Date: March 11, 2003

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: Section 7 Changes

0.1 Identification and hard reset sequence

0.1.1 Identification and hard reset sequence overview

After the phy reset sequence has been completed indicating the physical link is using SAS rather than SATA, each phy transmits either:

a) an IDENTIFY address frame (see 7.8.2); or
b) a HARD_RESET.

Each phy receives an IDENTIFY address frame or a HARD_RESET from the phy to which it is attached. The combination of a phy reset sequence, an optional hard reset sequence, and an identification sequence is called a link reset sequence ~~(see 4.4.1)~~(see 4.4.1).

If a device supports more than one phy, it shall transmit the same SAS address on all phys for which it is capable of sharing within a port.

If a device detects the same SAS address incoming on different phys, it shall consider those phys part of the same wide port.

If a device detects different SAS addresses incoming on different physical links, it shall consider those physical links as independent physical links and consider those phys part of different ports.

If a device does not receive a valid IDENTIFY address frame within 1 ms of phy reset sequence completion, it shall restart the phy reset sequence.

If a device receives an additional IDENTIFY address frame after receiving the first one, without an intervening phy reset sequence, it shall ignore the additional IDENTIFY address frame.

If a phy receives a HARD_RESET, it shall be considered a reset event and cause a hard reset ~~(see 4.4.2)~~(see 4.4.2) of the port containing that phy.

0.1.2 SAS initiator device rules

After a link reset sequence, or after receiving a BROADCAST (CHANGE), a management application client behind an SMP initiator port should perform a discover process ~~(see 4.6.7.4)~~(see 4.6.7.4).

When a discover process is performed after a link reset sequence, the management application client discovers all the devices in the SAS domain. When a discover process is performed after a BROADCAST (CHANGE), the management application client determines which devices have been added to or removed from the SAS domain.

The discover information may be used to select connection rates for connection requests.

If during the discover process ~~(see 4.6.7.4)~~(see 4.6.7.4) the management application client detects a port with a SAS address it has already encountered, it has found a routing loop and shall break the loop by disabling routing of destination SAS addresses through the expander port used to reach this previously encountered port by using the SMP CONFIGURE ROUTE INFORMATION function ~~(see clause 10.4.3.8)~~(see clause 10.4.3.8).

0.1.3 Fanout expander device rules

After completing the identification sequence, the ECM within a fanout expander device shall be capable of routing connection requests to attached devices.

After a link reset sequence, or after receiving a BROADCAST (CHANGE), the management application client behind an SMP initiator port in a fanout expander device that does not have a configurable expander route table shall follow the SAS initiator device rules (see 0.1.2) to perform a discover process.

The ECM of a fanout expander device that has a configurable expander route table is dependent on the completion of the discover process ~~(see 4.6.7.4)~~(see 4.6.7.4) for routing connection requests using the table routing method.

0.1.4 Edge expander device rules

After completing the identification sequence, the ECM within an edge expander device shall be capable of routing connection requests to attached devices.

The ECM of an edge expander device that has a configurable expander route table is dependent on the completion of the discover process ~~(see 4.6.7.4)~~(see 4.6.7.4) for routing connection requests using the table routing method.

0.1.5 SL_IR (link layer identification and hard reset) state machines

0.1.5.1 SL_IR state machines overview

The SL_IR (link layer identification and hard reset) state machines control the flow of dwords on the physical link that are associated with the identification and hard reset sequences. The state machines are as follows:

a) SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine (see 0.1.5.3);
b) SL_IR_RIF (receive IDENTIFY address frame) state machine (see 0.1.5.4); and
c) SL_IR_IRC (identification and hard reset control) state machine (see 0.1.5.5).

The SL_IR state machines send the following messages to the link layer state machine - ~~SL~~ SL_CC (see 0.6) in SAS devices or XL (see 0.7) in expander devices:

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

The SL_IR state machines shall maintain the timers listed in table 1.

Table 1 — SL_IR timers

Timer	Initial value
Receive Identify Timeout timer	1 ms

Figure 1 shows the SL_IR state machines.

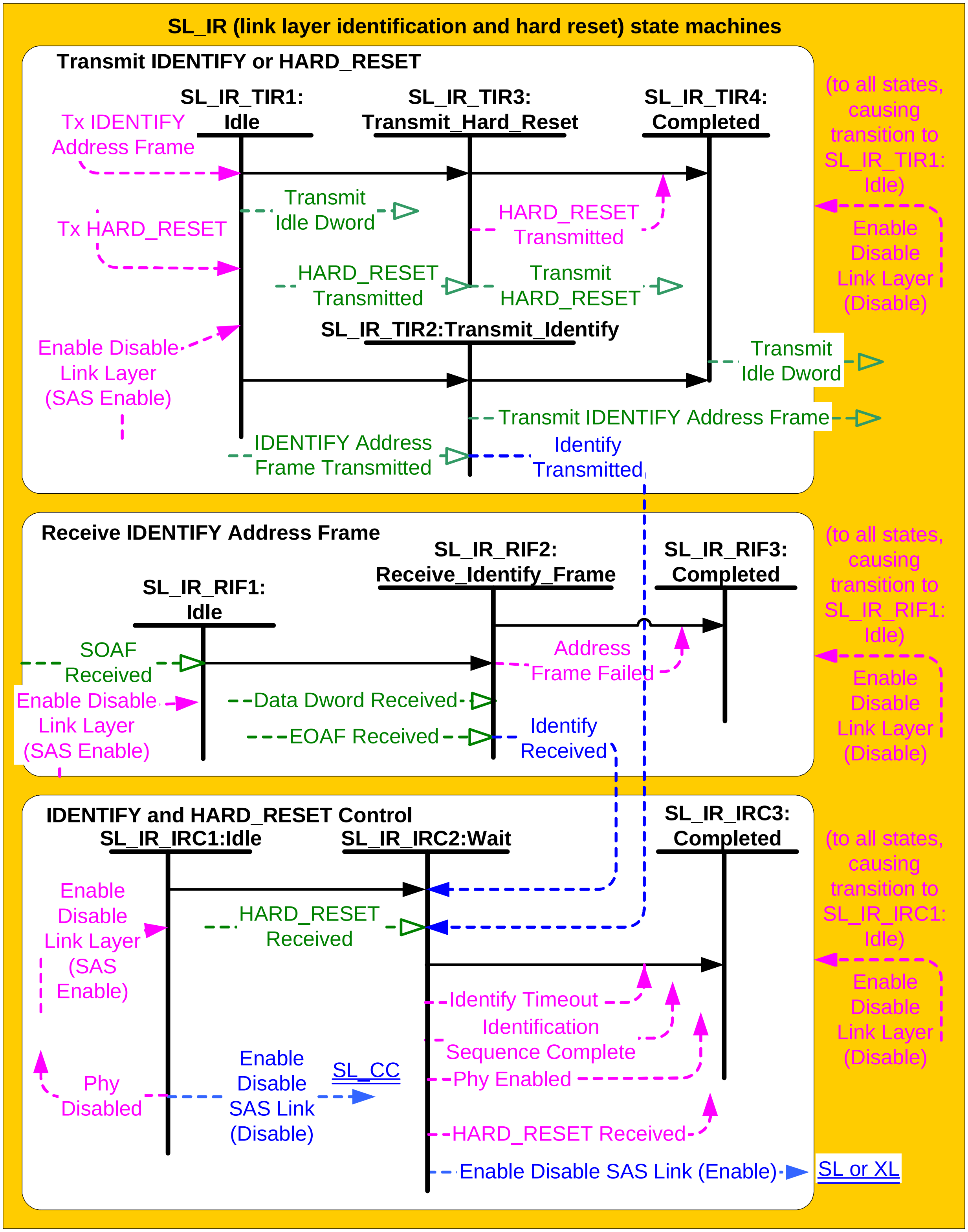

Figure 1 — SL_IR (link layer identification and hard reset) state machines

0.1.5.2 SL_IR transmitter and receiver

The SL_IR transmitter receives the following messages from the SL_IR state machines:

a) Transmit IDENTIFY Address Frame;
b) Transmit HARD_RESET; and
c) Transmit Idle Dword.

The SL_IR transmitter sends the following messages to the SL_IR state machines:

a) HARD_RESET Transmitted; and
b) IDENTIFY Address Frame Transmitted.

The SL_IR receiver sends the following messages to the SL_IR state machines:

a) SOAF Received;
b) Data Dword Received;
c) EOAF Received; and
d) HARD_RESET Received.

When the SL_IR transmitter is requested to transmit a dword from any state within the SL_IR state machines, it shall transmit that dword. The following are dwords that may be transmitted by the SL_IR transmitter:

a) HARD_RESET;
b) SOAF/IDENTIFY address frame/EOAF; or
c) idle dword.

0.1.5.3 SL_IR_TIR (transmit IDENTIFY or HARD_RESET) state machine

0.1.5.3.1 SL_IR_TIR state machine overview

The SL_IR_TIR state machine's function is to transmit a single IDENTIFY address frame or HARD_RESET after the phy layer enables the link layer. This state machine consists of the following states:

a) SL_IR_TIR1:Idle (see 0.1.5.3.2)(initial state);
b) SL_IR_TIR2:Transmit_Identify (see 0.1.5.3.3);
c) SL_IR_TIR3:Transmit_Hard_Reset (see 0.1.5.3.4); and
d) SL_IR_TIR4:Completed (see 0.1.5.3.5).

