To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com) and Jim Jones, Quantum (jim.jones@quantum.com)
Date: 28 July 2003
Subject: T10/03-186r1 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 July SAS working group meeting.

Related Documents

sas-r05 – Serial Attached SCSI - 1.0 revision 5
sas1r00 – Serial Attached SCSI - 1.1 revision 0
03-229r1 SAS-1.1 Transport layer retries ladders

Overview

In SAS-1, errors transmitting frames result in either the logical unit terminating the command with CHECK CONDITION status, or the application client aborting the command. Possible errors are:

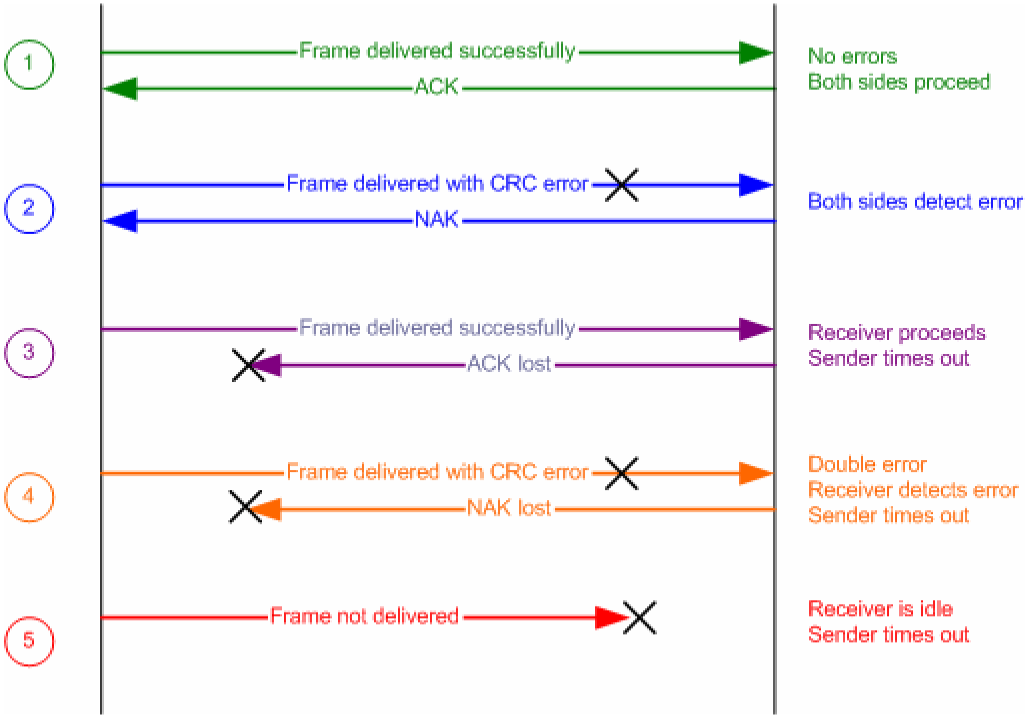

Figure 1. Error cases

Depending on the type of frame, different things happen in SAS-1.0:

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 with CRC error; NAK arrives 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 with 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 ACKs/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 would implement these features; disk drives probably would not. A Mode page per LUN to enable/disable special recovery mode (not the Enable Modify Data Pointers bit in the Disconnect-Reconnect mode page) would be added.

RESPONSE problems

- 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.
- If target detects an ACK/NAK timeout, resend as in SAS-1.0 (with **Retransmit**=1)
- Target shall retry RESPONSE problems at least one time.

COMMAND or TASK problems

- If initiator gets a NAK, resend as in SAS-1.0
- 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.
- 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.
- 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:
  - If FUNCTION SUCCEEDED, then assume it's running fine. If first burst is enabled, start sending write DATA frames.
  - If FUNCTION COMPLETE, then resend the command.
  - Continue watching for a RESPONSE frame while running QUERY TASK.
  - If a RESPONSE shows up, don't try to abort the command and don't report it as aborted to the application client.
  - 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

- 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
- If target detects an ACK/NAK timeout, resend with **Retransmit**=1 and a new TPTT
- Target shall retry XFER_RDY problems at least one time.
- If target sees write DATA frames before seeing an ACK for the XFER_RDY, target discards them
- Write DATA frames in response to an XFER_RDY with **Retransmit**=1 do not themselves have **Retransmit**=1
- Target shall change Target Port Transfer Tag between XFER_RDYs (either retransmitted or normal)
- 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.
- The target can reuse the first TPTT when it sees a data frame with the second TPTT.

