.2 SSP transmitter and receiver

The SSP transmitter receives the following messages from the SSP state machines specifying primitive sequences and frames to transmit:

- a) Transmit RRDY with an argument indicating the specific type (e.g., Transmit RRDY (Normal));
- b) Transmit CREDIT\_BLOCKED;
- c) Transmit ACK;
- d) Transmit NAK with an argument indicating the specific type (e.g., Transmit NAK (CRC Error));
- e) Transmit Frame (i.e., SOF/data dwords/EOF); and
- f) Transmit DONE with an argument indicating the specific type (e.g., Transmit DONE (Normal)).

The SSP transmitter sends the following messages to the SSP state machines based on dwords that have been transmitted:

- a) DONE Transmitted;
- b) RRDY Transmitted;
- c) CREDIT\_BLOCKED Transmitted;
- d) ACK Transmitted;
- e) NAK Transmitted; and
- f) Frame Transmitted.

When there is no outstanding message specifying a dword to transmit, the SSP transmitter shall transmit idle dwords.

The SSP receiver sends the following messages to the SSP state machines indicating primitive sequences and dwords received from the SP\_DWS receiver (see 6.9.2):

- a) ACK Received;
- b) NAK Received;
- c) RRDY Received;
- d) CREDIT\_BLOCKED Received;
- e) DONE Received with an argument indicating the specific type (e.g., DONE Received (Normal));
- f) SOF Received;
- g) Data Dword Received;
- h) EOF Received;
- i) [NOTIFY \(Power Loss Expected\) Received](#);
- j) ERROR Received; and
- k) Invalid Dword Received.

The SSP receiver shall ignore all other dwords.

### 7.16.8.3 SSP\_TIM (transmit interlocked frame monitor) state machine

#### 8.2.2 PL\_OC (port layer overall control) state machine

##### 8.2.2.1 PL\_OC state machine overview

A PL\_OC state machine:

- a) receives requests from the SSP, SMP, and STP transport layers;
- b) sends messages to the PL\_PM state machine;
- c) receives messages from the PL\_PM state machine;
- d) selects frames to transmit;
- e) selects phys on which to transmit frames;
- f) receives confirmations from the link layer;
- g) sends confirmations to the transport layer;
- h) has Arbitration Wait Time timers; and
- i) has I\_T Nexus Loss timers.

This state machine consists of the following states:

- a) PL\_OC1:Idle (see 8.2.2.2) (initial state); and

b) PL\_OC2:Overall\_Control (see 8.2.2.3).

After power on this state machine shall start in the PL\_OC1:Idle state.

The PL\_OC state machine shall maintain:

- a) a pool of pending Tx Frame messages for each destination SAS address; and
- b) as many pending Tx Open message slots as there are phys in the port.

The PL\_OC state machine shall maintain the timers listed in table 9.

**Table 9 — PL\_OC state machine timers**

Timer	Maximum number of timers	Initial value
I_T Nexus Loss timer	One per destination SAS address	Depending on the protocol used by the port: <ul style="list-style-type: none"> <li>a) for SSP target ports, the value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page (see 10.2.7.2);</li> <li>b) for SSP initiator ports, the value in the I_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page for the SSP target port with that destination SAS address (see 10.2.7.2);</li> <li>c) for STP target ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function (see 10.4.3.15);</li> <li>d) for STP initiator ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function (see 10.4.3.15) for the STP target port with that destination SAS address; or</li> <li>e) for SMP initiator ports, the value in the STP SMP I_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function (see 10.4.3.15).</li> </ul>
Arbitration Wait Time timer	One per pending Tx Open message	0000h, a vendor-specific value less than 8000h (see 7.12.3), or the value received with a Retry Open message.

Figure 5 shows the PL\_OC state machine.