This state machine shall start in the SL_IR_TIR1:Idle state. This state machine shall transition to the SL_IR_TIR1:Idle state from any other state after receiving an Enable Disable Link Layer (Disable) confirmation.

0.1.5.3.2 SL_IR_TIR1:Idle state

0.1.5.3.2.1 State description

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IR transmitter.

0.1.5.3.2.2 Transition SL_IR_TIR1:Idle to SL_IR_TIR2:Transmit_Identify

This transition shall occur after both:

a) an Enable Disable Link Layer (SAS Enable) confirmation is received; and
b) a Tx IDENTIFY Address Frame request is received.

0.1.5.3.2.3 Transition SL_IR_TIR1:Idle to SL_IR_TIR3:Transmit_Hard_Reset

This transition shall occur after both:

a) an Enable Disable Link Layer (SAS Enable) confirmation is received; and
b) a Tx HARD_RESET request is received.

0.1.5.3.3 SL_IR_TIR2:Transmit_Identify state

0.1.5.3.3.1 State description

Upon entry into this state, this state shall send a Transmit IDENTIFY Address Frame message to the SL_IR transmitter.

After this state receives an IDENTIFY Address Frame Transmitted message, this state shall send an Identify Transmitted message to the SL_IR_IRC state machine.

0.1.5.3.3.2 Transition SL_IR_TIR2:Transmit_Identify to SL_IR_TIR4:Completed

This transition shall occur after this state has sent an Identify Transmitted message.

0.1.5.3.4 SL_IR_TIR3:Transmit_Hard_Reset state

0.1.5.3.4.1 State description

Upon entry into this state, this state shall send a Transmit HARD_RESET message to the SL_IR transmitter.

After this state receives a HARD_RESET Transmitted message, this state shall send a HARD_RESET Transmitted confirmation to the management application layer.

0.1.5.3.4.2 Transition SL_IR_TIR3:Transmit_Hard_Reset to SL_IR_TIR4:Completed

This transition shall occur after sending a HARD_RESET Transmitted confirmation.

0.1.5.3.5 SL_IR_TIR4:Completed state

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL_IR transmitter.

0.1.5.4 SL_IR_RIF (receive IDENTIFY address frame) state machine

0.1.5.4.1 SL_IR_RIF state machine overview

The SL_IR_RIF state machine receives an IDENTIFY address frame and checks the IDENTIFY address frame to determine if the frame should be accepted or discarded by the link layer.

This state machine consists of the following states:

a) SL_IR_RIF1:Idle (see 0.1.5.4.2)(initial state);
b) SL_IR_RIF2:Receive_Identify_Frame (see 0.1.5.4.3); and
c) SL_IR_RIF3:Completed (see 0.1.5.4.4).

This state machine shall start in the SL_IR_RIF1:Idle state. This state machine shall transition to the SL_IR_RIF1:Idle state from any other state after receiving an Enable Disable Link Layer (Disable) confirmation.

0.1.5.4.2 SL_IR_RIF1:Idle state

0.1.5.4.2.1 State description

This state waits for an SOAF to be received from the physical link, indicating an address frame is arriving.

0.1.5.4.2.2 Transition SL_IR_RIF1:Idle to SL_IR_RIF2:Receive_Identify_Frame

This transition shall occur after both:

a) an Enable Disable Link Layer (SAS Enable) confirmation is received; and
b) an SOAF Received message is received.

0.1.5.4.3 SL_IR_RIF2:Receive_Identify_Frame state

0.1.5.4.3.1 State description

This state receives the dwords of an address frame and the EOAF.

This state shall ignore all primitives (e.g., BREAK and HARD_RESET) except SOAF.

If this state receives an SOAF, then this state shall discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

If this state receives more than eight data dwords after an SOAF and before an EOAF, then this state shall discard the address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

After receiving an EOAF, this state shall check if it the IDENTIFY address frame is valid.

This state shall accept an IDENTIFY address frame and send an Identify Received message to the SL_IR_IRC state machine if:

a) the ADDRESS FRAME TYPE field is set to Identify;
b) the number of bytes between the SOAF and EOAF is 32; and
c) the CRC field contains a valid CRC.

Otherwise, this state shall discard the IDENTIFY address frame and send an Address Frame Failed confirmation to the management application layer to indicate that an invalid IDENTIFY address frame was received.

0.1.5.4.3.2 Transition SL_IR_RIF2:Receive_Identify_Frame to SL_IR_RIF3:Completed

This transition shall occur after sending an Identify Received message or Address Frame Failed confirmation.

0.1.5.4.4 SL_IR_RIF3:Completed state

This state waits for the Enable Disable Link Layer (Disable) confirmation.

0.1.5.5 SL_IR_IRC (identification and hard reset control) state machine

0.1.5.5.1 SL_IR_IRC state machine overview

The SL_IR_IRC state machine ensures that IDENTIFY address frames have been both received and transmitted before enabling the rest of the link layer, and notifies the link layer if a HARD_RESET is received before an IDENTIFY address frame has been received.

This state machine consists of the following states:

a) SL_IR_IRC1:Idle (see 0.1.5.5.2)(initial state);
b) SL_IR_IRC2:Wait (see 0.1.5.5.3); and
c) SL_IR_IRC3:Completed (see 0.1.5.5.4).

This state machine shall start in the SL_IR_IRC1:Idle state. This state machine shall transition to the SL_IR_IRC1:Idle state from any other state after receiving an Enable Disable Link Layer (Disable) confirmation.

0.1.5.5.2 SL_IR_IRC1:Idle state

0.1.5.5.2.1 State description

This state waits for the link layer to be enabled. Upon entry into this state, this state shall:

a) send an Enable Disable SAS Link (Disable) message to the SL state machine (see 0.6) or XL state machine (see 0.7) halting any link layer activity; and
b) send a Phy Disabled confirmation to the port layer and the management application layer indicating that the phy is not ready for use.

0.1.5.5.2.2 Transition SL_IR_IRC1:Idle to SL_IR_IRC2:Wait

This transition shall occur when an Enable Disable Link Layer (SAS Enable) confirmation is received.

0.1.5.5.3 SL_IR_IRC2:Wait state

0.1.5.5.3.1 State description

This state ensures that an IDENTIFY address frame has been received by the SL_IR_RIF state machine and that a IDENTIFY address frame has been transmitted by the SL_IR_TIR state machine before enabling the rest of the link layer. The IDENTIFY address frames may be transmitted and received on the physical link in any order.

After this state receives an Identify Transmitted message, it shall initialize and start the Receive Identify Timeout timer. If an Identify Received message is received before the Receive Identify Timeout timer expires, this state shall:

a) send an Identification Sequence Complete confirmation to the management application layer, with arguments carrying the contents of the incoming IDENTIFY address frame;
b) in an expander device, send an Identification Sequence Complete confirmation to the link layer;
c) send an Enable Disable SAS Link (Enable) message to the SL state machine (see 0.6) in SAS phys or the XL state machine (see 0.7) in expander phys indicating that the rest of the link layer may start operation; and
d) send a Phy Enabled confirmation to the port layer and the management application layer indicating that the phy is ready for use.

If the Receive Identify Timeout timer expires before an Identify Received message is received, this state shall send an Identify Timeout confirmation to the management application layer to indicate that an identify timeout occurred.

If this state receives a HARD_RESET before an Identify Received message is received, this state shall send a HARD_RESET Received confirmation to the port layer.

If this state receives a HARD_RESET after an Identify Received message is received, the HARD_RESET shall be ignored.

0.1.5.5.3.2 Transition SL_IR_IRC2:Wait to SL_IR_IRC3:Completed

This transition shall occur after sending a HARD_RESET Received confirmation, Identify Timeout confirmation, Identification Sequence Complete, and/or Phy Enabled confirmation.

0.1.5.5.4 SL_IR_IRC3:Completed state

This state waits for the Enable Disable Link Layer (Disable) confirmation.

0.2 Power management

SATA interface power management is not supported in SAS.

STP initiator ports shall not generate SATA_PMREQ_P, SATA_PMREQ_S, or SATA_PMACK. If an STP initiator port receives SATA_PMREQ_P or SATA_PMREQ_S, it shall reply with SATA_PMNAK.

If an expander device receives SATA_PMREQ_P or SATA_PMREQ_S from a SATA device while an STP connection is not open, it shall not forward it to any STP initiator port and shall reply with a SATA_PMNAK. If one of these primitives arrives while an STP connection is open, it may forward the primitive to the STP initiator port.

SCSI idle and standby power conditions, implemented with the START STOP UNIT command (see SBC-2 and RBC) and the Power Condition mode page (see SPC-3), may be supported by SSP initiator ports and SSP target ports as described in ~~10.2.8~~10.2.8.

ATA idle and standby power modes, implemented with the IDLE, IDLE IMMEDIATE, STANDBY, STANDBY IMMEDIATE, and CHECK POWER MODE commands (see ATA/ATAPI-7 V1), may be supported by STP initiator ports. The ATA sleep power mode, implemented with the SLEEP command, shall not be used.

0.3 SAS domain changes

After power on or receiving BROADCAST (CHANGE), an application client in each SAS initiator port should scan the SAS domain using the discover process ~~(see 4.6.7.4)~~(see 4.6.7.4) to search for SAS initiator devices, SAS target devices, and expander devices.

