To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 29 December 2003
Subject: 03-186r3r4 SAS-1.1 Transport layer retries

Revision history

Revision 0 (6 May 2003) first revision
Revision 1 (2 July 2003) added more details
Revision 2 (28 July 2003) incorporated comments from the July SAS working group meeting.
Revision 3 (24 October 2003) incorporated comments from the September SAS working group meeting. Converted proposal into FrameMaker.
Revision 4 (29 December 2003) incorporated comments from before the November SAS working group meeting.

Related documents

sas1r01 - Serial Attached SCSI 1.1 revision 1
03-229 SAS-1.1 Transport layer retries ladders (Jim Jones, Quantum)

Overview

In SAS 1.0, errors transmitting frames (except for RESPONSE frames) result in either the logical unit terminating the command with CHECK CONDITION status, or the application client aborting the command. Possible errors are shown in figure 1.

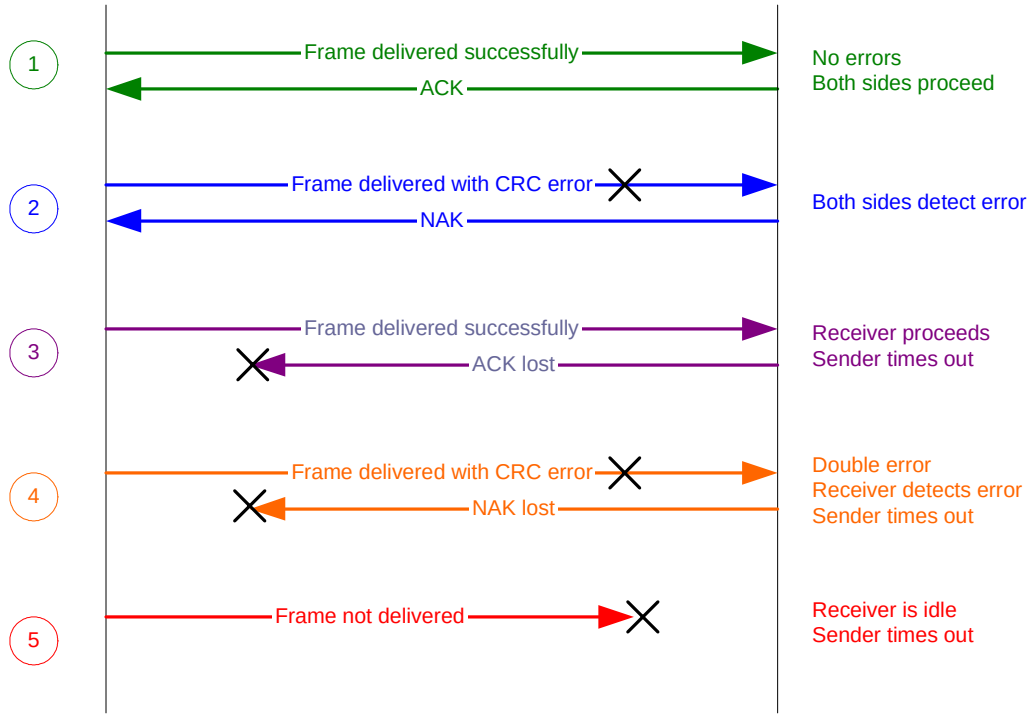

Figure 1 — Error cases in SAS

Depending on the type of frame, different things happen in SAS-1.0. Table 1 shows the SAS 1.0 behavior.

Table 1 — SAS 1.0 behavior

	COMMAND or TASK (I to T)	XFER_RDY (T to I)	RESPONSE (T to I)	read DATA (T to I)	write DATA (I to T)
1. Frame arrives OK; ACK arrives OK	Target runs the command after sending ACK.	Initiator replies with write DATA frame(s).	Both finish the command. Initiator can reuse tag after evidence of target progression.	Move on to more data or RESPONSE (or XFER_RDY for bidi commands)	Move on to more data, XFER_RDY, or RESPONSE (or read DATA for bidi commands)
2. Frame arrives w/ CRC error; NAK arr. OK	Target idle after sending NAK. Initiator sees the NAK and can resend the command.	Target terminates command.	Target can resend with retransmit=1.	Target terminates command	Initiator aborts command
3. Frame arrives OK; ACK lost	Target runs the command after sending ACK. I to T direction hung interlocked. Initiator detects ACK/NAK timeout in 1 ms. T to I direction could deliver DATA, XFER_RDY, or RESPONSE frames in that time. Initiator cannot tell between ACK lost, NAK lost, and frame lost w/o using QUERY TASK.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. I to T direction could deliver write DATA frames in that time. Target cannot tell between ACK lost, NAK lost, and frame lost.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. I to T direction not supposed to deliver new command with the same tag yet. Target resends with retransmit=1 in a new connection.	Subsequent ACKs/NAKs for data in flight are misassigned by the target. Target ACK/NAK timeout. Target terminates cmd.	Subsequent ACK/NAKs for data in flight are misassigned by the initiator. Initiator ACK/NAK timeout. Initiator aborts cmd.
4. Frame arrives w/ CRC error; NAK lost (double error)	Target idle after sending NAK. I to T direction hung interlocked. Initiator detects ACK/NAK timeout in 1 ms. Initiator can resend cmd later. Initiator cannot tell between ACK lost, NAK lost, and frame lost w/o using QUERY TASK.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. Target cannot tell between ACK lost, NAK lost, and frame lost.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. Target resends with retransmit=1 in new connection.	Subsequent ACK/NAKs for data in flight are misassigned by the target. Initiator sees data offset gap. Target ACK/NAK timeout. Target terminates command. Initiator aborts cmd because of gap (if subseq. DATA frames)	Subsequent ACK/NAKs for data in flight are misassigned by the initiator. Initiator ACK/NAK timeout. Initiator aborts cmd. Target terminates command because of gap (if subseq. DATA frames). Initiator Response Timeout may occur causing target to terminate cmd.
5. Frame lost	Target idle. I to T direction hung interlocked. Initiator detects ACK/NAK timeout in 1 ms. Initiator can resend cmd later. Initiator cannot tell between ACK lost, NAK lost, and frame lost w/o using QUERY TASK.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. Target terminates cmd. Target cannot tell between ACK lost, NAK lost, and frame lost.	T to I hung interlocked. Target detects ACK/NAK timeout in 1 ms. Target resends with retransmit=1 in new connection.	Subsequent ACK/NAKs for data in flight are misassigned by the target. Initiator sees data offset gap. Target ACK/NAK timeout. Target terminates command. Initiator aborts cmd because of gap.	Subsequent ACK/NAKs for data in flight are misassigned by the initiator. Initiator ACK/NAK timeout. Initiator aborts cmd. Target terminates command because of gap (if subseq. DATA frames). Initiator Response Timeout may occur causing target to terminate cmd.

Proposed enhancements

Some backup applications will fail the whole backup if any of their commands are terminated with CHECK CONDITION. A way to retransmit frames is desirable.

These special recovery features are not always wanted; per-logical unit controls are needed. Tape drives might implement these features; disk drives probably would not. A mode page per logical unit to enable/disable the special recovery mode, different than the Enable Modify Data Pointers bit in the Disconnect-Reconnect mode page, is needed.

RESPONSE problems

a) If target gets a NAK, resend as in SAS-1.0 (with RETRANSMIT=1). RETRANSMIT=1 isn't strictly necessary since the target didn't see anything; but it helps logic analyzers.
b) If target detects an ACK/NAK timeout, resend as in SAS-1.0 (with RETRANSMIT=1)
c) Target shall retry RESPONSE problems at least one time.

COMMAND or TASK problems

a) If initiator gets a NAK, resend as in SAS-1.0 if desired
b) If initiator detects a RESPONSE frame before it gets an ACK for a command, an ACK was lost. Treat the command as successful. Still close the connection with ACK/NAK Timeout, but do nothing more.
c) If initiator detects an XFER_RDY or read DATA frame (for a COMMAND frame, not for a TASK frame) before it gets an ACK for a command, an ACK was lost. Treat the command as successfully received. Still close the connection with ACK/NAK Timeout, but start servicing the XFER_RDY or read DATA.
d) If initiator detects an ACK/NAK timeout, close connection with ACK/NAK Timeout and send QUERY TASK in a new connection to see if the command is running in the target:
    A) If FUNCTION SUCCEEDED, then assume it's running fine. If first burst is enabled, start sending write DATA frames.
    B) If FUNCTION COMPLETE, then resend the command.
    C) Continue watching for a RESPONSE frame while running QUERY TASK.
    D) If a RESPONSE shows up, don't try to abort the command and don't report it as aborted to the application client.
    E) if QUERY TASK is not supported, initiator will have to use ABORT TASK and try to abort the command (it may get a RESPONSE anyway). Logical units behind target ports supporting transport layer retries shall support QUERY TASK.

XFER_RDY problems

a) If target gets a NAK, resend with RETRANSMIT=1 and a new TPTT. RETRANSMIT=1 isn't strictly necessary since the initiator didn't see anything, but it helps logic analyzers. New TPTT isn't strictly necessary since the old one isn't being used - however, this seems simpler
b) If target detects an ACK/NAK timeout, resend with RETRANSMIT=1 and a new TPTT
c) Target shall retry XFER_RDY problems at least one time.
d) If target sees write DATA frames before seeing an ACK for the XFER_RDY, target discards them
e) Write DATA frames in response to an XFER_RDY with RETRANSMIT=1 do not themselves have RETRANSMIT=1
f) Target shall change Target Port Transfer Tag between XFER_RDYs (either retransmitted or normal)
g) Initiator can continue sending data for the previous XFER_RDY when a new one arrives for the same command (same tag), but must not send data for the new XFER_RDY until it is done sending for the previous. Once it sends a data frame for the new XFER_RDY, no more data frames for the old are allowed. This applies to both RETRANSMIT=1 or RETRANSMIT=0 cases.
h) The target can reuse the first TPTT when it sees a data frame with the second TPTT.
i) If initiator receives an XFER_RDY with RETRANSMIT=0 but it is already servicing an XFER_RDY, it should assume an error and abort the command. This is more likely a bug than a single bit error worth recovering from.