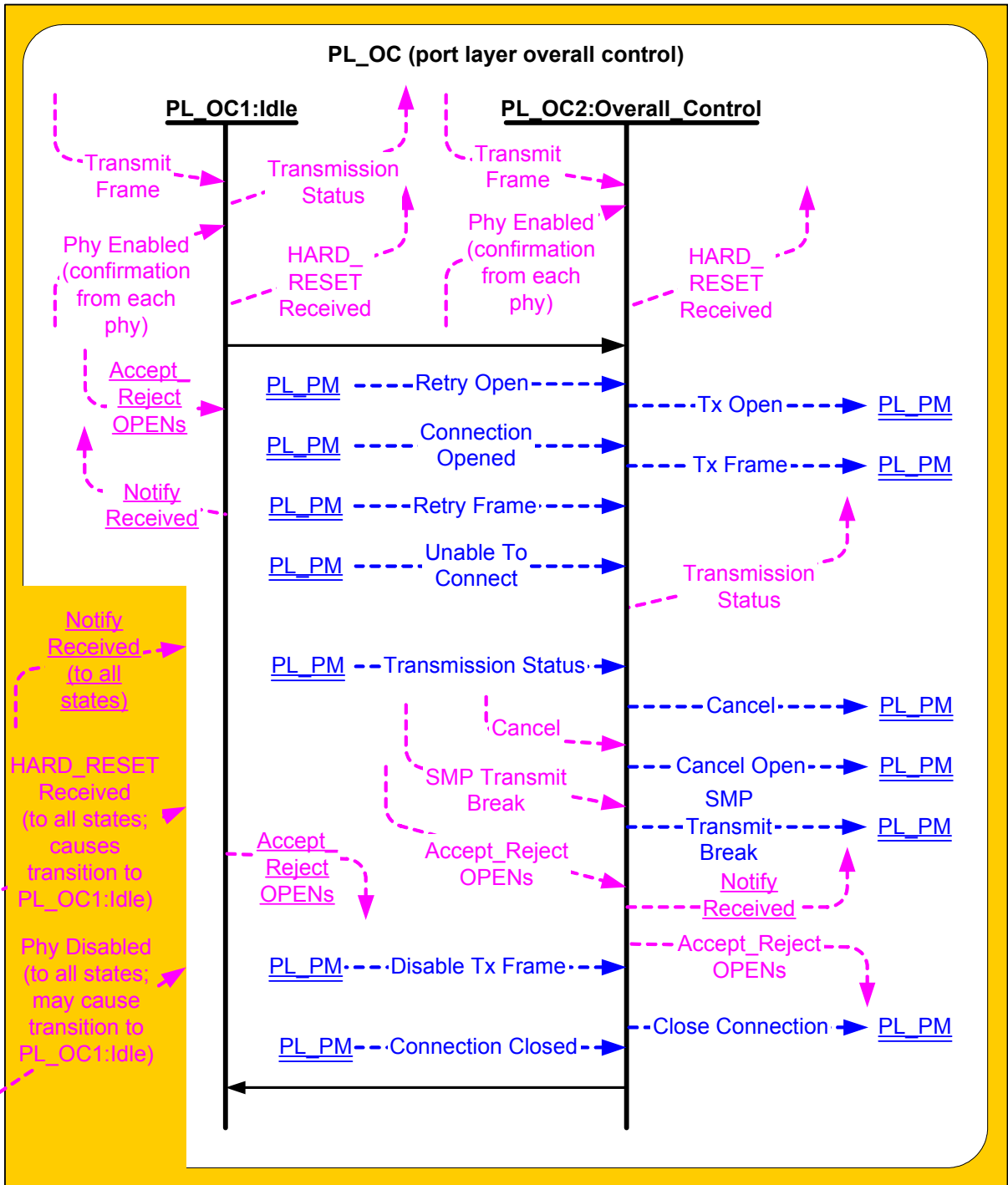


Figure 5 — PL\_OC (port layer overall control) state machine

---

Editor's Note 6: Figure 3 has changed. The changes are indicated by underlines on the confirmations, requests, and messages.

---



### 8.2.2.2 PL\_OC1:Idle state

#### 8.2.2.2.1 PL\_OC1:Idle state description

This state is the initial state of the PL\_OC state machine.

If this state receives a HARD\_RESET Received confirmation, then this state shall send a HARD\_RESET Received confirmation to the transport layer.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall send a Notify Received (Power Loss Expected) confirmation to the transport layer.

If this state receives an Accept\_Reject Opens request, then this state shall send an Accept\_Reject Opens request to all link layers in the port.

If this state receives a Transmit Frame request, then this state shall send a Transmission Status (No Phys In Port) confirmation to the transport layer.

If an I\_T Nexus Loss timer expires for a destination SAS address, this state shall perform the following:

- a) delete the I\_T Nexus Loss timer for the SAS address;
- b) send a Transmission Status (I\_T Nexus Loss) confirmation for each pending Tx Frame message for the SAS address; and
- c) discard each pending Tx Frame message for the SAS address and any corresponding pending Tx Open messages.

If the port is an STP target port or an STP initiator port, the port shall handle all pending commands as described in 4.5.

#### 8.2.2.2.2 Transition PL\_OC1:Idle to PL\_OC2:Overall\_Control

### 8.2.2.3 PL\_OC2:Overall\_Control state

#### 8.2.2.3.1 PL\_OC2:Overall\_Control state overview

This state may receive Transmit Frame requests from the transport layers (i.e., SSP and SMP) and Retry frame messages from PL\_PM state machines. This state shall create a pending Tx Frame message for each received Transmit Frame request and Retry Frame message. There may be more than one pending Tx Frame message at a time for each SSP transport layer. There shall be only one pending Tx Frame message at a time for each SMP transport layer.

This state selects PL\_PM state machines through which connections are established. This state shall only attempt to establish connections through PL\_PM state machines whose phys are enabled. In a vendor-specific manner, this state selects PL\_PM state machines on which connections are established to transmit frames. This state shall receive a response to a message from a PL\_PM state machine before sending another message to that PL\_PM state machine.

This state also:

- a) receives connection management requests from the transport layers;
- b) sends connection management messages to PL\_PM state machines;
- c) receives connection management messages from PL\_PM state machines; and
- d) sends connection management confirmations to the transport layers.

After receiving a Transmit Frame request for a destination SAS address for which there is no connection established and for which no I\_T Nexus Loss timer has been created, this state shall create an I\_T Nexus Loss timer for that SAS address if:

- a) the protocol is SSP, the port is an SSP target port, the Protocol-Specific Port mode page is implemented, and the I\_T NEXUS LOSS TIME field in the Disconnect-Reconnect mode page (see 10.2.7.1) is not set to 0000h;
- b) the protocol is STP, the port is an STP target port, and the STP SMP I\_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function is not set to 0000h; or

- c) the protocol is SMP, the port is an SMP initiator port, and the STP SMP I\_T NEXUS LOSS TIME field in the SMP CONFIGURE GENERAL function is not set to 0000h.

This state may create an I\_T Nexus Loss timer for that SAS address if:

- a) the protocol is SSP and the port is an SSP initiator port; or;
- b) the protocol is STP and the port is an STP initiator port.

When this state creates an I\_T Nexus Loss timer it shall:

- 1) initialize the I\_T Nexus Loss timer as specified in table 9 (see 8.2.2.1); and
- 2) not start the I\_T Nexus Loss timer.

If this state machine is in an SSP initiator port, then this state may create an I\_T Nexus Loss timer for the SAS address. If a state machine in an SSP initiator port and creates an I\_T Nexus Loss timer, then the state machine should use the value in the I\_T NEXUS LOSS TIME field in the Protocol-Specific Port mode page for the SSP target port (see 10.2.7.2) as the initial value for its I\_T Nexus Loss timer.

If there are no pending Tx Frame messages for a destination SAS address and an I\_T Nexus Loss timer has been created for that destination SAS address, then this state shall delete the I\_T Nexus Loss timer for that destination SAS address.

If this state receives a HARD\_RESET Received confirmation, then this state shall discard all pending Tx Frame messages and delete all I\_T Nexus Loss timers and send a HARD\_RESET Received confirmation to the transport layer.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall:

- a) discard all pending Tx Frame messages, if any;
- b) delete all I\_T Nexus Loss timers, if any;
- c) send a Close Connection message to all the PL\_PM state machines;
- d) send a Cancel Open message to all the PL\_PM state machines; and
- e) send a Notify Received (Power Loss Expected) confirmation to the transport layer.