The expander device shall transmit BROADCAST (CHANGE) from at least one phy in each expander port other than the expander port that is the cause for transmitting BROADCAST (CHANGE).

Expander devices shall transmit BROADCAST (CHANGE) for the following reasons:

a) after an expander phy has lost dword synchronization;
b) after the link reset sequence completes; and
c) after the expander device receives BROADCAST (CHANGE).

BROADCAST (CHANGE) may be sent by SAS initiator ports to force other SAS initiator ports and expander ports to re-run the discover process, but should not be sent by SAS target ports.

A SAS initiator port that detects BROADCAST (CHANGE) shall follow the SAS initiator device rules (see 0.1.2) to discover the topology.

A fanout expander device that detects BROADCAST (CHANGE) shall follow the fanout device rules (see 0.1.3) to discover the topology.

An edge expander device that detects BROADCAST (CHANGE) shall follow the edge device rules (see 0.1.4).

See ~~10.4.3.2~~10.4.3.2 for details on counting BROADCAST (CHANGE) generation in an expander device.

0.4 Connections

0.4.1 Connections overview

A connection is opened between a SAS initiator port and a SAS target port before communication begins. A connection is established between one SAS initiator phy in the SAS initiator port and one SAS target phy in the SAS target port.

SSP initiator ports open SSP connections to transmit SCSI commands, task management functions, or transfer data. SSP target ports open SSP connections to transfer data or transmit status.

SMP initiator ports open SMP connections to transmit SMP requests and receive SMP responses.

STP initiator ports and STP target ports open STP connections to transmit SATA frames. An STP target port in an expander device opens STP connections on behalf of SATA devices.

The OPEN address frame is used to request that a connection be opened. AIP, OPEN_ACCEPT and OPEN_REJECT are the responses to an OPEN address frame. BREAK is used to abandon connection requests and to unilaterally break a connection. CLOSE is used for orderly closing a connection.

Connections use a single pathway from the SAS initiator phy to the SAS target phy. While a connection is open, only one pathway shall be used for that connection.

For STP connections, connections may be between the STP initiator port and an STP target port in an expander device attached to a SATA device. The SATA device is not aware of SAS connection management.

A wide port may have separate connections on each of its phys.

0.4.2 Opening a connection

0.4.2.1 Connection request

The OPEN address frame (see 7.8.3) is used to open a connection from a source port to a destination port using one source phy and one destination phy.

To make a connection request, the source port shall transmit an OPEN address frame through an available phy. The source phy shall transmit idle dwords after the OPEN address frame until it receives a response or abandons the connection request with BREAK.

After transmitting an OPEN address frame, the source phy shall initialize and start a 1 ms Open Timeout timer. Whenever an AIP is received, the source phy shall reinitialize and restart the Open Timeout timer. Source phys are not required to enforce a limit on the number of AIPs received before abandoning the connection request. When any connection response is received, the source phy shall reinitialize the Open Timeout timer. If the Open Timeout timer expires before a connection response is received, the source phy may assume the destination port does not exist and shall transmit BREAK to abandon the connection request.

The OPEN address frame flows through expander devices onto intermediate physical links. If an expander device on the pathway is unable to forward the connect request because none of the prospective physical links support the requested connection rate, the expander device shall return OPEN_REJECT (CONNECTION RATE NOT SUPPORTED). If the OPEN address frame reaches the destination, it shall return either OPEN_ACCEPT or OPEN_REJECT. Rate matching shall be used on any physical links in the pathway with negotiated physical link rates that are faster than the requested connection rate (see clause 0.5).

0.4.2.2 Connection responses

Table 2 lists the responses to an OPEN address frame.

Table 2 — Connection responses

Response	Description
AIP	Arbitration in progress. When an expander device is trying to open a connection to the selected destination port, it returns an AIP to the source port. The source port shall reinitialize and restart its Open Timeout timer when it receives an AIP. AIP is sent by an expander device while it is internally arbitrating for access to an expander port.
OPEN_ACCEPT	Connection request accepted. This is sent by the destination port.
OPEN_REJECT	Connection request rejected. This is sent in response by the destination port or by an expander device. The different versions are described in 7.2.4.11.
OPEN address frame	If AIP has been previously detected, this indicates an overriding connection request. If AIP has not yet been detected, this indicates two connection requests crossing on the physical link. Arbitration fairness determines which one wins (see 0.4.3).
BREAK	The destination port or expander port may reply with BREAK indicating the connection is not being established.
No response; Open Timeout timer expires	The source port shall abandon the connection request by transmitting BREAK.

After an OPEN_REJECT (CONNECTION RATE NOT SUPPORTED) has been received, a SAS target port shall set the connection rate for future requests for that I_T_L_Q nexus to:

a) the last value received in a connection request from the SAS initiator port;
b) 1,5 Gbps; or
c) the connection rate in effect when the command was received.

0.4.3 Arbitration fairness

SAS supports least-recently used arbitration fairness.

Each SAS port and expander port shall include an Arbitration Wait Time timer which counts the time from when the port makes a connection request until its request is granted. The Arbitration Wait Time timer shall count in microseconds from 0 µs to 32 767 µs and in milliseconds from 32 768 µs to 32 767 ms + 32 768 µs. The Arbitration Wait Time timer shall stop incrementing when its value reaches 32 767 ms + 32 768 µs.

SAS ports (i.e., SAS initiator ports and SAS target ports) shall start the Arbitration Wait Time timer when they transmit the first OPEN address frame (see 7.8.3) for the connection request. When the SAS port retransmits the OPEN address frame (e.g., after losing arbitration and handling an inbound OPEN address frame), it shall set the ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer.

SAS ports should set the Arbitration Wait Time timer to zero when they transmit the first OPEN address frame for the connection request. A SAS initiator port or SAS target port may be unfair by setting the ARBITRATION WAIT TIME field in the OPEN address frame (see 7.8.3) to a higher value than its Arbitration Wait Time timer indicates. However, unfair SAS ports shall not set the ARBITRATION WAIT TIME field to a value greater than or equal to 8000h; this limits the amount of unfairness and helps prevent livelocks.

The expander port that receives an OPEN address frame shall set the Arbitration Wait Time timer to the value of the incoming ARBITRATION WAIT TIME field and start the Arbitration Wait Time timer as it arbitrates for internal access to the outgoing expander port. When the expander device transmits the OPEN address frame out another expander port, it shall set the outgoing ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer maintained by the incoming expander port.

A port shall stop the Arbitration Wait Time timer and set it to zero when it wins arbitration (i.e., it receives either OPEN_ACCEPT or OPEN_REJECT from the destination SAS port). A port shall stop the Arbitration Wait Time timer and set it to zero when it loses arbitration to a connection request that satisfies its arbitration request (i.e., it receives an OPEN address frame from the destination SAS port with the INITIATOR PORT bit set to the opposite value and a matching PROTOCOL field).

When arbitrating for access to an outgoing expander port, the expander device shall select the connection request from the expander port with the largest Arbitration Wait Time timer value. If the largest Abitration Wait timer values are identical, then the connection request with the largest SOURCE SAS ADDRESS shall win arbitration.

If two connection requests pass on a physical link, the winner shall be determined by comparing OPEN address frame field values in the following order:

1) largest ARBITRATION WAIT TIME field value; and
2) largest SOURCE SAS ADDRESS field value.

See 7.8.3 for details on the OPEN address frame and the ARBITRATION WAIT TIME field.

0.4.4 Arbitration and resource management in an expander device

0.4.4.1 Arbitration overview

The ECM shall arbitrate and assign or deny path resources for connection attempts requested by each expander phy in response to receiving valid OPEN address frames.

Arbitration includes adherence to the SAS arbitration fairness algorithm and path recovery. Path recovery is used to avoid potential deadlock scenarios within the SAS topology by deterministically choosing which partial pathway(s) to tear down to allow at least one connection to complete.

The ECM responds to connection requests by returning arbitration won, lost, and reject confirmations to the requesting expander phy.

Each path request contains the Arbitration Wait Time and the Source SAS Address arguments from the received OPEN address frame.

If two path requests contend, the winner shall be determined by comparing the OPEN address frame contents using the arbitration priority described in table 3.

Table 3 — Arbitration priority

Bits 83-68 (83 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

The ECM shall generate the arbitration reject confirmation when any of the following conditions are met:

a) the connection request does not map to a valid expander phy;
b) the connection request specifies an unsupported connection rate; or
c) the connection request specifies a destination port that contains at least one partial pathway and pathway recovery rules require this connection request to release path resources.

The ECM shall generate the arbitration lost confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; and
b) the destination expander phy of this connection request has received a higher priority OPEN address frame with this expander phy as its destination (i.e., when two expander phys both receive an OPEN address frame destined for each other, the ECM shall provide arbitration lost confirmation to the expander phy that received the lowest priority OPEN address frame).

The ECM shall generate the arbitration won confirmation when all of the following conditions are met:

a) the connection request maps to an available expander phy at a supported connection rate; and
b) no higher priority connection requests are present with this expander phy as the destination.

0.4.4.2 Arbitration status

Arbitration status shall be conveyed between expander devices and by expander devices to SAS endpoints using AIP primitives. This status is used to monitor the progress of connection attempts and to facilitate pathway recovery as part of deadlock avoidance.

The arbitration status of an expander phy is set to the last type of AIP received.