Write DATA problems

a) Set RETRY DATA FRAMES=1 bit in XFER_RDY to tell the initiator to retry write DATA frames that encounter errors. Only target ports supporting transport layer retries would set this bit (and only if their mode page is thus enabled).
b) If initiator gets a NAK, resend that write DATA frame and all that followed. It shall go back to the last XFER_RDY base offset. (An ACK/NAK balance point might also work, but XFER_RDY will be easier for targets to handle. Going back to the frame in error rather than the ACK/NAK balance point is risky because a lost ACK or NAK could point to the wrong frame.)
c) If initiator detects an ACK/NAK timeout, resend all write DATA frames since the last XFER_RDY base offset. (Last ACK/NAK balance point could work but not allowed)
d) Each resent write DATA frame has the correct Data Offset
e) No use of the RETRANSMIT bit for write DATA frames
f) First resent write DATA frame has CHANGING DATA POINTER=1
g) If an ACK (rather than a NAK) was lost, the target doesn't know and may follow with a RESPONSE or XFER_RDY that surprises the initiator (in a subsequent connection)
    A) If resent write DATA frames cross a RESPONSE frame, initiator has to accept the RESPONSE and complete the command without worrying about the missing ACK for its earlier write DATA frame. The target discards subsequent write data frames since they have a now-unknown tag and TPTT. Easiest if initiator has to stop sending frames for the command after receiving a RESPONSE.
    B) If resent write DATA frames cross an XFER_RDY, initiator has to accept the XFER_RDY. Target shall use a different Target Port Transfer Tag for the XFER_RDY to help differentiate them. Proposal: initiator has to stop sending write DATA frames for the old XFER_RDY before sending any write DATA frames for the new XFER_RDY..
h) Receiving a RESPONSE frame or XFER_RDY does NOT serve as a link layer ACK for the outbound direction - an ACK/NAK timeout must still occur. The inbound frame must be ACKed and honored, though (the T to I direction does not timeout).
i) Target shall not abort a command when it sees a relative offset error. Target shall start discarding data after the gap until it gets a frame with CHANGING DATA POINTER=1 (unlike the initiator on read data gaps, target does not have the option of storing bad data in the wrong place). Target may discard resent data up to the gap if it chooses (rather than overwriting).

Read DATA problems

a) If target gets a NAK, it shall resend that read DATA frame and all that followed. It shall go back to a ACK/NAK balance point (probably but not necessarily the last one) rather than retry starting from the read DATA frame it thinks was NAKed, because a lost ACK or NAK while pipelining could mean it mismatched the NAK.
b) If target detects an ACK/NAK timeout, it shall resend all read DATA frames since the last ACK/NAK balance point.
c) If target gets a NAK, it should stop sending read DATA frames.
d) No use of RETRANSMIT bit for read DATA frames
e) Set the CHANGING DATA POINTER bit to 1 when the target is going back to an earlier data offset
f) Target frames shall be monotonically increasing after CHANGING DATA POINTER=1
g) Initiator shall not abort a command when it sees a relative offset error. Initiator may start discarding data after the gap until it gets a frame with CHANGING DATA POINTER=1 (or it may store the data at the wrong offsets, or it may store the data at the correct offsets). Initiator may discard resent data up to the gap if it chooses (rather than overwriting).
h) Target shall retry Read DATA problems at least one time.

New bits for target ports supporting transport layer retries

Keep the RETRANSMIT bit in the SSP frame header (byte 10 bit 1).

a) SAS-1.0 only allowed this bit be used for RESPONSE frames.
b) SAS-1.1 also allow it for XFER_RDY frames

New CHANGING DATA POINTER bit for DATA frames (both read DATA and write DATA frames) (byte 10 bit 0)

a) 1 = the data offset is being set to a non-monotonically increasing value. (it must always go backwards)

New RETRY DATA FRAMES bit for XFER_RDY frames only (byte 9 bit 2)

a) The initiator has permission to retry write DATA frames that encounter NAKs or ACK/NAK timeouts

Add the Protocol-Specific Logical Unit mode page (18h) (new to SAS):

a) TRANSPORT LAYER RETRIES bit (located anywhere)
b) 1 = the device server in the logical unit instructs the target port to:
   A) set XFER_RDY frame header RETRY DATA FRAMES bit to 1
   B) retry read DATA frames that fail
   C) accept write DATA frames with CHANGING DATA POINTER = 1

Transition

If the target sends an XFER_RDY with RETRY DATA FRAMES set to 1 to an initiator that doesn't understand it:

a) If the initiator checks reserved fields, it will abort the command. If it determines it is aborting every one of its write commands, it might be prudent to ignore the reserved field. If the initiator is SAS-1.1 cognizant but does not support the feature, SAS-1.1 will require it to ignore the field and investigate why the mode page is set wrong.
b) If the initiator does not check reserved fields, it will ignore it and never try to retry. If it encounters an ACK/NAK timeout or a NAK, it will abort the command.

If the initiator sends CHANGING DATA POINTER=1 to a SAS-1 target:

a) If the target checks reserved fields, it will abort the command. This should only happen after an error, so that's how SAS-1 behaved anyway.
b) If the target does not check reserved fields, it will ignore it, detect a relative offset error, and abort the command. This should only happen after an error, so that's how SAS-1 behaved anyway.

If the target sends CHANGING DATA POINTER=1 to a SAS-1 initiator:

a) If the initiator checks reserved fields, it will abort the command. This should only happen after an error, so that's how SAS-1 behaved anyway.
b) If the initiator does not check reserved fields, it will ignore it, detect a relative offset error, and abort the command. This should only happen after an error, so that's how SAS-1 behaved anyway.

Suggested changes

This proposal contains these changes:

a) changes to the "SSP transport layer handling of link layer errors" section
b) changes to the ST state machines
c) changes to the application client error handling rules
d) the new Protocol-Specific Logical Unit mode page definition

9.2 SSP transport layer

9.2.1 SSP frame format

Table 2 defines the SSP frame format.

Table 2 — SSP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	FRAME TYPE							
1	(MSB)	HASHED DESTINATION SAS ADDRESS						
3								(LSB)
4	Reserved							
5	(MSB)	HASHED SOURCE SAS ADDRESS						
7								(LSB)
8	Reserved							
9	Reserved							
10	Reserved					RETRY DATA FRAMES	RETRANSMIT	CHANGING DATA POINTER
11	Reserved						NUMBER OF FILL BYTES	
12	Reserved							
13	Reserved							
15								
16	(MSB)	TAG						
17								(LSB)
18	(MSB)	TARGET PORT TRANSFER TAG						
19								(LSB)
20	(MSB)	DATA OFFSET						
23								(LSB)
24		INFORMATION UNIT						
m								
	Fill bytes, if needed							
n - 3	(MSB)	CRC						
n								(LSB)

...

The RETRY DATA FRAMES bit is set to one for XFER_RDY frames under the conditions defined in 9.2.4 and shall be set to zero for all other frame types. When set to one this bit indicates the SSP initiator port may retry write DATA frames that fail.

The RETRANSMIT bit is set to one for RESPONSE frames and XFER_RDY frames under ~~certain conditions (see 9.2.4.5)~~ the conditions defined in 9.2.4 and shall be set to zero for all other frame types. When set to one this bit indicates the frame is a retransmission after the SSP target port failed in its previous attempt to transmit the frame.

The CHANGING DATA POINTER bit is set to one for DATA frames under the conditions defined in 9.2.4 and shall be set to zero for all other frame types. When set to one this bit indicates the frame is a retransmission after the SSP target port failed in its previous attempt to transmit the frame or a subsequent frame and the data offset field of the frame may not be sequentially increased from that of the previous frame.

...

9.2.4 SSP transport layer handling of link layer errors [reorganized and rewritten]

9.2.4.1 SSP transport layer handling of link layer errors overview

The transport layer, sometimes assisted by the application layer, handles some link layer errors (e.g., NAKs and ACK/NAK timeouts).

Link layer errors that are encountered when transmitting XFER_RDY and DATA frames are handled differently based on the TRANSPORT LAYER RETRIES bit in the Protocol Specific Logical Unit mode page (see 10.2.6.3) of the logical unit that is the source of the frame.

If the TRANSPORT LAYER RETRIES bit is set to zero, a logical unit:

a) does not support transport layer retries;
b) sets the RETRY DATA FRAMES bit to zero in each XFER_RDY frame;
c) may or may not select a different value for the TARGET PORT TRANSFER TAG field in each XFER_RDY frame than that used in the previous XFER_RDY frame for that I_T_L_Q nexus;
d) processes XFER_RDY frame link layer errors as described in 9.2.4.4.3; and
e) processes DATA frame link layer errors as described in 9.2.4.5.3.

If the TRANSPORT LAYER RETRIES bit is set to one, the logical unit:

a) supports transport layer retries;
b) supports the QUERY TASK task management function (see SAM-3);
c) sets the RETRY DATA FRAMES bit to one in each XFER_RDY frame;
d) selects a different value for the TARGET PORT TRANSFER TAG field in each XFER_RDY frame than that used in the previous XFER_RDY frame for that I_T_L_Q nexus;
e) processes XFER_RDY frame link layer errors as described in 9.2.4.4.2; and
f) processes DATA frame link layer errors as described in 9.2.4.5.2.