#### 8.2.2.3.2 PL\_OC2:Overall\_Control state establishing connections

#### 8.2.2.3.3 PL\_OC2:Overall\_Control state connection established

#### 8.2.2.3.4 PL\_OC2:Overall\_Control state unable to establish a connection

#### 8.2.2.3.5 PL\_OC2:Overall\_Control state connection management

If this state receives an Accept\_Reject Opens request, then this state shall send an Accept\_Reject Opens request to all link layers ~~message to all phys~~ in the port.

If this state receives an SMP Transmit Break request, then this state shall send an SMP Transmit Break message to the PL\_PM state machine associated with the corresponding SMP transport state machine. If there is no PL\_PM state machine associated with the request, the PM\_OC state shall ignore the request.

If this state receives one of the following:

- a) a Connection Closed (Close Timeout) message;
- b) a Connection Closed (Break Requested) message; or
- c) a Connection Closed (Break Received) message,

then this state shall not send a Tx Open or Tx Frame message to the PL\_PM state machine that sent the message until this state receives a Connection Closed (Transition to Idle) message from that PL\_PM state machine.

If this state receives a Connection Closed (Normal) message or a Connection Closed (Transition to Idle) message indicating that a connection with a destination SAS address is no longer open and this state has pending Tx Open messages, then this state may send a Tx Open message to the PL\_PM state machine that sent the Connection Closed message.

If this port is a wide SSP port, then this state shall not reject an incoming connection request on one phy because it has an outgoing connection request on another phy.

If this port is an SSP port, there are no pending Tx Frame messages for a destination SAS address with which a PL\_PM state machine has established a connection, and the connection was established by a message from this state, then this state should send a Close Connection message to the PL\_PM state machine.

If this port is an SSP port, has no pending Tx Frame messages for a destination SAS address with which a PL\_PM state machine has established a connection, and the connection was established by the destination, then this state may wait a vendor-specific time and then shall send a Close Connection message to the PL\_PM state machine.

If this state has received a Disable Tx Frame message from a PL\_PM state machine, then this state should send a Close Connection message to the PL\_PM state machine.

NOTE 4 - The PL\_PM state machine sends a Close Connection request to the link layer upon receipt of a Close Connection message or on expiration of the Bus Inactivity Time Limit timer (see 8.2.3.4.1).

**8.2.2.3.6 PL\_OC2:Overall\_Control state frame transmission**

**8.2.2.3.7 PL\_OC2:Overall\_Control state frame transmission cancellations**

**8.2.2.3.8 Transition PL\_OC2:Overall\_Control to PL\_OC1:Idle**

This transition shall occur after:

- a) sending a HARD\_RESET Received confirmation to the transport layer; ~~or~~
- b) a Phy Disabled confirmation is received from all of the link layers in the port; or
- c) [sending a Notify Received \(Power Loss Expected\) confirmation to the transport layer.](#)

**9.2.6.3 ST\_T (transport layer for SSP target ports) state machines**

**9.2.6.3.1 ST\_T state machines overview**

The ST\_T state machines are as follows:

- a) ST\_TFR (target frame router) state machine (see 9.2.6.3.2); and
- b) ST\_TTS (target transport server) state machine (see 9.2.6.3.3).

The SAS target port includes:

- a) one ST\_TFR state machine; and
- b) one ST\_TTS state machine for each possible task and task management function (i.e., for each tag).

The ST\_TTS state machine may maintain the timers listed in table 10.

**Table 10 — ST\_T state machine timers**

Timer	Initial value
Initiator Response Timeout	The value in the INITIATOR RESPONSE TIMEOUT field in the Protocol-Specific Port mode page (see 10.2.7.2).

Figure 6 shows the ST\_T state machines.