0.4.4.3 Partial Pathway Timeout timer

Each expander phy shall maintain a Partial Pathway Timeout timer. This timer is used to identify potential deadlock conditions and to request resolution by the ECM. An expander phy shall initialize the Partial Pathway Timeout timer to the partial pathway timeout value (see 0.4.4.3) and run the Partial Pathway Timeout timer whenever the ECM provides confirmation to an expander phy that all expander phys within the requested destination port are blocked waiting on partial pathways.

NOTE 1 - The partial pathway timeout value allows flexibility in specifying how long an expander device waits before attempting pathway recovery. The recommended default value was chosen to cover a wide range of topologies. Selecting small partial pathway timeout value values within a large topology may compromise performance because of the time a device must wait after receiving OPEN_REJECT (PATHWAY BLOCKED) before it may retry the connection request. Similarly, selecting large partial pathway timeout value values within a small topology may compromise performance due to waiting longer than necessary to detect pathway blockage.

When the Partial Pathway Timeout timer is not running, an expander phy shall initialize and start the Partial Pathway Timeout timer when all of the following conditions are met:

a) there are no unallocated expander phys within a requested destination port available to complete the connection; and
b) at least one expander phy within the requested destination port contains a blocked partial pathway.

When one of the conditions above are not met, the expander phy shall stop the Partial Pathway Timeout timer. If the timer expires, pathway recovery shall occur (see 0.4.4.4).

0.4.4.4 Pathway recovery

Pathway recovery provides a means to abort connection requests in order to prevent deadlock using pathway recovery priority comparisons. Pathway recovery priority compares the ~~the~~ OPEN address frames of the blocked connection requests as described in table 4.

Table 4 — Pathway recovery priority

Bits 75-68 (75 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
PATHWAY BLOCKED COUNT field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

When the Partial Pathway Timeout timer for an arbitrating expander phy expires (i.e., reaches a value of zero), the ECM shall determine whether to continue the connection request or to abort the connection request.

The ECM shall instruct the arbitrating expander phy to reject the connection request by transmitting OPEN_REJECT (PATHWAY_BLOCKED) when the Partial Pathway Timeout timer expires and the pathway recovery priority of the arbitrating expander phy (i.e., the expander phy requesting the connection) is less than the pathway recovery priority of all expander phys within the destination port with an arbitration status of WAITING_ON_PARTIAL.

0.4.5 Expander devices and connection requests

0.4.5.1 All expander devices

Before an expander device transmits AIP, it may have transmitted an OPEN address frame on the same physical link. Arbitration fairness dictates which OPEN address frame wins (see 0.4.3).

After an expander device transmits an AIP, it shall not transmit an OPEN address frame unless it has higher arbitration priority than the incoming connection request.

Expander devices shall transmit no more than three consecutive AIPs without transmitting an idle dword. Expander devices shall transmit at least one AIP every 128 dwords.

Expander devices shall transmit an AIP within 128 dwords of receiving an OPEN address frame.

0.4.5.2 Edge expander devices

When an edge expander device receives a connection request, it shall compare the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys which have table routing attributes ~~(see 4.6.7.1)~~(see 4.6.7.1) and are attached to edge expander devices, it shall compare the destination SAS address to all the enabled routed SAS addresses in the expander route table.

If it finds a match in one or more phys, then the expander device shall arbitrate for access to one of the matching phys and forward the connection request.

If it does not find a match, but at least one phy has the subtractive routing attribute and is attached to an expander device (either an edge expander device or a fanout expander device), and the request did not come from that expander device, the connection request shall be forwarded to the expander device through any of the subtractive routing phys.

If it does not find a match and no subtractive routing phy is available, the edge expander device shall reply with OPEN_REJECT (NO DESTINATION).

If the destination phy is in the same expander port as the source phy, the edge expander device shall reply with OPEN_REJECT (BAD DESTINATION). If two edge expander device sets are attached (by subtractive ports), requests to non-existent devices return OPEN_REJECT (BAD DESTINATION) rather than OPEN_REJECT (NO DESTINATION). If a fanout expander device is in the SAS domain, an OPEN_REJECT (NO DESTINATION) is returned if the SAS address is not in the fanout expander device's route table or an OPEN_REJECT (BAD DESTINATION) is returned if the SAS address is in the fanout expander device's route table but the device is no longer attached to the edge expander device set to which the request is routed.

0.4.5.3 Fanout expander devices

When a fanout expander device receives a connection request, it shall compare the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys that are attached to edge expander devices, it shall compare the destination SAS addresses to all the enabled SAS addresses in the expander route table.

If it finds a match in one or more phys, it shall arbitrate for access to one of the matching phys and forward the connection request.

If it does not find a match, it shall reply with OPEN_REJECT (NO DESTINATION). If the destination phy is in the same expander port as the source phy, it shall reply with OPEN_REJECT (BAD DESTINATION).

0.4.6 Abandoning a connection request

BREAK may be used to abandon a connection request. The source port shall transmit a BREAK after the Open Timeout timer expires or if it chooses to abandon its request for any other reason.

After transmitting BREAK, the source port shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer. If the Break Timeout timer expires before a break response is received, the source port may assume the physical link is unusable.

Table 5 lists the possible responses to a BREAK sent before a connection response has been received.

Table 5 — Abandon connection responses

Response	Description
BREAK	This confirms that the connection request has been abandoned.
Open response listed in 0.4.2	The BREAK was too late and an open response arrived late. The originator shall honor this as a response to the open request it was attempting to abandon.
No response and Break Timeout timer expires	The originating port shall assume the connection request has been abandoned.

When a port sourcing a BREAK is attached to an expander device, the BREAK response to the source port is generated by the expander port to which the source port is attached, not the other SAS port in the connection. If the expander device has transmitted a connection request to the destination, it shall also transmit BREAK to the destination. If the expander device has not transmitted a connection request to the destination, it shall not transmit BREAK to the destination. The expander device shall transmit BREAK back to the originating port after it has ensured that an open response will not occur. Figure 2 shows an example of BREAK usage.

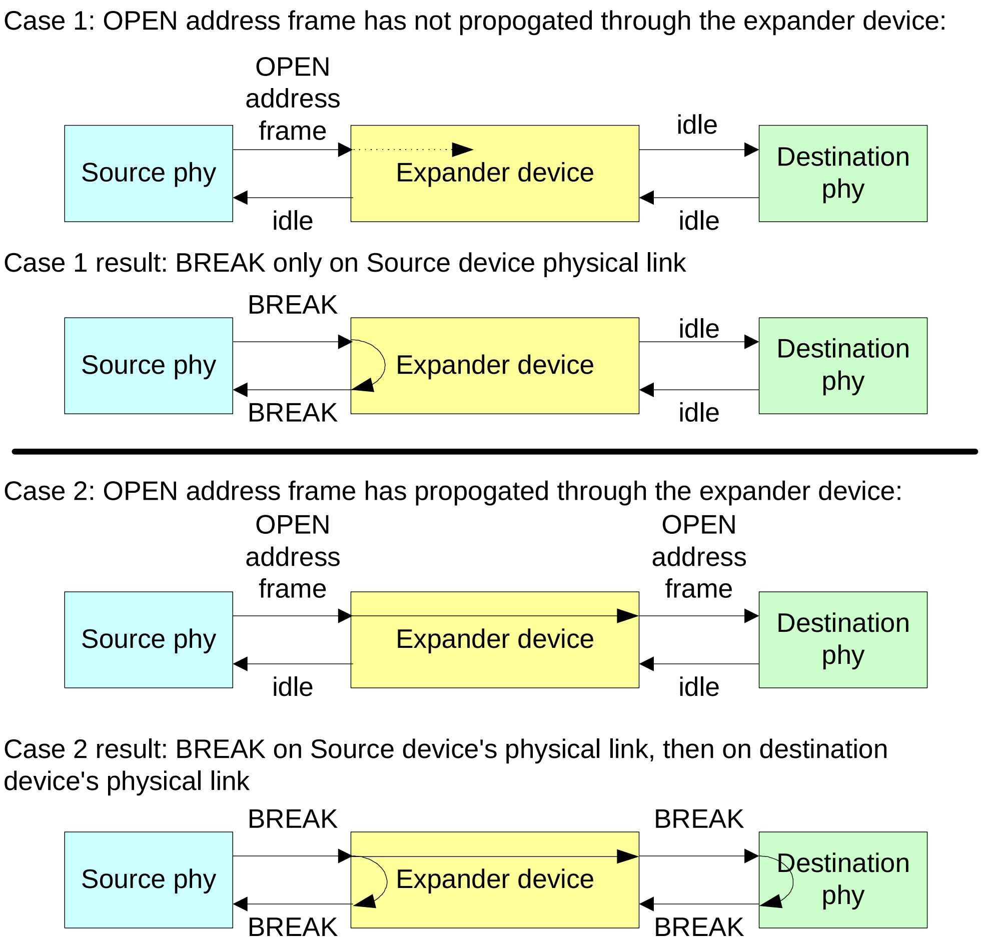

Figure 2 — BREAK usage

Figure 3 shows the sequence for a connection request where the Open Timeout timer expires.