9.2.4.2 COMMAND frame - handling of link layer errors

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken) it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) transmits, in a new connection, a QUERY TASK task management function to determine whether the command was received (see 10.2.2). The TAG OF TASK TO BE MANAGED field is set to the tag of the COMMAND frame.

If the SSP initiator port receives an XFER_RDY frame for the I_T_L_Q nexus of the command before the RESPONSE frame for the QUERY TASK, it considers the XFER_RDY frame valid and assumes that the COMMAND frame was received and that the command is being processed.

If the SSP initiator port receives a RESPONSE frame for the I_T_L_Q nexus of the command before the RESPONSE frame for the QUERY TASK, it considers the RESPONSE frame valid and assumes that the COMMAND frame was received and that the command is complete. The SSP initiator port may reuse the tag of the COMMAND frame.

If the SSP initiator port receives a RESPONSE frame for the QUERY TASK with a response code of TASK MANAGEMENT FUNCTION SUCCEEDED, it assumes that the COMMAND frame was received (i.e., ACKed) and the command is being processed.

If the SSP initiator port receives a RESPONSE frame for the QUERY TASK with a response code of TASK MANAGEMENT FUNCTION COMPLETE, it assumes that the COMMAND frame is not being processed. If neither an XFER_RDY frame nor a RESPONSE frame has been received for the I_T_L_Q nexus of the command, the SSP initiator port assumes the command was not received (i.e., NAKed or lost). The SSP initiator port may reuse the tag of the COMMAND frame.

An SSP initiator port should retransmit each COMMAND frame that does not receive an ACK at least one time.

9.2.4.3 TASK frame - handling of link layer errors

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken) it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) retransmits, in a new connection, the TASK frame using the same tag (see 10.2.2).

Editor's Note 1: could use the retransmit bit to tell the target not to be upset by a duplicate request. Otherwise it might complain about overlapped tags and return a response code INVALID FRAME (per 9.2.5.1), still leaving the initiator in limbo.

If the SSP initiator port receives a RESPONSE frame for the TASK frame that arrives before the ACK or NAK for the TASK frame, it considers the RESPONSE frame valid and assume that the TASK frame was received (i.e., ACKed) and that the task management function is complete. The initiator port may reuse the tag of the TASK frame.

An SSP initiator port should retransmit each TASK frame that does not receive an ACK at least one time.

9.2.4.4 XFER_RDY frame - handling of link layer errors

9.2.4.4.1 XFER_RDY frame overview

If an SSP target port has the TRANSPORT LAYER RETRIES bit set to one in the Protocol-Specific Logical Unit mode page (see 10.x.x.x), then it processes XFER_RDY frame link layer errors as described in 9.2.4.4.2.

If an SSP target port has the TRANSPORT LAYER RETRIES bit set to zero, then it processes XFER_RDY frame link layer errors as described in 9.2.4.4.3.

9.2.4.4.2 XFER_RDY frame with transport layer retries

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) retransmits, in a new connection, the XFER_RDY frame with a different value in the TARGET PORT TRANSFER TAG field and with the RETRANSMIT bit set to one (see 9.2.6.3.3.6).

If an SSP target port transmits an XFER_RDY frame and receives a NAK for that frame, it retransmits the XFER_RDY frame with a different value in the TARGET PORT TRANSFER TAG field and with the RETRANSMIT bit set to one (see 9.2.6.3.3.6).

If an SSP initiator port receives a new XFER_RDY frame with the RETRANSMIT bit set to one while processing the previous XFER_RDY frame for that I_T_L_Q nexus, it should stop processing the previous XFER_RDY frame (i.e., stop sending write DATA frames) and start servicing the new XFER_RDY frame. The SSP initiator port does not send any write DATA frames for the previous XFER_RDY after sending a write DATA frame for the new XFER_RDY.

The SSP target port may reuse the value in the TARGET PORT TRANSFER TAG field from the previous XFER_RDY frame when it receives a write DATA frame for the new XFER_RDY frame.

An SSP target port retransmits each XFER_RDY frame that does not receive an ACK at least one time.

9.2.4.4.3 XFER_RDY frame without transport layer retries

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) transmits, in a new connection, a RESPONSE frame returning a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.2.3).

If an SSP target port transmits an XFER_RDY frame and receives a NAK for that frame, it returns a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.2.3).

9.2.4.5 DATA frame - handling of link layer errors

9.2.4.5.1 DATA frame overview

If an SSP target port sends a read DATA frame for a logical unit that has its TRANSPORT LAYER RETRIES bit set to one in the Protocol-Specific Logical Unit mode page (see 10.x.x.x), then the SSP target port processes read DATA frame link layer errors as described in 9.2.4.5.2. If the logical unit has its TRANSPORT LAYER RETRIES bit set to zero, then the SSP target port processes read DATA frame link layer errors as described in 9.2.4.5.3.

An SSP initiator port processes write DATA frames sent in response to an XFER_RDY frame that has its RETRY DATA FRAMES bit set to one as described in 9.2.4.5.2. An SSP initiator port processes write DATA frames sent in response to an XFER_RDY frame that has its RETRY DATA FRAMES bit set to zero as described in 9.2.4.5.3.

9.2.4.5.2 DATA frame with transport layer retries

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) retransmits, in a new connection, all the read DATA frames since a previous time when ACK/NAK balance occurred (see 7.16.7.3).

If an SSP target port transmits a read DATA frame and receives a NAK for that frame, it retransmits, in the same or in a new connection, all the read DATA frames since a previous time when ACK/NAK balance occurred.

If an SSP initiator port transmits a write DATA frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) retransmits, in a new connection, all the write DATA frames for the previous XFER_RDY.

If it receives a new XFER_RDY or a RESPONSE frame for the command, it processes the XFER_RDY or RESPONSE frame and should stop sending the retransmitted write DATA frames. It does not send a write DATA frame for the previous XFER_RDY after sending a write DATA frame for the new XFER_RDY.

If an SSP initiator port transmits a write DATA frame and receives a NAK for that frame, it retransmits, in the same or in a new connection, all the write DATA frames for the previous XFER_RDY.

For both reads and writes, the first retransmitted DATA frame has its CHANGING DATA POINTER bit set to one and subsequent DATA frames have their CHANGING DATA POINTER bits set to zero.

An SSP port retransmits each DATA frame that does not receive an ACK at least one time. The number of times it retransmits each DATA frame is vendor-specific.

9.2.4.5.3 DATA frame without transport layer retries

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) transmits, in a new connection, a RESPONSE frame returning a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT (see 10.2.3).

If an SSP target port transmits a read DATA frame and receives a NAK for that frame, it returns a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED (see 10.2.3).

If an SSP initiator port transmits a write DATA frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it closes the connection with DONE (ACK/NAK TIMEOUT) and aborts the command (see 10.2.2).

If an SSP initiator port transmits a write DATA frame and receives a NAK for that frame, it aborts the command (see 10.2.2).

9.2.4.6 RESPONSE frame - handling of link layer errors

If an SSP target port transmits a RESPONSE frame and does not receive an ACK or NAK for that frame (e.g., times out, or the connection is broken), it:

1) closes the connection with DONE (ACK/NAK TIMEOUT); and
2) retransmits, in a new connection, the RESPONSE frame with the RETRANSMIT bit set to one.

If an SSP target port transmits a RESPONSE frame and receives a NAK for that frame, it retransmits the RESPONSE frame at least one time with the RETRANSMIT bit set to zero (see 9.2.6.3.3.8).

An SSP target port retransmits each RESPONSE frame that does not receive an ACK at least one time. The number of times it retransmits each RESPONSE frame is vendor-specific.

If an SSP initiator port receives a RESPONSE frame with a RETRANSMIT bit set to one, and it has previously received a RESPONSE frame for the same I_T_L_Q nexus, it discards the extra RESPONSE frame. If it has not previously received the RESPONSE frame, it considers it to be the valid RESPONSE frame (see 10.2.2).

9.2.6 ST (transport layer for SSP ports) state machines [modifications throughout]

9.2.6.1 ST state machines overview

The ST state machines perform the following functions:

a) receive and process transport protocol service requests and transport protocol service responses from the SCSI application layer;
b) receive and process other SAS connection management requests from the application layer;
c) send transport protocol service indications and transport protocol service confirmations to the SCSI application layer;
d) send requests to the port layer to transmit frames and manage SAS connections; and
e) receive confirmations from the port layer.

The following confirmations between the ST state machines and the port layer:

a) Transmission Status; and
b) Frame Received;

include the following as arguments:

a) the tag;
b) the destination SAS address; and
c) the source SAS address;

and are used to route the confirmations to the correct ST state machines.

9.2.6.2 ST_I (transport layer for SSP initiator ports) state machines

9.2.6.2.1 ST_I state machines overview

The ST_I state machines are as follows:

a) ST_ISF (initiator send frame) state machine (see 9.2.6.2.2);
b) ST_IPD (initiator process data) state machine (see 9.2.6.2.3);
c) ST_IPR (initiator process response) state machine (see 9.2.6.2.4); and
d) ST_IFR (initiator frame router) state machine (see 9.2.6.2.5).

Figure 2 shows the ST_I state machines.