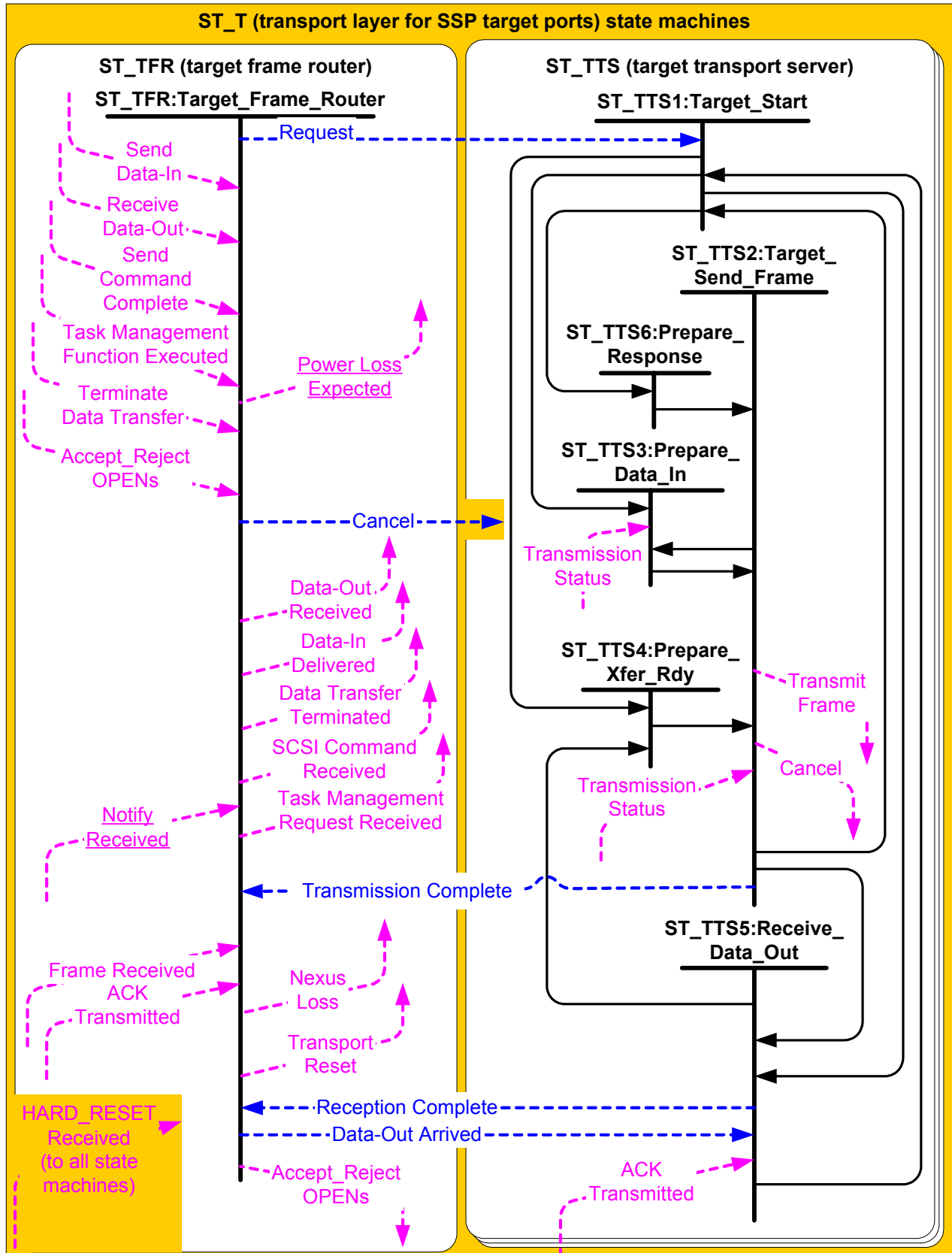


Figure 6 — ST\_T (transport layer for SSP target ports) state machines

---



---

Editor's Note 7: Figure 4 has changed. The changes are indicated by underlines on the confirmations, requests, and messages.

---



---

### 9.2.6.3.2 ST\_TFR (target frame router) state machine

#### 9.2.6.3.2.1 ST\_TFR state machine overview

The ST\_TFR state machine performs the following functions:

- a) receives confirmations from the port layer;
- b) receives transport protocol service requests from the SCSI application layer;
- c) sends transport protocol service indications to the SCSI application layer;
- d) sends messages to the ST\_TTS state machine;
- e) receives messages from the ST\_TTS state machine;
- f) receives Accept\_Reject OPENs requests from the SCSI application layer;
- g) sends Accept\_Reject OPENs requests to the port layer;
- h) sends Nexus Loss event notifications to the SCSI application layer; ~~and~~
- i) sends Transport Reset event notifications to the SCSI application layer; and
- j) sends Power Loss Expected event notifications to the SCSI application layer.

This state machine consists of one state.

This state machine shall be started after power on.

#### 9.2.6.3.2.2 Processing Frame Received confirmations

If this state machine receives a Frame Received (ACK/NAK Balanced) or Frame Received (ACK/NAK Not Balanced) confirmation, then this state machine shall check the frame type in the received frame (see table 135 in 9.2.1). If the frame type is not COMMAND, TASK, or DATA, then this state machine shall discard the frame. If the confirmation was Frame Received (ACK/NAK Not Balanced) and the frame type is not DATA, then this state machine shall discard the frame.

This state machine may check that reserved fields in the received frame are zero. If any reserved fields are checked and they are not set to zero, then this state machine shall send the following to an ST\_TTS state machine that does not have an active task and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the destination SAS address argument set to the SAS address from which the invalid frame was received; and
- c) the Service Response argument set to Invalid Frame.

The check of reserved fields within the frame shall not apply to the reserved fields within the CDB in a COMMAND frame. Checking of reserved fields in a CDB is described in SPC-4.

The following is the list of Transport Response arguments:

- a) connection rate;
- b) initiator connection tag;
- c) destination SAS address (i.e., the SAS address to which the RESPONSE frame is to be transmitted);
- d) source SAS address set to the SAS address of the SAS port containing the state machine;
- e) tag; and
- f) service response.

If the frame type is correct relative to the Frame Received confirmation, then this state machine may check that the hashed source SAS address matches the SAS address of the SAS port that transmitted the frame and that the hashed destination SAS address matches the SAS address of the SAS port that received the frame based on the connection information. If this state machine checks these SAS addresses, and they do not match, then this state machine shall discard the frame.

If the frame type is COMMAND or TASK then this state machine shall check the length of the information unit. If the length of the information unit is not correct, then this state machine shall send the following to an ST\_TTS state machine that does not have an active task and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the destination SAS address set to the SAS address from which the invalid frame was received; and
- c) the Service Response argument set to Invalid Frame.

If the frame type is TASK, this state machine checks tags, the RETRANSMIT bit in the new TASK frame is set to one, and the tag for the new TASK frame is the same as the tag for a previous TASK frame where the task management function for the previous TASK frame is not complete, then this state machine shall discard the new TASK frame and not send a Task Management Request Received confirmation to the port layer.

If the frame type is TASK and this state machine does not check tags, then this state machine shall ignore the RETRANSMIT bit.

If the frame type is COMMAND or TASK, then this state machine may check the target port transfer tag. If this state checks the target port transfer tag and the tag is set to a value other than FFFFh, then this state machine shall send the following to an ST\_TTS state machine that does not have an active task and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the destination SAS address set to the SAS address from which the invalid frame was received; and
- c) the Service Response argument set to Invalid Frame.

If the frame type is TASK, then this state machine shall check the logical unit number. If the logical unit number is unknown, then this state machine shall send the following to an ST\_TTS state machine that does not have an active task and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the destination SAS address set to the SAS address from which the invalid frame was received; and
- c) the Service Response argument set to Incorrect Logical Unit Number.

If the frame type is DATA and this frame is for first burst data or this state machine did not assign a target port transfer tag for the data transfer, then this state machine may check the target port transfer tag. If target port transfer tag is set to a value other than FFFFh, then this state machine shall send the following to an ST\_TTS state machine that does not have an active task and discard the frame:

- a) a Request (Send Transport Response) message with the Transport Response arguments;
- b) the destination SAS address set to the SAS address from which the invalid frame was received; and
- c) the Service Response argument set to Invalid Frame.

If the frame type is COMMAND or TASK and the fields checked in the frame are correct, then this state machine shall wait to receive an ACK Transmitted confirmation.

If the frame type is COMMAND, the fields checked in the frame are correct, and this state machine receives an ACK Transmitted confirmation, then this state machine shall send a SCSI Command Received transport protocol service indication with the following arguments to the SCSI application layer:

- a) source SAS address (i.e., the SAS address that transmitted the COMMAND frame);
- b) tag;
- c) logical unit number;
- d) task attribute;
- e) task priority;
- f) CDB; and
- g) additional CDB bytes, if any.