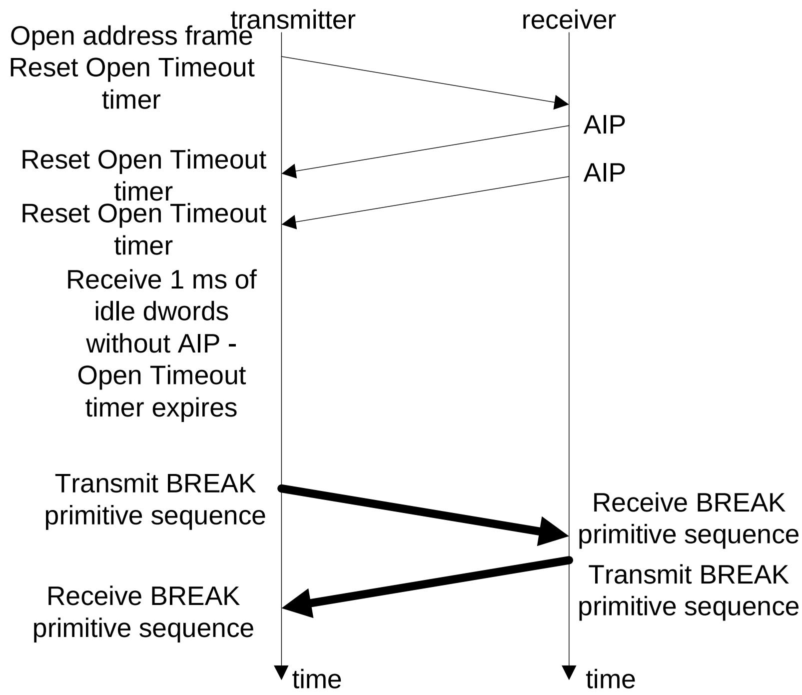

Figure 3 — Connection request timeout example

If a port detects a timeout while closing a connection (e.g., while exchanging DONEs in SSP, or while exchanging CLOSEs in any protocol) it may transmit a BREAK to break the connection.

0.4.7 Breaking a connection

A BREAK may also be used to break a connection, in cases where CLOSE is not available. After transmitting BREAK, the originating port shall ignore all incoming dwords except for BREAKs.

Table 6 lists the possible responses to a BREAK sent after a connection has been established.

Table 6 — Break connection responses

Response	Description
BREAK	This confirms that the connection has been broken.
No response and Break Timeout timer expires	The originating port shall assume the connection has been broken.

In addition to a BREAK, a connection is considered broken due to loss of dword synchronization ~~(see 6.8)~~(see 6.8).

In addition to the actions described in this subclause and in 0.4.6, the following shall be the responses by an SSP phy to a broken connection:

a) Received frames having no CRC error may be considered valid regardless of whether an ACK has been transmitted in response to the frame prior to the broken connection;
b) Transmitted frames for which an ACK has been received prior to a broken connection shall be considered successfully transmitted; and
c) Transmitted frames for which an ACK or NAK has not been received prior to a broken connection shall be considered not successfully transmitted.

0.4.8 Closing a connection

CLOSE is used to close a connection of any protocol. See 0.8.6 for details on closing SSP connections, 0.9.5 for details on closing STP connections, and 0.10.3 for details on closing SMP connections.

No additional dwords for the connection shall follow the CLOSE. Expander devices shall close the full-duplex connection upon detecting a CLOSE in each direction.

When a port has both transmitted and received CLOSE, it shall consider the connection closed.

Figure 4 shows example sequences for closing a connection.

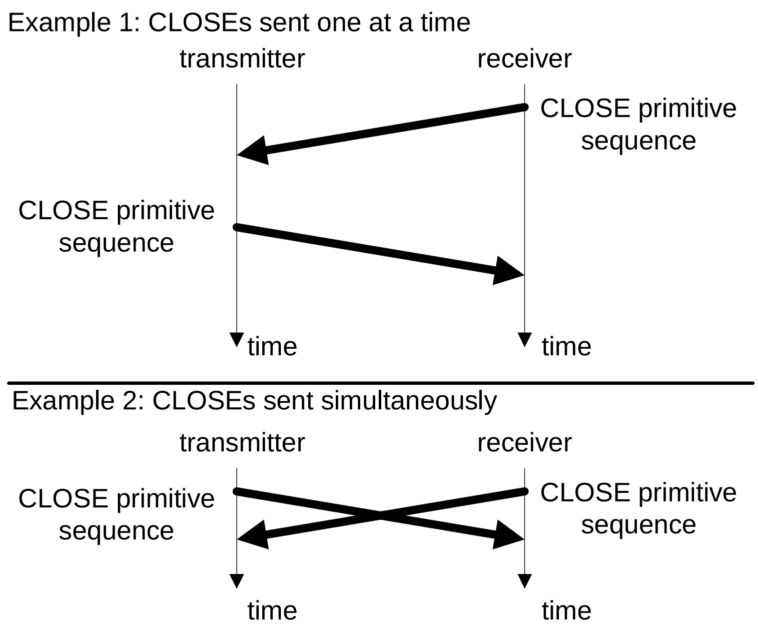

Figure 4 — Closing a connection example

0.5 Rate matching

Each successful connection request contains the connection rate ~~(see 4.1.10)~~(see 4.1.10) of the pathway.

Every phy in the physical link shall insert ALIGNs or NOTIFYs between dwords to match the connection rate. Phys receiving ALIGNs and NOTIFYs delete them irregardless of whether the ALIGNs and NOTIFYs were inserted for clock skew management (see 7.3) or for rate matching.

The faster phy shall rotate between ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) to reduce EMI. NOTIFYs may be used to replace ALIGNs (see 7.2.4.9).

Figure 5 shows an example of rate matching between a 3,0 Gbps source phy and a 3,0 Gbps destination phy, with an intermediate 1,5 Gbps physical link in between.

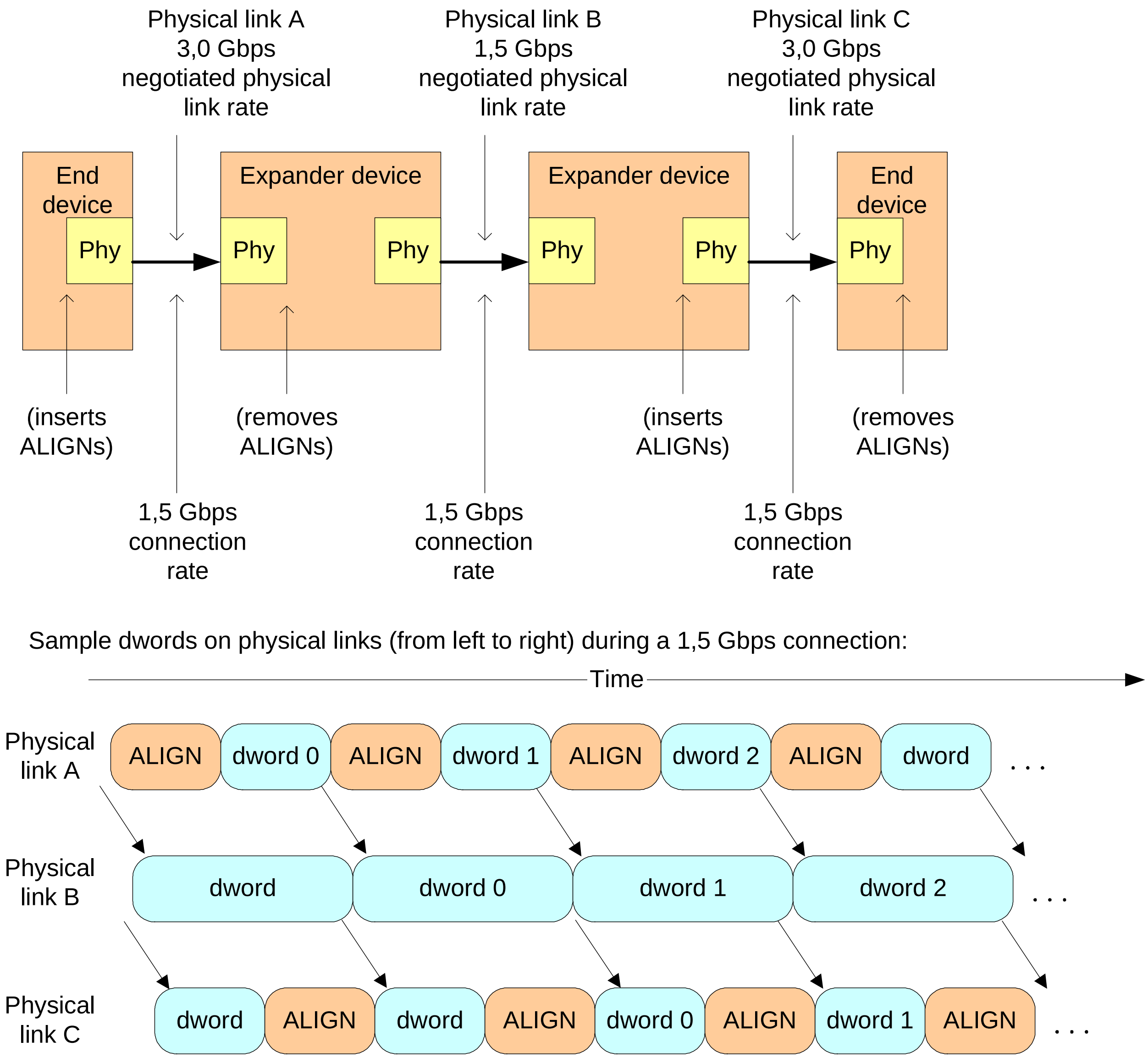

Figure 5 — Rate matching example

The source phy shall start inserting ALIGNs and NOTIFYs for rate matching at the selected connection rate with the first dword following the EOAF for the OPEN address frame. The source phy transmits idle dwords including ALIGNs and NOTIFYs at the selected connection rate while waiting for the connection response. This enables each expander device to start forwarding dwords from the source phy to the destination phy after forwarding an OPEN_ACCEPT.