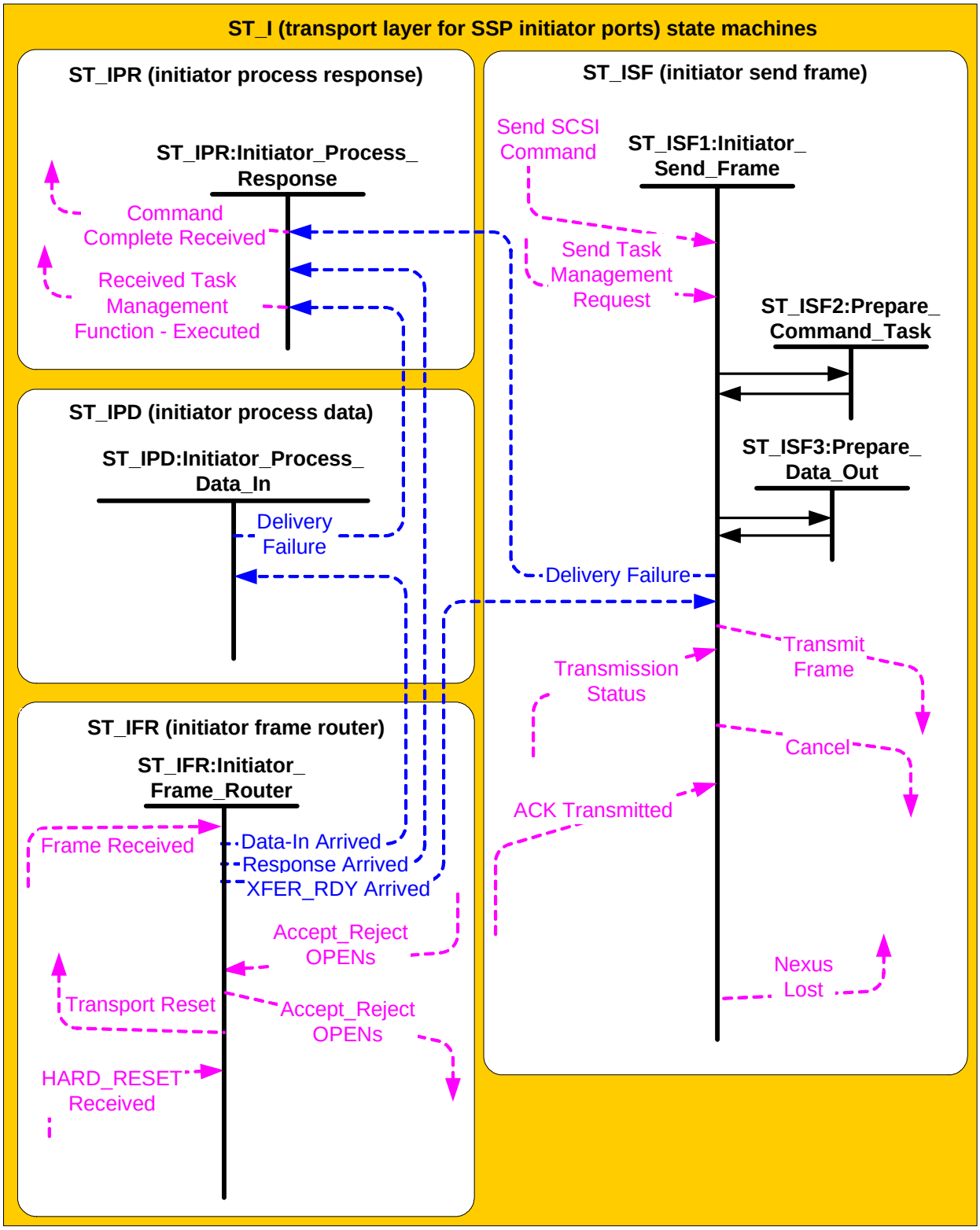

Figure 2 — ST_I (transport layer for SSP initiator ports) state machines

9.2.6.2.2 ST_ISF (initiator send frame) state machine

9.2.6.2.2.1 ST_ISF state machine overview

The ST_ISF state machine receives transport protocol service requests from the SCSI application layer, receives XFER_RDY Arrived messages from the ST_IFR state machine, and constructs COMMAND, TASK, or data-out DATA frames. The service request may be to process either a command or task management function. This state machine also communicates with the port layer via requests and confirmations regarding frame transmission, and may communicate to the ST_IPR state machine regarding service delivery subsystem failures.

This state machine consists of the following states:

a) ST_ISF1:Send_Frame (see 9.2.6.2.2.2)(initial state);
b) ST_ISF2:Prepare_Command_Task (see 9.2.6.2.2.3); and
c) ST_ISF3:Prepare_Data_Out (see 9.2.6.2.2.4).

This state machine shall be started when a Send SCSI Command or a Send Task Management Request transport protocol service request is received from the SCSI application layer or when an XFER_RDY Arrived message is received.

9.2.6.2.2.2 ST_ISF1:Send_Frame state

9.2.6.2.2.2.1 State description

A Send SCSI Command transport protocol service request or a Send Task Management Request transport protocol service request includes the following to be used in any OPEN address frames required to service the request:

a) connection rate;
b) initiator connection tag; and
c) destination SAS address.

A Send SCSI Command transport protocol service request also includes the following to be used in any SSP frame for the request:

a) logical unit number;
b) tag;
c) task attribute;
d) additional CDB length;
e) CDB; and
f) additional CDB bytes.

If the request is for a data-out command and first burst is enabled (see 9.2.2.1), then the request also includes the number of bytes for the first burst size for the logical unit.

A Send Task Management Request transport protocol service request includes the following to be used in the TASK frame:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

If this state machine was started as the result of receiving an XFER_RDY Arrived message, then:

a) If an XFER_RDY frame is not expected for the command (e.g., for a read command), then this state shall discard the frame and shall send a Delivery Failure (XFER_RDY Not Expected) message to the ST_IPR state machine. This state machine shall terminate if it sends the message;
b) If the length of the information unit is not 12 bytes, then this state shall discard the frame and shall send a Delivery Failure (XFER_RDY Information Unit Too Short) message or Delivery Failure (XFER_RDY Information Unit Too Long) to the ST_IPR state machine. This state machine shall terminate after sending the message;

c) If the length of the XFER_RDY information unit is 12 bytes and the write data length is zero or exceeds the amount of data remaining to be transferred for the data-out command, then this state shall send a Delivery Failure (XFER_RDY Incorrect Write Data Length) message to the ST_IPR state machine. This state machine shall terminate after sending the message; or
d) If the length of the XFER_RDY information unit is 12 bytes and the requested offset is not expected, then this state shall send a Delivery Failure (XFER_RDY Requested Offset Error) message to the ST_IPR state machine. This state machine shall terminate after sending the message.

If this state is entered from the ST_ISF2:Prepare_Command_Task state, then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_ISF3:Prepare_Data_Out state, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer.

A Transmit Frame request shall include the SSP frame and the following to be used for any OPEN address frame:

a) the initiator port bit set to one;
b) protocol set to SSP;
c) connection rate;
d) initiator connection tag;
e) destination SAS address; and
f) source SAS address set to the SAS address of the SSP initiator port.

After sending a Transmit Frame request to the port layer this state shall wait for a Transmission Status confirmation. If the confirmation is not Transmission Status (Frame Transmitted) and either:

a) the frame is not a DATA frame;
b) the frame is a DATA frame and the XFER_RDY had its RETRY DATA FRAMES bit set to zero; or
c) the frame is a DATA frame, the XFER_RDY had its RETRY DATA FRAMES bit set to one, and the state machine has exceeded its vendor-specific number of retries for the DATA frame,

then this state shall send a Delivery Failure (Service Delivery Subsystem Failure) message to the ST_IPR state machine. The Delivery Failure message shall include:

a) any argument received with the Transmission Status confirmation; and
b) I_T_L_x nexus information (i.e., destination SAS address and tag);

If the transmitted frame was a DATA frame, then this state shall transition to the ST_ISF3:Prepare_Data_Out state after receiving a Transmission Status (Frame Transmitted) confirmation if there is more data to transfer.

After receiving a Transmission Status (Frame Transmitted) confirmation, the state machine shall expect one of the following confirmations for the frame:

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 transmitted frame was a COMMAND frame or TASK frame requiring a data-out operation, then the state machine shall wait to receive a Transmission Status (ACK Received), Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) confirmation before transitioning from this state. If the transmitted frame was a DATA frame, the state machine may transition to ST_ISF3:Prepare_Data_Out as described in 9.2.6.2.2.2.4.

If a Transmission Status (NAK Received), Transmission Status (ACK/NAK Timeout), or Transmission Status (Connection Lost Without ACK/NAK) is received and either:

a) the frame is not a DATA frame;
b) the frame is a DATA frame and the XFER_RDY had its RETRY DATA FRAMES bit set to zero; or
c) the frame is a DATA frame and the XFER_RDY had its RETRY DATA FRAMES bit set to one, but the state machine has exceeded its vendor-specific number of retries for the DATA frame,

then this state shall:

a) send a Delivery Failure (Service Delivery Subsystem Failure - NAK Received) message to the ST_IPR state machine if Transmission Status (NAK Received) was received; or
b) send a Delivery Failure (Service Delivery Subsystem Failure - Connection Failed) message to the ST_IPR state machine if Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK) was received.

After sending a Delivery Failure message to the ST_IPR state machine, this state machine shall terminate.

This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request. A Cancel request shall include the following arguments:

a) the destination SAS address; and
b) the tag.

This state machine shall terminate upon receipt of a Transmission Status (Cancel Acknowledge) confirmation.

9.2.6.2.2.3 Transition ST_ISF1:Send_Frame to ST_ISF2:Prepare_Command_Task

This transition shall occur after receiving a Send SCSI Command or Send Task Management Request transport protocol service request.

9.2.6.2.2.4 Transition ST_ISF1:Send_Frame to ST_ISF3:Prepare_Data_Out

This transition shall occur after:

a) receiving a Transmission Status (ACK Received) confirmation for a COMMAND frame for a data-out operation if first burst is enabled;
b) receiving an XFER_RDY Arrived message followed by an ACK Transmitted confirmation, if the XFER_RDY frame is valid (see 9.2.6.2.2.1); or
c) receiving a Transmission Status (Frame Transmitted) confirmation for a Transmit Frame (Non-Inter-locked) request if the number of data bytes that has been transmitted for the request is less than the first burst size or the write data length specified in the XFER_RDY.