If the frame type is TASK, the fields checked in the frame are correct, and this state machine receives an ACK Transmitted confirmation, then this state machine shall send a Task Management Request Received transport protocol service indication with the following arguments to the SCSI application layer:

- a) source SAS address (i.e., the SAS address that transmitted the TASK frame);
- b) tag;
- c) logical unit number;
- d) task management function; and

- e) tag of the task to be managed.

If the frame type is DATA, and the tag does not match a tag for an outstanding command performing write operations, then this state machine shall discard the frame.

If the frame type is DATA, and the tag matches a tag for an outstanding command performing write operations when first burst is disabled or for which no Transmission Complete (Xfer\_Rdy Delivered) message has been received from an ST\_TTS state machine, then this state machine shall discard the frame.

If the frame type is DATA and a target port transfer tag was received in a Transmission Complete (Xfer\_Rdy Delivered) message, then this state machine shall check the target port transfer tag. If the target port transfer tag received in the DATA frame does not match the Target Transport Tag argument in the Transmission Complete (Xfer\_Rdy Delivered) message, then this state machine shall discard the frame.

If the frame type is DATA and the fields checked in the frame are correct, and first burst is enabled or this state machine has received a Transmission Complete (Xfer\_Rdy Delivered) from the ST\_TTS state machine for the request, then this state machine shall send a Data-Out Arrived message to the ST\_TTS state machine specified by the tag in the frame. The message shall include the content of the write DATA frame.

If this state receives a Notify Received (Power Loss Expected) confirmation, then this state shall:

- a) send a Cancel message to all the ST\_TTS state machines; and
- b) send a Power Loss Expected confirmation to the SCSI application layer.

### 10.2.5 SCSI transport protocol event notifications

Table 11 lists the SCSI transport protocol event notifications supported by this standard.

**Table 11 — SCSI transport protocol events**

Event notification	SAS SSP implementation
Transport Reset	Receipt of a hard reset sequence (see 4.4.2)
Nexus Loss	Receipt of specific OPEN_REJECTs for a specific time period (see 4.5).
<u>Power Loss Expected</u>	<u>Receipt of a NOTIFY (Power Loss Expected) (see 7.2.5.10.3).</u>

## 10.3 SAM-4 changes

### 5.6 Aborting tasks

#### 5.6.1 Mechanisms that cause tasks to be aborted

A task is aborted when an event or SCSI initiator device action causes termination of the task prior to its successful completion.

The following events cause a task or several tasks to be aborted:

- a) The return of an **Execute Command** service response of SERVICE DELIVERY OR TARGET FAILURE as described in 5.1;
- b) An I\_T nexus loss (see 6.3.4);
- c) A logical unit reset (see 6.3.3);
- d) A hard reset (see 6.3.2);
- e) A power on condition (see 6.3.1); ~~or~~
- f) A power loss expected (see 6.3.5); or
- g) SCSI transport protocol specific conditions.

An action transmitted via one I\_T nexus may abort task(s) received on that I\_T nexus and/or task(s) received on other I\_T nexuses.

The following actions affect only the task(s) received on the I\_T nexus on which the action is transmitted:

- a) Completion of an ABORT TASK task management function directed to the specified task;
- b) Completion of an ABORT TASK SET task management function under the conditions specified in 7.3;
- c) Completion of an I\_T NEXUS RESET task management function; or
- d) Completion of a command with a CHECK CONDITION status, without establishing an ACA condition (see 5.8.1.3) or establishing an ACA condition (see 5.8.2.2), while the Control mode page (see SPC-3) contains fields that are set as follows:
  - A) The QERR field set to 01b and the TST field set to 001b; or
  - B) The QERR field set to 11b.

The actions shown in table 12 affect the task(s) received on the I\_T nexus on which the action is transmitted and/or task(s) received on other I\_T nexuses.

**Table 12 — Actions that affect task(s) received on this or other I\_T nexuses**

Action	Unit attention additional sense code, if any (see 5.6.3)
Completion of a CLEAR TASK SET task management function referencing the task set containing the specified task	COMMANDS CLEARED BY ANOTHER INITIATOR
Completion of a command with a CHECK CONDITION status, with or without establishing an ACA condition, and the QERR field was set to 01b and the TST field was set to 000b in the Control mode page (see SPC-3)	COMMANDS CLEARED BY ANOTHER INITIATOR
Completion of a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action with a reservation key that is associated with the I_T nexus on which the task was received (see SPC-3)	COMMANDS CLEARED BY ANOTHER INITIATOR
Completion of a LOGICAL UNIT RESET task management function (see 7.7) directed to the logical unit	BUS DEVICE RESET FUNCTION OCCURRED
Receipt of a <a href="#">Power Loss Expected</a> indication (see 6.3.5).	<a href="#">COMMANDS CLEARED BY POWER LOSS NOTIFICATION</a>
SCSI transport protocol specific conditions	As defined by the applicable SCSI transport protocol standard

If one or more tasks are cleared or aborted, the affected tasks are also cleared from the SCSI initiator ports in a manner that is outside the scope of this standard.



**5.6.2 When a SCSI initiator port aborts tasks received on its own I\_T nexus**

**5.6.3 When a SCSI initiator port aborts tasks received on other I\_T nexuses**

## **5.8 Command processing considerations and exception conditions**

**5.8.1 Commands that complete with CHECK CONDITION status**

**5.8.1.1 Overview**

**5.8.1.2 Handling tasks when ACA is not in effect**

**5.8.1.3 Aborting other tasks when CHECK CONDITION status is returned without establishing an ACA**

**5.8.2 Auto contingent allegiance (ACA)**

**5.8.2.1 ACA Overview**

**5.8.2.2 Establishing an ACA**

**5.8.2.3 Handling new tasks received on the faulted I\_T nexus when ACA is in effect**

**5.8.2.4 Handling new tasks received on non-faulted I\_T nexuses when ACA is in effect**

**5.8.2.4.1 Command processing permitted for tasks received on non-faulted I\_T nexuses during ACA**

**5.8.2.4.2 Handling new tasks received on non-faulted I\_T nexuses when ACA is in effect**

**5.8.2.5 Clearing an ACA condition**

An ACA condition shall only be cleared:

- a) As a result of a hard reset (see 6.3.2), logical unit reset (see 6.3.3), ~~or~~ I\_T nexus loss (see 6.3.4), [or power loss expected \(see 6.3.5\)](#);
- b) By a CLEAR ACA task management function (see 7.4) received on the faulted I\_T nexus;
- c) By a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action with the ACA task attribute received on the faulted I\_T nexus that clears the tasks received on the faulted I\_T nexus (see SPC-3);
- d) By a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action with a task attribute other than ACA task attribute received on a non-faulted I\_T nexus that clears the tasks received on the faulted I\_T nexus;
- e) When a command with the ACA task attribute received on the faulted I\_T nexus terminates with a CHECK CONDITION status; or
- f) When a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action terminates in a CHECK CONDITION status.