A phy shall stop inserting ALIGNs and NOTIFYs for rate matching after transmitting the first dword in a CLOSE.

If an STP initiator port discovers a SATA device behind an STP/SATA bridge with a physical link rate greater than the maximum connection rate supported by the pathway from the STP initiator port, the STP initiator port should use the SMP PHY CONTROL function ~~(see 10.4.3.9)~~(see 10.4.3.9) to set the MAXIMUM PHYSICAL LINK RATE field of the expander phy attached to the SATA device to the maximum connection rate supported by the pathway.

0.6 SL (link layer for SAS phys) state ~~machine~~machines

0.6.1 SL state ~~machine~~ machines overview

~~The SL (link layer for SAS phys) state machine controls connections, handling both connection requests (OPEN address frames), CLOSEs, and BREAKs. The state machine consists of the following states:~~

a) ~~SL0:Idle (see 0.6.3.2)(initial state);~~
b) ~~SL1:ArbSel (see 0.6.3.3);~~
c) ~~SL2:Selected (see 0.6.3.4);~~
d) ~~SL3:Connected (see 0.6.3.5);~~
e) ~~SL4:DisconnectWait (see 0.6.3.6);~~
f) ~~SL5:BreakWait (see 0.6.3.7); and~~
g) ~~SL6:Break (see 0.6.3.8).~~

The SL (link layer for SAS phys) contains state machines that run in parallel. Those state machines control SSP, SMP, and STP connections. The SL state machines are as follows:

a) SL_CC (connection control) state machine (see 0.6.3); and
b) SL_RA (receive OPEN address frame) state machine (see 0.6.4);

~~The state machine shall start in~~ All the ~~SL0:Idle state. The~~ SL state ~~machine~~ machines shall ~~transition to the SL0:Idle state from any other state~~ begin after receiving an Enable Disable SAS Link (~~Disable~~Enable) message from the SL_IR state machines (see 0.1.5).

~~The SL state machine receives the following messages from the SSP link layer state machine (see 0.8.7), the STP link layer state machine, and SMP link layer state machine (see 0.10.4):~~

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

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

a) ~~Enable Disable SSP;~~
b) ~~Enable Disable STP; and~~
c) ~~Enable Disable SMP.~~

~~The SL state machine receives the following messages from the SL_IR state machines (see 0.1.5):~~

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

~~Unless otherwise stated within the state description, all running disparity errors, illegal characters, and unexpected primitives (i.e., any primitive not described in the description of the SL state) received within any SL state shall be ignored and idle dwords shall be transmitted.~~

~~Any detection of an internal error shall cause the SL state machine to transition to the SL5:BreakWait state.~~

~~The SL state machine shall maintain the timers listed in table 8.~~

Table 7 — SL timers

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

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

Figure 6 shows the SL state machine.

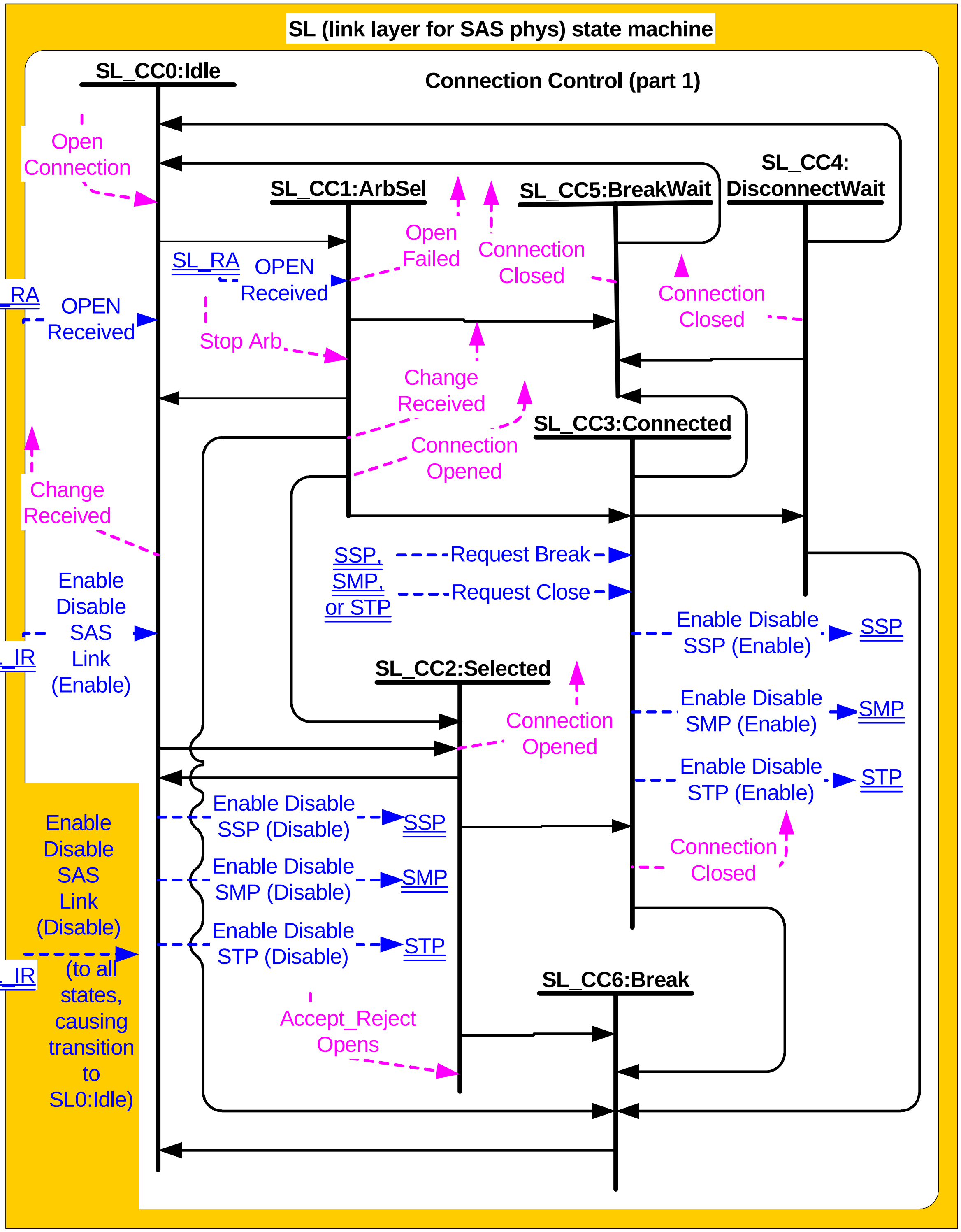

Figure 6 — SL (link layer for SAS phys) state machine (part 1)

Figure 7 shows the messages sent to the SL transmitter and received from the SL receiver.