This transition shall occur and include a Retry argument if:

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

   and

b) the XFER_RDY frame contained a RETRY DATA FRAMES bit set to one; and
c) the vendor-specific number of retries for the DATA frame has not been exceeded.

Editor's Note 2: this reason (with Retry) is new

9.2.6.2.3 ST_ISF2:Prepare_Command_Task state

9.2.6.2.3.1 State description

This state shall construct either a COMMAND or TASK frame.

If the frame to be constructed is a COMMAND frame, then this state shall include the following values received from the SCSI application layer in the frame:

a) logical unit number;
b) tag;
c) task attribute;
d) additional CDB length;

e) CDB; and
f) additional CDB bytes.

If the frame to be constructed is a TASK frame, then this state shall include the following values received from the SCSI application layer in the frame:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

This state shall generate and include the following values in either a COMMAND or TASK frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero; and
e) number of fill bytes.

9.2.6.2.2.3.2 Transition ST_ISF2:Prepare_Command_Task to ST_ISF1:Send_Frame

This transition shall occur after constructing a COMMAND or TASK frame.

9.2.6.2.2.4 ST_ISF3:Prepare_Data_Out state

9.2.6.2.2.4.1 State description

This state shall construct a DATA frame. This state shall include the following values in the frame, received either from the SCSI application layer or included in an XFER_RDY Arrived message:

a) logical unit number;
b) tag;
c) target port transfer tag;
d) data offset; and
e) data.

This state shall generate and include the following values in the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes; and
f) fill bytes.

If this state is entered from the ST_ISF1:Send_Frame state with a Retry argument, the CHANGING DATA POINTER bit shall be set to one and data offset shall be set to a data offset associated with a previous ACK/NAK balance. Otherwise, the CHANGING DATA POINTER bit shall be set to zero and the data offset shall be set to the data offset of the previous DATA frame plus the size of the previous DATA frame.

9.2.6.2.2.4.2 Transition ST_ISF3:Prepare_Data_Out to ST_ISF1:Send_Frame

This transition shall occur after constructing a DATA frame.

9.2.6.2.3 ST_IPD (initiator process data) state machine

The ST_IPD state machine receives and processes a message from the ST_IFR state machine containing a DATA frame.

This state machine consists of one state.

This state machine shall be started when a Data-In Arrived message is received.

This state shall check the length and data offset of the DATA information unit.

If the length of the information unit is zero, then this state shall send a Delivery Failure (DATA Incorrect Read Data Length) message to the ST_IPR state machine.

If the length of the information unit exceeds the amount of data remaining to be transferred for the data-in command, then this state shall send a Delivery Failure (DATA Too Much Read Data) message to the ST_IPR state machine.

If the data offset is not an expected offset (i.e., the changing data pointer bit is set to one and the data offset is not set to a data offset associated with a previous ACK/NAK balance, or the changing data pointer bit is set to zero and the data offset is not set to the previous data offset plus the previous data size), then this state shall send a Delivery Failure (DATA Offset Error) message to the ST_IPR state machine.

If the DATA information unit is valid, this state shall process the data-in data.

This state machine shall terminate after sending a message or processing the data-in data.

9.2.6.2.4 ST_IPR (initiator process response) state machine

The ST_IPR state machine receives a message from the ST_IFR state machine containing a RESPONSE frame or a message containing a service delivery subsystem failure from the ST_ISF state machine. This state machine processes the RESPONSE frame or the service delivery subsystem failure and sends a transport protocol service confirmation to the SCSI application layer.

This state machine consists of one state.

This state machine shall be started when a Response Arrived message is received or a Delivery Failure message is received.

If a Delivery Failure message is received, this state shall send a Command Complete Received or Received Task Management Function – Executed confirmation to the SCSI application layer with the Service Response argument set as indicated by table 3.

Table 3 — Delivery Failure to Command Complete Received mapping

Delivery Failure argument	Command Complete Received (Service Response)
XFER_RDY Information Unit Too Short	Service Delivery or Target Failure - XFER_RDY Information Unit Too Short
XFER_RDY Information Unit Too Long	Service Delivery or Target Failure - XFER_RDY Information Unit Too Long
XFER_RDY Incorrect Write Data Length	Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length
XFER_RDY Requested Offset Error	Service Delivery or Target Failure - XFER_RDY Requested Offset Error
XFER_RDY Not Expected	Service Delivery or Target Failure - XFER_RDY Not Expected
DATA Incorrect Data Length	Service Delivery or Target Failure - DATA Incorrect Data Length
DATA Too Much Read Data	Service Delivery or Target Failure - DATA Too Much Read Data
DATA Offset Error	Service Delivery or Target Failure - DATA Offset Error
Service Delivery Subsystem Failure - NAK Received	Service Delivery or Target Failure - NAK Received
Service Delivery Subsystem Failure - Connection Failed	Service Delivery or Target Failure - Connection Failed

If a Response Arrived message is received, this state shall check the length of the RESPONSE information unit.

If the length of the information unit is not correct, then this state shall send a Command Complete Received (Service Response = Service Delivery or Target Failure) or Received Task Management Function – Executed (Service Response = Service Delivery or Target Failure) confirmation to the SCSI application layer. The confirmation shall include the tag.

If the length is correct, this state shall send a Command Complete Received (Service Response = Task Complete), Command Complete Received (Service Response = Linked Command Complete), or Received Task Management Function – Executed (Service Response = Function Complete) transport protocol service confirmation. The confirmation shall also include a Retransmit argument indicating the state of the RETRANSMIT bit.

This state machine shall terminate after sending a confirmation.

9.2.6.2.5 ST_IFR (initiator frame router) state machine

The ST_IFR state machine receives confirmations from the port layer and, depending on the confirmation, may send a message to the ST_ISF, ST_IPD, or ST_IPR state machines. This state machine receives connection information from the port layer. This state machine also receives Accept_Reject OPENs requests from the SCSI application layer and sends these requests to the port layer.

This state machine consists of one state.

This state machine shall be started after:

a) an Accept_Reject OPENs request is received;
b) a Frame Received confirmation is received; or
c) a HARD_RESET Received confirmation is received.

If this state machine was started as the result of receiving an Accept_Reject OPENs (Accept SSP) or Accept_Reject OPENs (Reject SSP) request, then this state shall send an Accept_Reject OPENs request along with the received argument to the port layer. This state machine shall terminate after sending an Accept_Reject OPENs request to the port layer.

If this state machine was started as the result of a Frame Received (ACK/NAK Balanced) or Frame Received (ACK/NAK Not Balanced) confirmation, then this state shall check the frame type in the received frame. If the confirmation was Frame Received (ACK/NAK Balanced) and the frame type is not XFER_RDY, RESPONSE, or DATA, then this state machine shall discard the frame and terminate. 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 and terminate.

If the frame type is correct relative to the confirmation, then this state may check that the hashed source SAS address matches the SAS address of the SAS port transmitting the frame and the hashed destination SAS address in the frame matches the SAS address of the SAS port receiving the frame based on the connection. If this state checks these SAS addresses and they do not match, then this state machine shall terminate.

If the frame type is:

a) XFER_RDY, then this state shall send a XFER_RDY Arrived message to the ST_ISF1:Send_Frame state specified by the tag;
b) RESPONSE, then this state shall send a Response Arrived message to the ST_IPR state machine specified by the tag; or
c) DATA, then this state shall send a Data-In Arrived message to the ST_IPD state machine specified by the tag.

Each of these messages shall contain the content of the SAS frame. If the tag does not specify a valid state machine, then this state shall discard the frame and may send a vendor-specific confirmation to the SCSI application layer to abort the command using that tag.

If this state machine was started as a result of a HARD_RESET Received confirmation, then this state shall send a Transport Reset event notification to the SCSI application layer.

This state machine shall terminate after sending a message or confirmation.

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).

If implemented, this state machine shall maintain the timers listed in table 4.

Table 4 — 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 10.2.6.2).

Figure 3 shows the ST_T state machines.