Cases e) and f) may result in the establishment of a new ACA based on the value of the NACA bit.

When an ACA condition is cleared and no new ACA condition is established, the state of all tasks in the task set shall be modified as described in 8.8.

### 5.8.3 Overlapped commands

### 5.8.4 Incorrect logical unit selection

### 5.8.5 Task attribute exception conditions

### 5.8.6 Sense data

Sense data shall be made available by the logical unit in the event a command completes with a CHECK CONDITION status or other conditions (e.g., the processing of a REQUEST SENSE command). The format, content, and conditions under which sense data shall be prepared by the logical unit are specified in this standard, SPC-3, the applicable command standard, and the applicable SCSI transport protocol standard.

Sense data associated with an I\_T nexus shall be preserved by the logical unit until:

- a) The sense data is transferred;
- b) A logical unit reset (see 6.3.3) occurs; ~~or~~
- c) [A power loss expected occurs; or](#)
- d) An I\_T nexus loss (see 6.3.4) occurs for the I\_T nexus associated with the preserved sense data.

When a command completes with a CHECK CONDITION status, sense data shall be returned in the same I\_T\_L\_Q nexus transaction (see 3.1.53) as the CHECK CONDITION status. After the sense data is returned, it shall be cleared except when it is associated with a unit attention condition and the UA\_INTLCK\_CTRL field in the Control mode page (see SPC-3) contains 10b or 11b.

The return of sense data in the same I\_T\_L\_Q nexus transaction as a CHECK CONDITION status shall not affect ACA (see 5.8.2) or the sense data associated with a unit attention condition when the UA\_INTLCK\_CTRL field contains 10b or 11b.

### 5.8.7 Unit Attention condition

Each logical unit shall generate a unit attention condition whenever one of the following events occurs:

- a) A hard reset (see 6.3.2), logical unit reset (see 6.3.3), ~~or~~ I\_T nexus loss (see 6.3.4) ~~occurs, or power loss expected (see 6.3.5);~~
- b) A removable medium may have been changed;
- c) The mode parameters associated with this I\_T nexus have been changed by a task received on another I\_T nexus (i.e., SCSI initiator ports share mode parameters, see SPC-3);
- d) The log parameters associated with this I\_T nexus have been changed by a task received on another I\_T nexus (i.e., SCSI initiator ports share log parameters, see SPC-3);
- e) The version or level of microcode has been changed (see SPC-3);
- f) Tasks received on this I\_T nexus have been cleared by a task or a task management function associated with another I\_T nexus and the TAS bit was set to zero in the Control mode page associated with this I\_T nexus (see SPC-3);
- g) INQUIRY data has been changed (see SPC-3);
- h) The logical unit inventory has been changed (see SPC-3);
- i) The mode parameters in effect for the associated I\_T nexus have been restored from non-volatile memory (see SPC-3); or
- j) Any other event requiring the attention of the SCSI initiator device.

Logical units may queue unit attention conditions. After the first unit attention condition is cleared, another unit attention condition may exist (e.g., a unit attention condition with an additional sense code set to POWER ON OCCURRED may be followed by one with an additional sense code set to MICROCODE HAS BEEN CHANGED).

A unit attention condition shall persist on the logical unit for the SCSI initiator port associated with each I\_T nexus until the SCSI initiator port associated with the I\_T nexus clears the condition as described in the remainder of this subclause.

If an INQUIRY command enters the enabled task state, the logical unit shall perform the INQUIRY command and shall neither report nor clear any unit attention condition.

If a REPORT LUNS command enters the enabled task state, the logical unit shall perform the REPORT LUNS command and shall not report any unit attention condition. For each logical unit accessible by the I\_T nexus on which the REPORT LUNS command was received, any pending unit attention condition established for the SCSI initiator port associated with that I\_T nexus as a result of a change in the logical unit inventory shall be cleared. Other pending unit attention conditions shall not be cleared.

If a REQUEST SENSE command enters the enabled task state while a unit attention condition exists for the SCSI initiator port associated with the I\_T nexus on which the REQUEST SENSE command was received, then the logical unit shall return GOOD status and either:

- a) Report any pending sense data as parameter data and preserve all unit attention conditions on the logical unit; or,
- b) Report a unit attention condition as parameter data for the REQUEST SENSE command to the SCSI initiator port associated with the I\_T nexus on which the REQUEST SENSE command was received. The logical unit may discard any pending sense data and shall clear the reported unit attention condition for the SCSI initiator port associated with that I\_T nexus.

If the logical unit has already generated the ACA condition (see 5.8.2) for a unit attention condition, the logical unit shall report the unit attention condition (i.e., option b) above).

If a command other than INQUIRY, REPORT LUNS, or REQUEST SENSE enters the enabled task state while a unit attention condition exists for the SCSI initiator port associated with the I\_T nexus on which the command was received, the logical unit shall terminate the command with a CHECK CONDITION status. The logical unit shall provide sense data that reports a unit attention condition for the SCSI initiator port that sent the command on the I\_T nexus.

If a logical unit reports a unit attention condition with a CHECK CONDITION status and the UA\_INTLCK\_CTRL field in the Control mode page contains 00b (see SPC-3), then the logical unit shall clear the reported unit attention condition for the SCSI initiator port associated with that I\_T nexus on the logical unit. If the UA\_INTLCK\_CTRL field contains 10b or 11b, the logical unit shall not clear unit attention conditions reported with a CHECK CONDITION status.