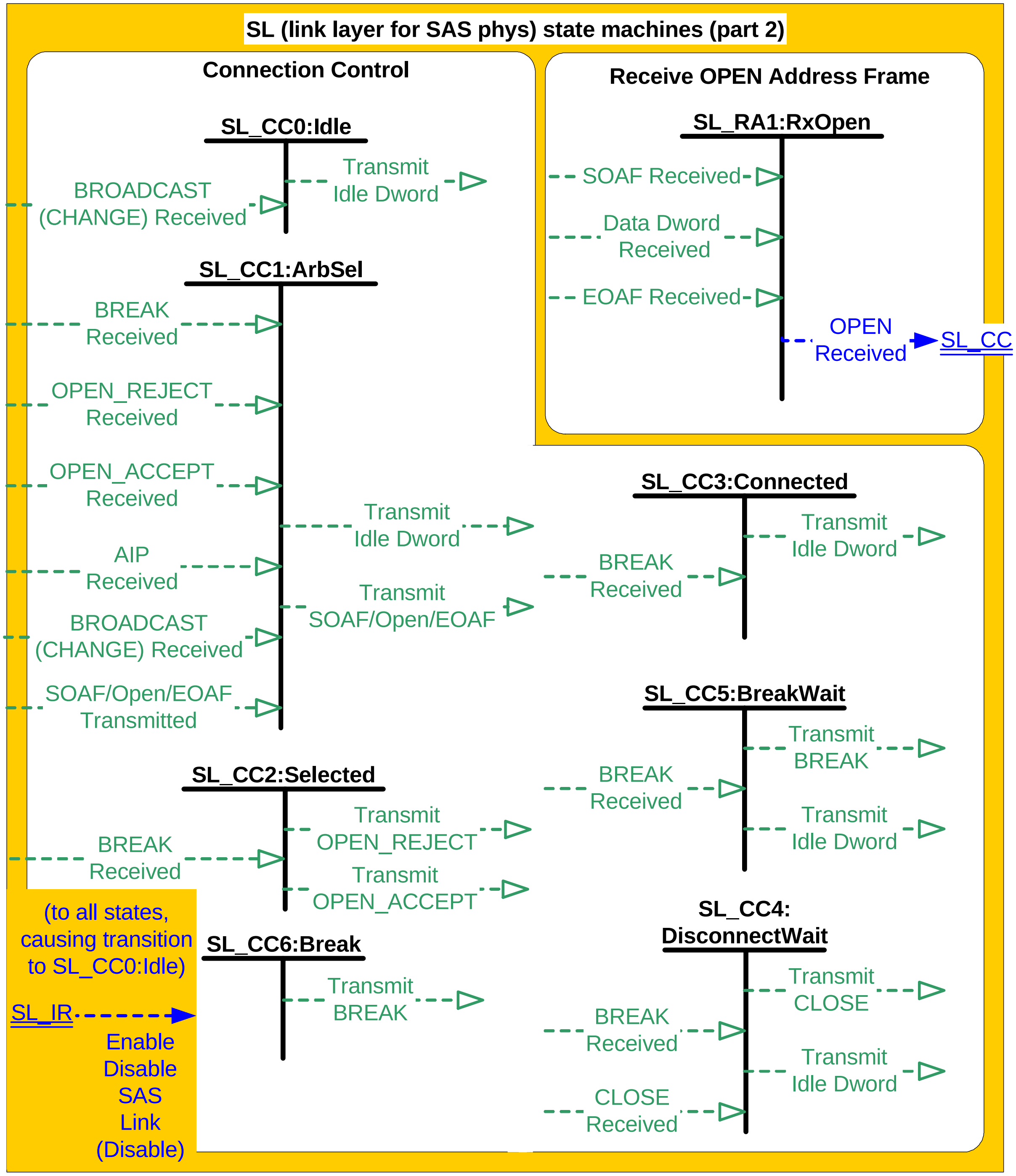

Figure 7 — SL (link layer for SAS phys) state machine (part 2)

0.6.2 SL transmitter and receiver

The SL transmitter receives the following messages from the SL state ~~machine~~machines:

~~a) Transmit Idle Dword;~~
~~b) Transmit SOAF;~~
a) Transmit ~~Data~~ Idle Dword;
b) Transmit SOAF/Data Dword/EOAF;
c) Transmit OPEN_ACCEPT;

d) Transmit OPEN_REJECT with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (Retry));
e) Transmit BREAK; and
f) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal)).

The SL transmitter sends the following messages from the SL state machines:

a) SOAF/Data Dword/EOAF Transmitted;

The SL receiver sends the following messages to the SL state ~~machine~~machines:

a) SOAF Received;
b) Data Dword Received;
c) EOAF Received;
d) BROADCAST Received with an argument indicating the specific type (e.g., BROADCAST Received (Change));
e) BREAK Received;
f) OPEN_ACCEPT Received;
g) OPEN_REJECT Received with an argument indicating the specific type (e.g., OPEN_REJECT Received (No Destination));
h) AIP Received; and
i) CLOSE Received.

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

1) BREAK;
2) CLOSE;
3) OPEN_ACCEPT or OPEN_REJECT;
4) SOAF or data dword or EOAF; then
5) idle dword.

~~When the SL transmitter has been requested to transmit a OPEN address frame (i.e., SOAF, the data dwords of the OPEN address frame, and an EOAF) and a primitive is requested to be transmitted while the frame is being transmitted, the SL transmitter shall transmit the indicated primitive by inserting the primitive between the frames' dwords.~~

0.6.3 SL_CC (connection control) state machine

0.6.3.1 SL_CC state machine overview

This state machine controls connections, handling both connection requests (OPEN address frames), CLOSEs, and BREAKs.

The state machine consists of the following states:

a) SL_CC0:Idle (see 0.6.3.2)(initial state);
b) SL_CC1:ArbSel (see 0.6.3.3);
c) SL_CC2:Selected (see 0.6.3.4);
d) SL_CC3:Connected (see 0.6.3.5);
e) SL_CC4:DisconnectWait (see 0.6.3.6);
f) SL_CC5:BreakWait (see 0.6.3.7); and
g) SL_CC6:Break (see 0.6.3.8).

The state machine shall start in the SL_CC0:Idle state. The state machine shall transition to the SL_CC0:Idle state from any other state after receiving an Enable Disable SAS Link (Disable) message from the SL_IR state machines (see 0.1.5).

The SL_CC state machine receives the following messages from the SSP link layer state machine (see 0.8.7), the STP link layer state machine, and SMP link layer state machine (see 0.10.4):

a) Request Break; and

b) Request Close.

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

a) Enable Disable SSP;
b) Enable Disable STP; and
c) Enable Disable SMP.

The SL_CC state machine receives the following messages from the SL_IR state machines (see 0.1.5):

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

Unless otherwise stated within the state description, all running disparity errors, illegal characters, and unexpected primitives (i.e., any primitive not described in the description of the SL_CC state) received within any SL_CC state shall be ignored and idle dwords shall be transmitted.

Any detection of an internal error shall cause the SL_CC state machine to transition to the SL_CC5:BreakWait state.

The SL_CC state machine shall maintain the timers listed in table 8.

Table 8 — SL_CC timers

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

0.6.3.2 ~~SL0~~SL_CC0:Idle state

0.6.3.2.1 State description

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

Upon entry to this state, this state shall send Enable Disable SSP (Disable), Enable Disable SMP (Disable), and Enable Disable STP (Disable) messages to the SSP, SMP, and STP link layer state machines.

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

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

~~An address frame (i.e., all the data dwords between an SOAF and EOAF) shall be discarded if any of the following conditions are true:~~

~~a) the number of data dwords between the SOAF and EOAF is not 8 data dwords; or~~
~~b) there is a CRC error.~~

~~If consecutive SOAF Received messages are received without an intervening EOAF Received message (i.e., SOAF, data dwords, SOAF, data dwords, and EOAF instead of SOAF, data dwords, EOAF, SOAF, data dwords, and EOAF) then this state shall discard all data dwords between those SOAFs.~~

~~If the frame is discarded then no further action is taken by this state.~~

0.6.3.2.2 Transition ~~SL0~~SL_CC0:Idle to ~~SL1~~SL_CC1:ArbSel

This transition shall occur after receiving both an Enable Disable SAS Link (Enable) confirmation and an Open Connection request. The Open Connection request includes these arguments:

a) source SAS address;

b) destination SAS address;
c) protocol;
d) arbitration wait time;
e) connection rate;
f) initiator port bit; and
g) initiator connection tag.

0.6.3.2.3 Transition ~~SL0~~SL_CC0:Idle to ~~SL2~~SL_CC2:Selected

This transition shall occur after receiving both an Enable Disable SAS Link (Enable) confirmation and ~~a valid~~ an OPEN ~~address frame~~Received message.

~~A valid OPEN address frame is 8 data dwords long and has no CRC error (see 7.8.3).~~

0.6.3.3 ~~SL1~~SL_CC1:ArbSel state

0.6.3.3.1 State description

This state is used to make a connection request.

This state shall:

1) request an OPEN address frame be transmitted by sending ~~an Transmit SOAF message, multiple Transmit Data Dword messages, and~~ a Transmit SOAF/Open/EOAF message to the SL transmitter with the dwords containing the OPEN address frame with its fields set to the arguments received with the Open Connection request;
2) initialize and start the Open Timeout timer; and
3) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter.

~~This state shall ignore incoming OPEN_REJECTs or OPEN_ACCEPTs until after the OPEN address frame has been transmitted.~~

This state shall ignore incoming OPEN_REJECTs or OPEN_ACCEPTs from the time a Transmit SOAF/Open/EOAF message is sent to the SL transmitter until an SOAF/Open/EOAF Transmitted message is received from the SL transmitter.

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

If an AIP Received message is received after requesting the OPEN address frame be transmitted, this state shall reinitialize and restart the Open Timeout timer. The state machine shall not enforce a limit on the number of AIPs received.

~~An address frame (i.e., all the data dwords between an SOAF and EOAF) shall be discarded if any of the following conditions are true:~~

~~a) the number of data dwords between the SOAF and EOAF is not 8 data dwords; or~~
~~b) there is a CRC error.~~

~~If consecutive SOAF Received messages are received without an intervening EOAF Received message (i.e., SOAF, data dwords, SOAF, data dwords, and EOAF instead of SOAF, data dwords, EOAF, SOAF, data dwords, and EOAF) then this state shall discard all data dwords between those SOAFs.~~

~~If the frame is discarded then no further action is taken by this state relating to the invalid address frame.~~

If this state receives an OPEN_REJECT Received (No Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (No Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Bad Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (Bad Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Wrong Destination) message after transmitting the OPEN address frame, this state shall send an Open Failed (Wrong Destination) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Connection Rate Not Supported) message after transmitting the OPEN address frame, this state shall send an Open Failed (Connection Rate Not Supported) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Protocol Not Supported) message after transmitting the OPEN address frame, this state shall send an Open Failed (Protocol Not Supported) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Retry) message after transmitting the OPEN address frame, this state shall send an Open Failed (Retry) confirmation to the port layer.

If this state receives an OPEN_REJECT Received (Pathway Blocked) message after transmitting the OPEN address frame, this state shall send an Open Failed (Pathway Blocked) confirmation to the port layer.

0.6.3.3.2 Transition ~~SL1~~SL_CC1:ArbSel to ~~SL0~~SL_CC0:Idle

This transition shall occur after sending an Open Failed confirmation.