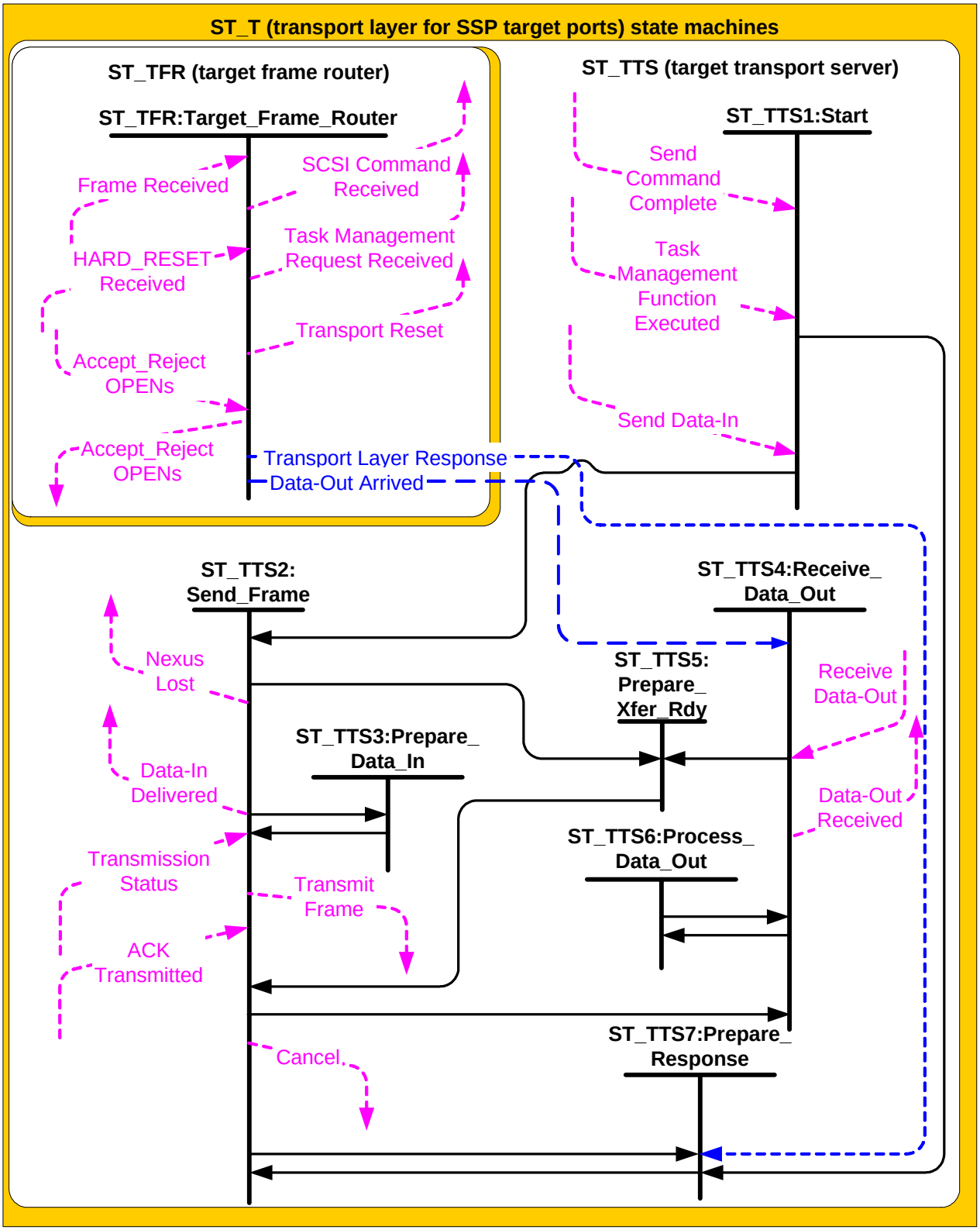

Figure 3 — ST_T (transport layer for SSP target ports) state machines

9.2.6.3.2 ST_TFR (target frame router) state machine

The ST_TFR state machine receives confirmations from the port layer and sends a transport protocol service indication to the SCSI application layer or a message to the ST_TTS state machine. This state machine also receives Accept_Reject OPENs requests from the application layer and sends corresponding requests to the port layer.

This state machine consists of one state.

This state machine shall be started after:

a) an Accept_Reject OPENs request is received;
b) a Frame Received confirmation is received; or
c) a HARD_RESET Received confirmation is received.

If this state machine was started as the result of receiving an Accept_Reject OPENs (Accept SSP) or Accept_Reject OPENs (Reject SSP) request, then this state machine shall send a corresponding Accept_Reject OPENs request to the port layer. This state machine shall terminate after sending an Accept_Reject OPENs request to the port layer.

If this state machine was started as the result of receiving 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 94). If the frame type is not COMMAND, TASK, or DATA, then this state machine shall discard the frame and terminate.

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 and terminate.

This state machine may check that reserved fields in the frame are zero. If any reserved fields are not zero, then this state machine may send a Transport Layer Response (Invalid Frame) message to the ST_TTS state machine including the logical unit number and tag.

NOTE 1 - This check only applies to reserved fields defined in the SSP frame formats (e.g. formats defined in this clause), not reserved fields within the CDB in a COMMAND frame. Handling checking of reserved fields in a CDB is described in SAM-3.

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

If the frame type is DATA, and the tag does not match a tag for an outstanding data-out command, then this state machine shall discard the frame and terminate.

If the frame type is DATA, and the tag matches a tag for an outstanding data-out command without first burst data for which no XFER_RDY frame is outstanding, then this state machine shall discard the frame and terminate.

If the frame type is COMMAND, then this state machine shall check the length of the information unit. If the length of the information unit is not correct (see 9.2.2.2), then this state machine shall send a Transport Layer Response (Invalid Frame) message to the ST_TTS state machine including the logical unit number and tag.

If the frame type is TASK, then this state machine shall check the length of the information unit. If the length of the information unit is not correct (see 9.2.2.2), then this state machine shall send a Transport Layer Response (Invalid Frame) message to the ST_TTS state machine including the logical unit number and tag.

If the frame type is COMMAND and the length of the information unit is correct, then this state machine shall send a SCSI Command Received transport protocol service indication to the SCSI application layer.

If the frame type is TASK, then this state machine shall check the logical unit number. If there is no logical unit at the specified logical unit number, then this state machine shall send a Transport Layer Response (Invalid Logical Unit Number) message to the ST_TTS state machine including the logical unit number and tag.

If the frame type is COMMAND or TASK, then this state machine may check the target port transfer tag. If target port transfer tag is invalid, then this state machine may send a Transport Layer Response (Invalid Frame) message to the ST_TTS state machine including the logical unit number and tag.

If the frame type is TASK and the length of the information unit is correct, then this state may check if the tag conflicts with an existing tag (i.e., an existing command or task management function). If the tag is checked and it conflicts, this state shall send a Transport Layer Response (Invalid Frame) message to the ST_TTS state machine including the logical unit number and tag. If it does not check the tag or the tag does not conflict, this state machine shall send a Task Management Request Received transport protocol service indication to the SCSI application layer. If the frame type is DATA, then this state machine shall send a Data-Out Arrived message to the ST_TTS4:Receive_Data_Out state. Each indication or message shall contain the content of the SAS frame.

If this state machine was started as the result of receiving a HARD_RESET Received confirmation, then this state machine shall send a Transport Reset event notification to the SCSI application layer and terminate.

This state machine shall terminate after sending a message, transport protocol service indication, or event notification.

9.2.6.3.3 ST_TTS (target transport server) state machine

9.2.6.3.3.1 ST_TTS state machine overview

The ST_TTS state machine performs the following functions:

a) processes and sends transport protocol service confirmations to the SCSI application layer;
b) receives and processes transport protocol service requests and responses from the SCSI application layer; and
c) communicates with the port layer via requests and confirmations regarding frame transmission.

This state machine consists of the following states:

a) ST_TTS1:Start (see 9.2.6.3.3.2);
b) ST_TTS2:Send_Frame (see 9.2.6.3.3.3);
c) ST_TTS3:Prepare_Data_In (see 9.2.6.3.3.4);
d) ST_TTS4:Receive_Data_Out (see 9.2.6.3.3.5);
e) ST_TTS5:Prepare_Xfer_Rdy (see 9.2.6.3.3.6);
f) ST_TTS6:Process_Data_Out (see 9.2.6.3.3.7); and
g) ST_TTS7:Prepare_Response (see 9.2.6.3.3.8).

This state machine shall be started in the ST_TTS1:Start state if one of the following is received:

a) a Send Data-In transport protocol service request;
b) a Task Management Function Executed transport protocol service response; or
c) a Send Command Complete transport protocol service response.

This state machine shall be started in the ST_TTS4:Receive_Data_Out state if it is not already running and:

a) a Receive Data-Out transport protocol service request is received; or
b) a Data-Out Arrived message is received and first burst is enabled.

This state machine shall be started in the ST_TTS7:Prepare_Response state if it is not already running and a a Transport Layer Response message is received.

9.2.6.3.3.2 ST_TTS1:Start state

9.2.6.3.3.2.1 State description

The request or response that caused this state machine to be started includes the following to be used in any OPEN address frames required to service the request or response:

a) connection rate;
b) initiator connection tag; and
c) destination SAS address.

A Send Data-In transport protocol service request also includes the following:

a) logical unit number;
b) tag;
c) device server buffer (e.g., starting logical block address); and
d) request byte count (e.g., transfer length).

A Task Management Function Executed transport protocol service response or Send Command Complete transport protocol service response also includes the following:

a) logical unit number;
b) tag;
c) task management function; and
d) tag of task to be managed.

9.2.6.3.3.2.2 Transition ST_TTS1:Start to ST_TTS2:Send_Frame

This transition shall occur after receiving a Send Data-In transport protocol service request and shall include all the information received in the Send Data-In transport protocol service request as arguments.

9.2.6.3.3.2.3 Transition ST_TTS1:Start to ST_TTS7:Prepare_Response

This transition shall occur after receiving a Task Management Function Executed transport protocol service response or a Send Command Complete transport protocol service response and shall include all the information received in the Task Management Function Executed transport protocol service response or Send Command Complete transport protocol service response as arguments.

9.2.6.3.3.3 ST_TTS2:Send_Frame state

9.2.6.3.3.3.1 State description

If this state is entered from the ST_TTS3:Prepare_Data_In state for transmission of a DATA frame, then this state shall send a Transmit Frame (Non-Interlocked) request to the port layer.

If this state is entered from the ST_TTS5:Prepare_Xfer_Rdy state for transmission of an XFER_RDY frame and this state has received an ACK Transmitted confirmation for each DATA frame previously received (i.e., received by this state machine with a Data-Out Arrived message and acknowledged), then this state shall send a Transmit Frame (Interlocked) request to the port layer.

If this state is entered from the ST_TTS7:Prepare_Response state for transmission of a RESPONSE frame and this state has received an ACK Transmitted confirmation for each DATA frame previously received (i.e., received by this state machine with a Data-Out Arrived message and acknowledged), then this state shall send a Transmit Frame (Interlocked) request to the port layer.