## **6 SCSI events and event notification model**

### **6.1 SCSI events overview**

SCSI events may occur or be detected in either:

- a) The SCSI device;
- b) One or more SCSI ports within a SCSI device; or
- c) The application client, task manager, or device server.

The detection of any event may require processing by the object that detects it.

Events that occur in the SCSI device are assumed to be detected and processed by all objects within the SCSI device.

When a SCSI port detects an event, it shall use the event notification services (see 6.4) to notify SCSI application layer objects that the event has been detected.

The events detected and event notification services usage depends on whether the SCSI device is a SCSI target device (see figure 7) or a SCSI initiator device (see figure 8).

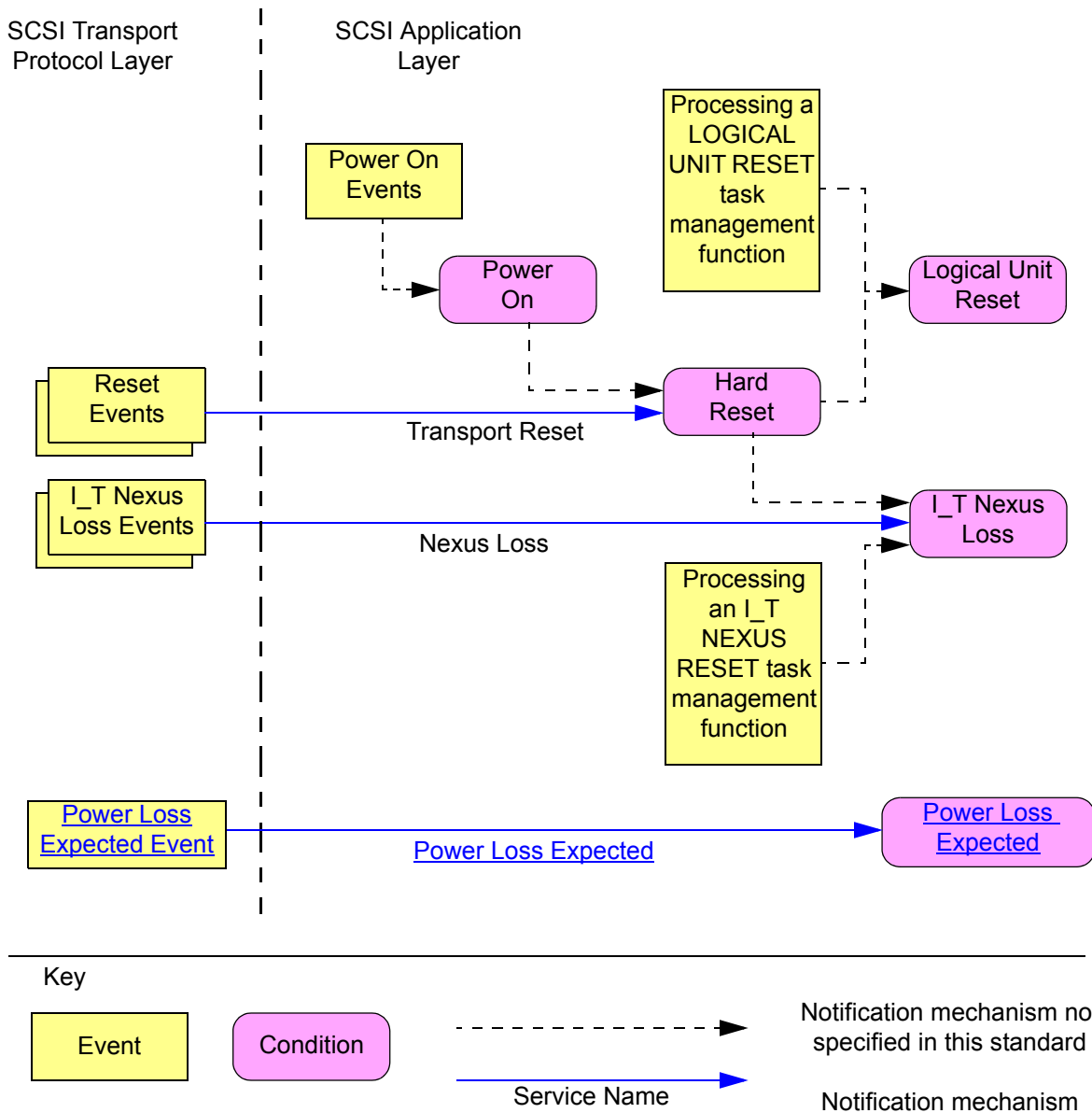


Figure 7 — Events and event notifications for SCSI target devices

---



---

Editor's Note 8: Figure 5 has changed. The changes are indicated by blue underlines.