- 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

- 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).
- 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.)
- 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)
- Each resent write DATA frame has the correct Data Offset
- No use of the **Retransmit** bit for write DATA frames
- First resent write DATA frame has **Changing Data Pointer**=1
- 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)
    - 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.
    - 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..
- 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).
- 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

- 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.
- If target detects an ACK/NAK timeout, it shall resend all read DATA frames since the last ACK/NAK balance point.
- If target gets a NAK, it should stop sending read DATA frames.
- No use of **Retransmit** bit for read DATA frames
- Set the **Changing Data Pointer** bit to 1 when the target is going back to an earlier data offset
- Target frames shall be monotonically increasing after **Changing Data Pointer**=1
- 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).
- 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).

- SAS-1.0 only allowed this bit be used for RESPONSE frames.
- 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)

- 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)

- 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):

- **Optimized Recovery** bit (located anywhere)
- 1 = the device server in the logical unit instructs the target port to:
    - set XFER_RDY frame header **Retry Data Frames** bit to 1
    - retry read DATA frames that fail
    - 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:

- 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.
- 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:

- 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.
- 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:

- 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.
- 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.

Detailed proposal in standardese

9.2 SSP transport layer

9.2.1 SSP frame format

Table 92 defines the SSP frame format.

Table 92 — SSP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0 to 9	as is							
10						RETRY DATA FRAMES	RETRANSMIT	CHANGING DATA POINTER
11 to n	as is							

...

The RETRY DATA FRAMES bit is set to one for XFER_RDY frames under certain conditions (see 9.2.4) and shall be set to zero for all other frame types. 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~~) and shall be set to zero for all other frame types. 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 certain conditions (see 9.2.4) and shall be set to zero for all other frame types. This bit indicates the frame is a retransmission after the SSP target port failed in its previous attempt to transmit this or a subsequent frame and the data offset field is not sequentiall increasing.

...

9.2.4 SSP transport layer handling of link layer errors

9.2.4.0 Transport layer handling of link layer errors overview

If an SSP initiator port is exchanging frames with a logical unit that supports transport layer retries, it shall follow the rules in 9.2.4.2. If it has reached its retry limit or is exchanging frames with a logical unit that does not support transport layer retries, it shall follow the rules in 92.4.3.

Logical units supporting transport layer retries shall:
a) support the QUERY TASK task management function; and
b) select a different value for the TARGET PORT TRANSFER TAG field in an XFER_RDY frame than that used in the previous XFER_RDY frame.

9.2.4.2.0 Transport layer retries

9.2.4.3.1 COMMAND frame

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall close the connection with DONE (ACK/NAK TIMEOUT) and transmit a QUERY TASK task management function in the next connection to determine whether the command was received. If QUERY TASK returns a TASK MANAGEMENT FUNCTION SUCCEEDED response, the SSP initiator port shall assume the COMMAND frame was ACKed. If QUERY TASK returns a TASK MANAGEMENT FUNCTION COMPLETE response, and neither an XFER_RDY frame nor a RESPONSE frame has been received for that I_T_L_Q nexus, the SSP initiator port shall assume the command was NAKed or lost and may reuse the tag (see 10.2.2).

If an XFER_RDY frame or RESPONSE frame arrives before QUERY TASK completes, the SSP initiator shall assume the COMMAND frame was successfully received and the command is being processed (i.e., the XFER_RDY frame or RESPONSE frame is honored).

9.2.4.3.2 TASK frame

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall retransmit the TASK frame in a new connection (see 10.2.2).