NOTE 2 - The XFER_RDY and RESPONSE frame rules ensure that wide ports do not send an XFER_RDY or RESPONSE frame on a phy until all the ACKs have been transmitted for write DATA frames on a different phy. In a narrow port, the link layer ensures that ACK/NAKs are balanced before transmitting an interlocked frame.

A Transmit Frame request from this state shall include the SSP frame and the following to be used for any OPEN address frame:

a) the initiator port bit set to zero;
b) protocol set to SSP;
c) connection rate;
d) initiator connection tag;
e) destination SAS address; and
f) source SAS address set to the SAS address of the SSP target port.

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), this state shall send a Nexus Lost confirmation to the SCSI application layer.

If the confirmation is Transmission Status (Frame Transmitted) confirmation, then this state machine shall expect to receive one of the following confirmations for the frame:

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 frame transmitted was an XFER_RDY frame or a RESPONSE frame, then the state machine shall wait to receive one of those confirmations before transitioning from this state.

If the frame transmitted was a DATA frame, then the state machine may transition to ST_TTS3:Prepare_Data_In as described in 9.2.6.3.3.3.3 without waiting for one of those confirmations.

If transport layer retries are enabled, this state machine shall remember the data offset for the last DATA frame transmitted for which ACK/NAK Balance was achieved (i.e., when the number of DATA frames sent matches the number of ACK Received confirmations received) for use as the restart point in case of a retry.

If:

a) for a DATA frame, this state has received a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation for each of the DATA frames transmitted and the number of bytes moved for the Send Data-In transport protocol service request equals the Request Byte Count; or
b) for a RESPONSE frame, this state receives a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK Received) confirmation,

then this state shall send a Data-In Delivered (Delivery Result = DELIVERY SUCCESSFUL) transport protocol service confirmation to the SCSI application layer.

If:

a) for a DATA or XFER_RDY frame, this state receives a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (NAK Received) confirmation; and
b) the vendor-specific number of retries for the DATA or XFER_RDY frame has been exceeded,

then this state shall send a Data-In Delivered (Delivery Result = DELIVERY FAILURE - NAK RECEIVED) transport protocol service confirmation to the SCSI application layer.

If:

a) for a DATA or XFER_RDY frame, this state receives a Transmission Status (Frame Transmitted) confirmation followed by a Transmission Status (ACK/NAK Timeout) or Transmission Status (Connection Lost Without ACK/NAK) confirmation; and
b) the vendor-specific number of retries for the DATA or XFER_RDY frame has been exceeded,

then this state shall send a Data-In Delivered (Delivery Result = DELIVERY FAILURE - ACK/NAK TIMEOUT) transport protocol service confirmation to the SCSI application layer.

A Data-In Delivered transport protocol service confirmation to the SCSI application layer shall include the following:

a) any argument received from the port layer (e.g., Transmission Status (Frame Transmitted) or Service Delivery Subsystem Failure); and
b) I_T_L_x nexus information (i.e., destination SAS address and tag).

This state machine shall terminate after sending the Data-In Delivered confirmation.

This state may also send a Cancel request to the port layer to cancel a previous Transmit Frame request. A Cancel request shall include the following arguments:

a) the destination SAS address; and
b) the tag.

This state machine shall terminate upon receipt of a Transmission Status (Cancel Acknowledge) confirmation.

9.2.6.3.3.3.3 Transition ST_TTS2:Send_Frame to ST_TTS3:Prepare_Data_In

This transition shall occur if:

a) this state machine was started as the result of receiving a Send Data-In transport protocol service request;
b) the specified values are included with the request; and
c) this state has received an ACK Transmitted confirmation for the COMMAND frame,

or if:

a) this state receives a Transmission Status (Frame Transmitted) confirmation for a DATA frame; and
b) the number of bytes moved for the Send Data-In transport protocol service request is less than the Request Byte Count.

NOTE 3 - The COMMAND frame rule ensures that ports do not send a read DATA frame until the ACK has been transmitted for the COMMAND frame.

The transition shall occur and include a Retry argument if:

a) this state receives one of the following confirmations for a DATA frame:
   A) Transmission Status (NAK Received);
   B) Transmission Status (ACK/NAK Timeout); or
   C) Transmission Status (Connection Lost Without ACK/NAK),
b) the TRANSPORT LAYER RETRIES bit is set to one in the Protocol-Specific Logical Unit mode page (see 10.x.x.x); and
c) the vendor-specific number of retries for the DATA frame has not been exceeded.

Editor's Note 3: the above Retry reason is new

9.2.6.3.3.3.4 Transition ST_TTS2:Send_Frame to ST_TTS4:Receive_Data_Out

This transition shall occur after receiving a Transmission Status (ACK Received) confirmation for an XFER_RDY frame.

9.2.6.3.3.3.5 Transition ST_TTS2:Send_Frame to ST_TTS5:Prepare_Xfer_Rdy

This transition shall occur and include a Retry argument if:

a) this state receives one of the following Transmission Status confirmations for a XFER_RDY frame:
   A) Transmission Status (NAK Received);
   B) Transmission Status (ACK/NAK Timeout); or
   C) Transmission Status (Connection Lost Without ACK/NAK),
b) the TRANSPORT LAYER RETRIES bit is set to one in the Protocol-Specific Logical Unit mode page (see 10.x.x.x); and
c) the vendor-specific number of retries for the XFER_RDY frame has not been exceeded.

Editor's Note 4: this is a new transition

9.2.6.3.3.3.4 Transition ST_TTS2:Send_Frame to ST_TTS7:Prepare_Response

This transition shall occur and include a Retry argument after receiving one of the following Transmission Status confirmations for a RESPONSE frame:

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

This transition shall occur if the vendor-specific number of retries for a DATA frame has been exceeded.

Editor's Note 5: this is a new reason

9.2.6.3.3.4 ST_TTS3:Prepare_Data_In state

9.2.6.3.3.4.1 State description

This state fetches data from the Device Server Buffer and constructs a DATA frame. This state shall use the logical unit number and tag received from the SCSI application layer to construct the frame.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes;
f) fill bytes;
g) data offset; and
h) data.

If this state is entered from the ST_TTS2:Send_Frame state with a Retry argument, the CHANGING DATA POINTER bit shall be set to one and data offset shall be set to a data offset associated with a previous ACK/NAK balance. Otherwise, the CHANGING DATA POINTER bit shall be set to zero and the data offset shall be set to the data offset of the previous DATA frame plus the size of the previous DATA frame.

9.2.6.3.3.4.2 Transition ST_TTS3:Prepare_Data_In to ST_TTS2:Send_Frame

This transition shall occur after constructing a DATA frame.

9.2.6.3.3.5 ST_TTS4:Receive_Data_Out state

9.2.6.3.3.5.1 State description

If a Receive Data-Out transport protocol service request caused this state machine to be started, then the request includes the following to be used in any OPEN address frames required to service the request:

a) connection rate;
b) initiator connection tag;
c) destination SAS address;
d) logical unit number;
e) tag;
f) device server buffer (e.g., starting logical block address); and
g) request byte count (e.g., transfer length).

If a Data-Out Arrived message caused this state machine to be started (i.e., first burst is enabled) and a Receive Data-Out transport protocol service request has not been received, then this state shall wait to process the Data-Out Arrived message until this state receives a Receive Data-Out transport protocol service request. The data received in the Data-Out Arrived message shall be saved in a first burst buffer until this state receives a Receive Data-Out transport protocol service request.

After this state receives a Receive Data-Out transport protocol service request and a Data-Out Arrived message, then this state shall verify the received data frame as follows:

1) check the target port transfer tag value in the DATA frame if the target port transfer tag is being used. If the value is incorrect, then this state shall discard the frame and terminate;
2) check the data offset. If the data offset was not expected (i.e., the CHANGING DATA POINTER bit is set to 1 and the data offset is not set to a data offset associated with a previous ACK/NAK balance, or the CHANGING DATA POINTER bit is set to zero and the data offset is not set to the previous data offset plus

the previous data size), then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - DATA OFFSET ERROR) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation;

3) check the length of the data. If an XFER_RDY frame was sent for the data (i.e., it is not first burst data) and the length of the data exceeds that specified by the XFER_RDY frame that requested the data, then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - TOO MUCH WRITE DATA) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation; and
4) check the length of the data. If the length of the data is zero, then this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - INFORMATION UNIT TOO SHORT) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation.

If the Initiator Response Timeout timer is implemented, this state shall initialize and start the Initiator Response Timeout timer after any of the following occur:

a) a Data-Out Arrived message is received;
b) this state is entered from the ST_TTS2:Send_Frame state; or
c) this state is entered from the ST_TTS6:Process_Data_Out state.

If the Initiator Response Timeout timer expires this state shall send a Data-Out Received (Delivery Result = DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. This state machine shall terminate after sending the confirmation.

If this state is entered from the ST_TTS2:Send_Frame state without a Retry argument, then this state shall wait for a Data-Out Arrived message.

If this state is entered from the ST_TTS6:Process_Data_Out state and the number of bytes moved for the Receive Data-Out transport protocol service request is less than the Request Byte Count, then this state shall wait for a Data-Out Arrived message.

If this state is entered from the ST_TTS6:Process_Data_Out state and number of bytes moved for the Receive Data-Out transport protocol service request equals the Request Byte Count, then this state shall send a Data-Out Received (Delivery Result = DELIVERY SUCCESSFUL) transport protocol service confirmation to the SCSI application layer. This confirmation shall include the tag. If this state has no more bytes in its first burst buffer, then this state machine shall terminate after sending the confirmation. If this state has more bytes to move in its first burst buffer, then this state machine shall wait for a Receive Data-Out transport protocol service request.