---



---

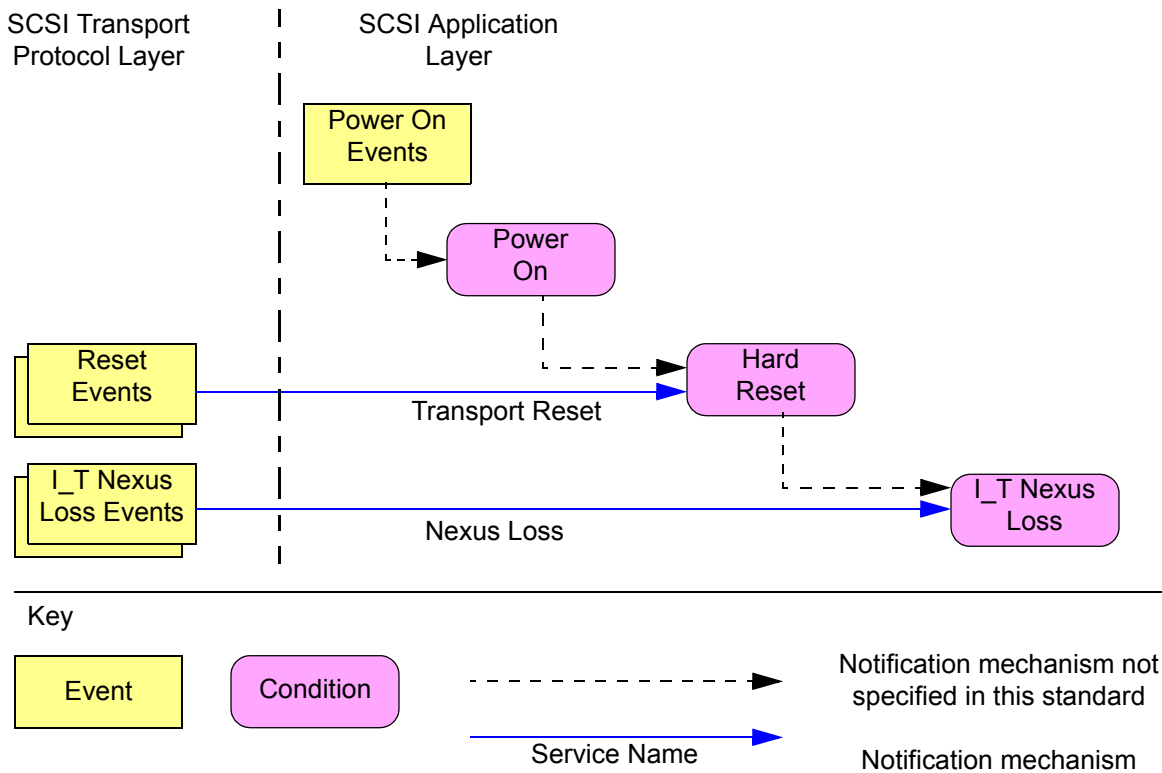


Figure 8 — Events and event notifications for SCSI initiator devices

## 6.2 Establishing a unit attention condition subsequent to detection of an event

Table 13 shows the additional sense code that a logical unit shall use when a unit attention (see 5.8.7) is established for each of the conditions shown in figure 7 (see 6.1). A SCSI transport protocol may define a more specific additional sense code than SCSI BUS RESET OCCURRED for reset events. The most specific condition in table 13 known to the logical unit should be used to establish the additional sense code for a unit attention.

Table 13 — Unit attention additional sense codes for events detected by SCSI target devices

Condition	Additional Sense Code	Specificity
Logical unit is unable to distinguish between the conditions	POWER ON, RESET, OR BUS DEVICE RESET OCCURRED	Lowest
<a href="#">Power Loss Expected</a>	<a href="#">COMMANDS CLEARED BY POWER LOSS NOTIFICATION</a>	
Power on	POWER ON OCCURRED or DEVICE INTERNAL RESET	
Hard reset	SCSI BUS RESET OCCURRED or protocol specific	
Logical unit reset	BUS DEVICE RESET FUNCTION OCCURRED	
I_T nexus loss	I_T NEXUS LOSS OCCURRED	Highest

NOTE 5 - The names of the unit attention conditions listed in the subclause (e.g., SCSI BUS RESET OCCURRED) are based on usage in previous versions of this standard. The use of these unit attention condition names is not to be interpreted as a description of how the unit attention conditions are represented by any given SCSI transport protocol.

A logical unit may use the I\_T NEXUS LOSS OCCURRED additional sense code when establishing a unit attention condition if:

- a) The SCSI initiator port to which the sense data is being delivered is the SCSI initiator port that was associated with the I\_T nexus loss, and the logical unit has maintained all state information specific to that SCSI initiator port since the I\_T nexus loss; and
- b) The I\_T nexus being used to deliver the sense data is the same I\_T nexus that was lost, and the logical unit has maintained all state information specific to that I\_T nexus since the I\_T nexus loss.

Otherwise, the logical unit shall use one of the less specific additional sense codes (e.g., POWER ON OCCURRED) when establishing a unit attention condition.

## 6.3 Conditions resulting from SCSI events

### 6.3.1 Power on

### 6.3.2 Hard reset

### 6.3.3 Logical unit reset

### 6.3.4 I\_T nexus loss

### 6.3.5 Power loss expected

Power loss expected is a SCSI device condition resulting from a power loss expected event indicated by a **Power Loss Expected** event notification (see 6.4).

A power loss expected event is an indication from the SCSI transport protocol to the SCSI application layer that power loss may occur within a protocol specific period of time. SCSI transport protocols may define power loss expected events.

Each SCSI transport protocol standard that defines power loss expected events should specify when those events result in the delivery of a **Power Loss Expected** event notification to the SCSI applications layer.

The power loss expected condition applies only to SCSI target devices.

When a SCSI target port detects a power loss expected, a **Power Loss Expected** event notification indication shall be delivered to each logical unit to which the I\_T nexus has access. In response to the resulting I\_T power loss expected condition a logical unit shall take the following actions:

- a) Abort all tasks as described in 5.6;
- b) Clear all ACA conditions (see 5.8.2.5) in all task sets in the logical unit;
- c) Establish a unit attention condition (see 5.8.7 and 6.2); and
- d) Perform any additional functions required by the applicable protocol standards.

## 6.4 Event notification SCSI transport protocol services

The SCSI transport protocol services described in this subclause are used by a SCSI initiator port or a SCSI target port to deliver an indication to the SCSI application layer that a SCSI event has been detected.

All SCSI transport protocol standards should define the SCSI transport protocol specific requirements for implementing the **Nexus Loss** indication, ~~and~~ the **Transport Reset** indication, and the **Power Loss**

[Expected indication](#) described in this subclause and when these indications are to be delivered to the SCSI applications layer.

The **Nexus Loss** [indication](#) and the **Transport Reset** indications are defined for both SCSI target devices and SCSI initiator devices.

Indication delivered to device servers, task managers, and application clients:

**Nexus Loss (IN ( I\_T Nexus ))**

Argument description:

**I\_T Nexus:** The specific I\_T nexus that has been detected as lost.

Indication delivered to device servers, task managers, and application clients:

**Transport Reset (IN ( SCSI Port ))**

Argument descriptions:

**SCSI Port:** The specific SCSI port in the SCSI device for which a transport reset was detected.

[The Power Loss Expected indication is defined for SCSI target devices.](#)

[Indication delivered to device servers and task managers.](#)

**[Power Loss Expected \(IN \( SCSI Port \)\)](#)**

Argument description:

**[SCSI Port:](#)** [The specific SCSI port in the SCSI device for which an expected power loss was detected.](#)