9.2.4.3.3 XFER_RDY frame

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and retransmit the XFER_RDY frame with the RETRANSMIT bit set to one and with a different TARGET PORT TRANSFER TAG.

If an SSP target port transmits an XFER_RDY frame and receives a NAK, it shall retransmit the XFER_RDY frame with the RETRANSMIT bit set to one and with a different TARGET PORT TRANSFER TAG.

If an SSP initiator port receives an XFER_RDY frame with the RETRANSMIT bit set to one while it is still processing the previous XFER_RDY frame for that command, 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 shall not send any write DATA frames for the old XFER_RDY after sending a write DATA frame for the new XFER_RDY. The SSP target port may reuse the TARGET PORT TRANSFER TAG value from the first XFER_RDY frame when it receives a write DATA frame for the new XFER_RDY frame.

9.2.4.34 DATA frame

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and retransmit all the read DATA frames since one of the previous ACK/NAK balance points.
If an SSP target port transmits a read DATA frame and receives a NAK, it shall retransmit all the read DATA frames since one of the previous ACK/NAK balance points.

If an SSP initiator port transmits a write DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and retransmit all the write DATA frames for the XFER_RDY. If it receives an XFER_RDY or RESPONSE frame for the command, it shall process the XFER_RDY or RESPONSE frame and should stop sending the retransmitted write DATA frames (but it may continue transmitting them). It shall not send a write DATA frame for the old 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, it shall retransmit all the write DATA frames for the XFER_RDY.

In all cases, the first retransmitted read DATA frame shall have its CHANGING DATA POINTER bit set to one; subsequent read DATA frames shall have their CHANGING DATA POINTER bits set to zero.

9.2.4.3.5 RESPONSE frame

If an SSP target port transmits a RESPONSE frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and retransmit the RESPONSE frame with the RETRANSMIT bit set to one in a new connection. It shall do this at least one time.

If an SSP target port transmits a RESPONSE frame and receives a NAK, it shall retry transmitting the RESPONSE frame at least one time (see 9.2.6.3.3.8).

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 shall discard the extra RESPONSE frame. If it has not previously received the RESPONSE frame, it shall treat it as the valid RESPONSE frame (see 10.2.2).

9.2.4.3.0 No transport layer retries

9.2.4.3.1 COMMAND frame

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall close the connection with DONE (ACK/NAK TIMEOUT) and transmit a QUERY TASK task management function in the next connection to determine whether the command was received. If QUERY TASK returns a TASK MANAGEMENT FUNCTION SUCCEEDED response, the SSP initiator port shall assume the COMMAND frame was ACKed. If QUERY TASK returns a TASK MANAGEMENT FUNCTION COMPLETE response, and a RESPONSE frame has not yet been received for that I_T_L_Q nexus, the SSP initiator port shall assume the command was NAKed or lost and may reuse the tag (see 10.2.2).

9.2.4.3.2 TASK frame

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it shall retransmit the TASK frame in a new connection (see 10.2.2).

9.2.4.3.3 XFER_RDY frame

If an SSP target port transmits an XFER_RDY frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and return 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, it shall return 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).

9.2.4.34 DATA frame

If an SSP target port transmits a read DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and return 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, it shall return 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 (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and abort the command (see 10.2.2).

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

9.2.4.3.5 RESPONSE frame

If an SSP target port transmits a RESPONSE frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it shall close the connection with DONE (ACK/NAK TIMEOUT) and retransmit ~~try transmitting~~ the RESPONSE frame with the RETRANSMIT bit set to one again in a new connection. It shall do this at least one time. ~~The RETRANSMIT bit shall be set to one on each of the retries (see 9.2.6.3.3.8).~~

If an SSP target port transmits a RESPONSE frame and receives a NAK, it shall retransmit ~~retry transmitting~~ the RESPONSE frame at least one time (see 9.2.6.3.3.8).

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 shall discard the extra RESPONSE frame. If it has not previously received the RESPONSE frame, it shall treat it as the valid RESPONSE frame (see 10.2.2).

9.2.6 ST (transport layer for SSP ports) state machines

[state machine modifications go here]