9.2.6.3.3.5.2 Transition ST_TTS4:Receive_Data_Out to ST_TTS5:Prepare_Xfer_Rdy

This transition shall occur after:

a) this state has received a Receive Data-Out transport protocol service request and first burst is not enabled; or
b) this state has received a Receive Data-Out transport protocol service request, first burst is enabled, all the first burst data has been processed, and the first burst data did not satisfy the requested byte count.

9.2.6.3.3.5.3 Transition ST_TTS4:Receive_Data_Out to ST_TTS6:Process_Data_Out

This transition shall occur after:

a) a Receive Data-Out transport protocol service request is received; and
b) a Data-Out Arrived message is received and verified (see 9.2.6.3.3.5.1).

9.2.6.3.3.6 ST_TTS5:Prepare_Xfer_Rdy state

9.2.6.3.3.6.1 State description

This state shall construct an XFER_RDY frame. This state shall use the following values received from the SCSI application layer to construct the frame:

a) logical unit number;
b) tag;
c) target port transfer tag;
d) requested offset; and
e) write data length.

If first burst is enabled, this state shall adjust the write data length to reflect the amount of first burst data.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
b) retry data frames bit set to the value of the TRANSPORT LAYER RETRIES bit in the Protocol-Specific Logical Unit mode page; and
a) number of fill bytes.

If:

a) this state was entered from the ST_TTS2:Send_Frame state with a Retry argument (i.e., the XFER_RDY frame transmission was unsuccessful and transport layer retries are supported); and
b) the vendor-specific number of retries for the XFER_RDY frame has not been exceeded,

then this state shall generate a XFER_RDY frame using all of the values for the previous XFER_RDY frame except that:

a) the RETRANSMIT bit shall be set to one; and
b) the TARGET PORT TRANSFER TAG field shall be set to a different value than the previous XFER_RDY frame. The new TARGET PORT TRANSFER TAG field value shall not conflict with any other target port transfer tag currently in use.

9.2.6.3.3.6.3 Transition ST_TTS5:Prepare_Xfer_Rdy to ST_TTS2:Send_Frame

This transition shall occur after constructing an XFER_RDY frame.

9.2.6.3.3.7 ST_TTS6:Process_Data_Out state

9.2.6.3.3.7.1 State description

This state shall process the data received in the Data-Out Arrived message using the Device Server Buffer (e.g., logical block address) to which the data is to be transferred.

9.2.6.3.3.7.2 Transition ST_TTS6:Process_Data_Out to ST_TTS4:Receive_Data_Out

This transition shall occur after data received in a Data-Out message has been processed.

9.2.6.3.3.8 ST_TTS7:Prepare_Response state

9.2.6.3.3.8.1 State description

This state shall construct a RESPONSE frame if this state was entered as the result of this state machine either:

a) receiving a Transport Layer Response message;
b) receiving a Task Management Function Executed transport protocol service response; or
c) receiving a Send Command Complete transport protocol service response.

If this state was entered as the result of receiving a Transport Layer Response message, this state shall use the logical unit number and tag received in the message and shall construct the frame as described in table 5.

Table 5 — Transport Layer Response argument to RESPONSE frame content mapping

Transport Layer Response argument	RESPONSE frame
Information Unit Too Short	The DATAPRES field shall be set to SENSE_DATA, the STATUS field shall be set to CHECK CONDITION and the additional sense code shall be set to INFORMATION UNIT TOO SHORT.
Information Unit Too Long	The DATAPRES field shall be set to SENSE_DATA, the STATUS field shall be set to CHECK CONDITION and the additional sense code shall be set to INFORMATION UNIT TOO LONG.
Invalid Frame	The DATAPRES field shall be set to RESPONSE_DATA and the RESPONSE CODE field shall be set to INVALID FRAME.
Invalid Logical Unit Number	The DATAPRES field shall be set to RESPONSE_DATA and the RESPONSE CODE field shall be set to INVALID LOGICAL UNIT NUMBER.

If this state was entered as a result of receiving a Task Management Function Executed transport protocol service response or a Send Command Complete transport protocol service response, this state shall use the following values received from the SCSI application layer to construct the frame:

a) logical unit number;
b) tag;
c) status;
d) response data; and
e) sense data.

This state shall generate the following values when constructing the frame:

a) frame type;
b) hashed destination SAS address;
c) hashed source SAS address;
d) retransmit bit set to zero;
e) number of fill bytes;
f) fill bytes;
g) data present;
h) sense data length; and
i) response data length.

If:

a) this state was entered from the ST_TTS2:Send_Frame state with a Retry argument (i.e., the RESPONSE frame transmission was unsuccessful); and
b) the vendor-specific number of retries for the RESPONSE frame has not been exceeded,

then this state shall generate a RESPONSE frame using all of the values for the previous RESPONSE frame except that the RETRANSMIT bit shall be set to one.

9.2.6.3.3.8.3 Transition ST_TTS7:Prepare_Response to ST_TTS2:Send_Frame

This transition shall occur after constructing a RESPONSE frame or if the vendor-specific number of retries for ~~transmission of~~ the RESPONSE frame has been exceeded.

10.2.2 Application client error handling [modifications throughout]

If an SSP initiator port calls Command Complete Received () and delivers a Service Response of:

a) Service Delivery or Target Failure - XFER_RDY Information Unit Too Short;

b) Service Delivery or Target Failure - XFER_RDY Information Unit Too Long;
c) Service Delivery or Target Failure - XFER_RDY Incorrect Write Data Length;
d) Service Delivery or Target Failure - XFER_RDY Requested Offset Error;
e) Service Delivery or Target Failure - XFER_RDY Not Expected;
f) Service Delivery or Target Failure - DATA Incorrect Data Length;
g) Service Delivery or Target Failure - DATA Too Much Read Data;
h) Service Delivery or Target Failure - DATA Data Offset Error;
i) Service Delivery or Target Failure - NAK Received; or
j) Service Delivery or Target Failure - Connection Failed,

it shall abort the command (e.g., by sending an ABORT TASK task management function).

After an application client calls Send SCSI Command (), if Command Complete Received () returns a Service Response of Service Delivery or Target Failure - ACK/NAK Timeout, the application client shall send a QUERY TASK task management function with Send Task Management Request () to determine whether the command was received successfully. If Received Task Management Function Executed () returns a Service Response of:

a) FUNCTION SUCCEEDED: the application client shall assume the command was delivered successfully; or
b) FUNCTION COMPLETE: if Command Complete Received () has not yet been invoked a second time for the command in question (e.g., indicating a RESPONSE frame arrived for the command before the QUERY TASK was processed), the application client shall assume the command was not delivered successfully and may reuse the tag. The application client should call Send SCSI Command () again with identical arguments.

If an application client calls Send Task Management Request () and an SSP initiator port calls Received Task Management Function Executed () and delivers a Service Response of Service Delivery or Target Failure - ACK/NAK Timeout, the application client should call Send Task Management Request () again with identical arguments.

If an application client calls Send SCSI Command () and an SSP initiator port calls Command Complete Received () a second time for the command, and the second call includes a Retransmit argument set to one, the application client shall ignore the second call.

If an application client calls Send Task Management Request () and an SSP initiator port calls Received Task Management Function Executed () a second time for the task management function, and the second call includes a Retransmit argument set to one, the application client shall ignore the second call.

After a Command Complete Received () or Received Task Management Function Executed () call, an application client shall not reuse the tag until it determines the tag is no longer in use by the logical unit (e.g., the ACK for the RESPONSE frame was seen by the SSP target port). Examples of ways the application client may determine that a tag may be used are:

a) receiving another frame in the same connection;
b) receiving a DONE (NORMAL) or DONE (CREDIT TIMEOUT) in the same connection; or
c) receiving a DONE (ACK/NAK TIMEOUT) in the same connection, then running a QUERY TASK task management function to confirm that the tag is no longer active in the logical unit.

10.2.3 Device server error handling [no changes]

If a device server calls Receive Data-Out () and receives a Delivery Result set to a value in table 128, it shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code as indicated by table 128.

...

10.2.6.4 Protocol-Specific Logical Unit mode page

The Protocol-Specific Logical Unit mode page (see SPC-3) contains parameters that affect SSP target port operation on behalf of the logical unit. If the mode page is implemented, the mode page policy shall be shared (i.e., there is one copy of the mode page shared by all SSP initiator ports)(see SPC-3).

If a SAS target device has multiple SSP target ports, changes in the short page parameters for one SSP target port should not affect other SSP target ports.

Table 6 defines the subpages of this mode page.

Table 6 — Protocol-Specific Logical Unit mode page subpages

Subpage	Description	Reference
Short page	Short format	10.2.6.4.1
Long page 00h	Not allowed	
Long page E0h - FEh	Vendor specific	
Long page FFh	Return all subpages for the Protocol-Specific Port mode page	SPC-3
All others	Reserved	

10.2.6.4.1 Protocol-Specific Logical Unit mode page - short format

Parameters in this page shall affect all phys in the SSP target port, and may affect all SSP target ports in the SAS target device.

Table 7 defines the format of the page for SAS SSP.

Table 7 — Protocol-Specific Logical Unit mode page for SAS SSP - short format

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (18h)					
1	PAGE LENGTH (06h)							
2	Reserved			TRANSPORT LAYER RETRIES	PROTOCOL IDENTIFIER (6h)			
3	Reserved							
4	Reserved							
7								

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-3.

The SPF field shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 18h.

The PAGE LENGTH field shall be set to 06h.

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

A TRANSPORT LAYER RETRIES bit set to one specifies that the target port shall support transport layer retries for XFER_RDY and DATA frames for the logical unit as described in 9.x.x.x. A TRANSPORT LAYER RETRIES bit set to zero specifies that transport layer retries shall not be used.