0.6.3.3.3 Transition ~~SL1~~SL_CC1:ArbSel to ~~SL2~~SL_CC2:Selected

This transition shall occur after transmitting the OPEN address frame if:

a) one or more AIP Received messages have been received before ~~a valid~~ an OPEN ~~address frame~~ received message is received (i.e., the incoming OPEN address frame overrides the outgoing OPEN address frame); or
b) no AIP Received messages have been received before ~~a valid~~ an OPEN ~~address frame~~ received message is received, and the arbitration fairness rules (see 0.4.3) indicate the received OPEN address frame overrides the outgoing OPEN address frame.

The arbitration fairness comparison shall use the value of the arbitration wait time argument to the Open Connection request for the outgoing OPEN address frame and the value of the ARBITRATION WAIT TIME field received in the incoming OPEN address frame.

~~A valid OPEN address frame is 8 data dwords long and has no CRC error.~~

0.6.3.3.4 Transition ~~SL1~~SL_CC1:ArbSel to ~~SL3~~SL_CC3:Connected

This transition shall occur this state receives an OPEN_ACCEPT Received message after transmitting the OPEN address frame.

If the PROTOCOL field in the transmitted OPEN address frame was set to STP, then this state shall send a Connection Opened (STP, Source Opened) confirmation to the port layer before the transition.This transition shall include an Open STP Connection argument. At this point an STP connection has been opened between the source phy and the destination phy.

If the PROTOCOL field in the transmitted OPEN address frame was set to SSP, then this state shall send a Connection Opened (SSP, Source Opened) confirmation to the port layer before the transition. This transition shall include an Open SSP Connection argument, Source Port Hashed Value argument (i.e., hashed value of the source port identifier), and the Destination Port Hashed Value argument (i.e., hashed value of the source destination identifier). At this point an SSP connection has been opened between the source phy and the destination phy.

If the PROTOCOL field in the transmitted OPEN address frame was set to SMP, then this state shall send a Connection Opened (SMP, Source Opened) confirmation to the port layer before the transition. This transition shall include an Open SMP Connection argument, a Source Port Hashed Value argument (i.e., hashed value of the source port identifier), and a Destination Port Hashed Value argument (i.e., hashed value of the source destination identifier). At this point an SMP connection has been opened between the source phy and the destination phy.

0.6.3.3.5 Transition ~~SL1~~SL_CC1:ArbSel to ~~SL5~~SL_CC5:BreakWait

This transition shall occur if a BREAK Received message has not been received and after:

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

0.6.3.3.6 Transition ~~SL1~~SL_CC1:ArbSel to ~~SL6~~SL_CC6:Break

This transition shall occur after:

a) receiving a BREAK Received message; and
b) sending an Open Failed (Break Received) confirmation to the port layer.

0.6.3.4 ~~SL2~~SL_CC2:Selected state

0.6.3.4.1 State description

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

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

1) If the OPEN address frame DESTINATION SAS ADDRESS field does not match the SAS address of this device, this state shall send a Transmit OPEN_REJECT (Wrong Destination) message to the SL transmitter;
2) If the OPEN address frame PROTOCOL field is set to a protocol that is not supported, this ~~this~~ state shall send a Transmit OPEN_REJECT (Protocol Not Supported) message the SL transmitter;
3) If the OPEN address frame CONNECTION RATE field is set to a connection rate that is not supported, this ~~this~~ state shall send a Transmit OPEN_REJECT (Connection Rate Not Supported) message the SL transmitter;
4) If the OPEN address frame PROTOCOL field is set to STP, the source SAS address is not that of the STP initiator port with an affiliation established or the source SAS address is not that of an STP initiator port with a task file register set resources (see 0.9.3), this state shall send a Transmit OPEN_REJECT (STP Resources Busy) message the SL transmitter;
5) If an Accept_Reject Opens (Reject SSP) request, Accept_Reject Opens (Reject SMP) request, or Accept_Reject Opens (Reject STP) request is received and the requested protocol is the corresponding protocol, this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter;
6) If the requested protocol is SSP and this state has not received an Accept_Reject Opens (Reject SSP) request then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SSP, Destination Opened) confirmation to the port layer;
7) If the requested protocol is SMP and this state has not received an Accept_Reject Opens (Reject SMP) request then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SMP, Destination Opened) confirmation to the port layer; or
8) If the requested protocol is STP and this state has not received an Accept_Reject Opens (Reject STP) request then this ~~this~~ state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (STP, Destination Opened) confirmation to the port layer.

0.6.3.4.2 Transition ~~SL2~~SL_CC2:Selected to ~~SL0~~SL_CC0:Idle

This transition shall occur after this state sends a Transmit OPEN_REJECT message to the SL transmitter.

0.6.3.4.3 Transition ~~SL2~~SL_CC2:Selected to ~~SL3~~SL_CC3:Connected

This transition shall occur after sending a Connection Opened confirmation.

This transition shall include an Open SSP Connection, Open STP Connection, or Open SMP Connection argument based on the requested protocol.

0.6.3.4.4 Transition ~~SL2~~SL_CC2:Selected to ~~SL6~~SL_CC6:Break

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

0.6.3.5 ~~SL3~~SL_CC3:Connected state

0.6.3.5.1 State description

This state enables the SSP, STP, or SMP link layer state machine to transmit dwords during a connection.

If this state is entered from ~~SL1~~SL_CC1:ArbSel state or the ~~SL2~~SL_CC2:Selected state with an argument of Open SMP Connection then this state shall send an Enable Disable SMP (Enable) message to the SMP link layer state machines (see 0.10.4).

If this state is entered from ~~SL1~~SL_CC1:ArbSel state or the ~~SL2~~SL_CC2:Selected state with an argument of Open SSP Connection then this state shall send an Enable Disable SSP (Enable) message to the SSP link layer state machines (see 0.8.7).

If this state is entered from ~~SL1~~SL_CC1:ArbSel state or the ~~SL2~~SL_CC2:Selected state with an argument of Open STP Connection then this state shall send an Enable Disable STP (Enable) message to the STP link layer state machines (see 0.9.7).

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SL transmitter until the SSP, SMP, or STP link layer state machine starts transmitting.

0.6.3.5.2 Transition ~~SL3~~SL_CC3:Connected to ~~SL4~~SL_CC4:DisconnectWait

This transition shall occur if a Request Close message is received.

0.6.3.5.3 Transition ~~SL3~~SL_CC3:Connected to ~~SL5~~SL_CC5:BreakWait

This transition shall occur if a Request Break message is received and a BREAK Received message has not been received.

0.6.3.5.4 Transition ~~SL3~~SL_CC3:Connected to ~~SL6~~SL_CC6:Break

This transition shall occur if a BREAK Received message is received and after this state has sent a Connection Closed (Break Received) confirmation to the port layer.

0.6.3.6 ~~SL4~~SL_CC4:DisconnectWait state

0.6.3.6.1 State description

This state closes the connection and releases all resources associated with the connection.

This state shall:

1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter;
2) send at least three Transmit Idle Dword messages to the SL transmitter; and
3) initialize and start the Close Timeout timer.

A CLOSE Received message may be received at any time while in this state.

0.6.3.6.2 Transition ~~SL4~~SL_CC4:DisconnectWait to ~~SL0~~SL_CC0:Idle

This transition shall occur after:

a) sending a Transmit CLOSE message to the SL transmitter;
b) receiving a CLOSE Received message; and
c) sending a Connection Closed (Normal) confirmation to the port layer.

0.6.3.6.3 Transition ~~SL4~~SL_CC4:DisconnectWait to ~~SL5~~SL_CC5:BreakWait

This transition shall occur if:

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

0.6.3.6.4 Transition ~~SL4~~SL_CC4:DisconnectWait to ~~SL6~~SL_CC6:Break

This transition shall occur after receiving a BREAK Received message and after sending a Connection Closed (Break Received) confirmation to the port layer.

0.6.3.7 ~~SL5~~SL_CC5:BreakWait state

0.6.3.7.1 State description

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

This state shall:

1) send a Transmit BREAK message to the SL transmitter;
2) send at least six Transmit Idle Dword messages to the SL transmitter; and
3) initialize and start the Break Timeout timer.

0.6.3.7.2 Transition ~~SL5~~SL_CC5:BreakWait to ~~SL0~~SL_CC0:Idle

This transition shall occur after receiving a BREAK Received message or if the Break Timeout timer expires. If a BREAK Received message is not received before the Break Timeout timer expires, this state shall send a Connection Closed (Break Timeout) confirmation to the port layer before making this transition.

0.6.3.8 ~~SL6~~SL_CC6:Break state

0.6.3.8.1 State description

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

This state shall send a Transmit BREAK message to the SL transmitter.

While in this state all primitives received shall be ignored.

0.6.3.8.2 Transition ~~SL6~~SL_CC6:Break to ~~SL0~~SL_CC0:Idle

This transition shall occur after sending a Transmit BREAK message to the SL transmitter.

0.6.4 SL_RA (receive OPEN address frame) state machine

The SL_RA state machine's function is to receive address frames and determine if the received address frame is an OPEN address frame and whether or not it was received successfully. This state machine consists of one state.

This state receives SOAFs, dwords of an OPEN address frames, and EOAFs.

This state shall ignore all primitives except SOAF and EOAF.

If this state receives a subsequent SOAF after receiving an SOAF but before receiving an EOAF, then this state shall discard the data dwords received before the subsequent SOAF.

If this state receives more than eight data dwords after an SOAF and before an EOAF, then this state shall discard the address frame.

After receiving an EOAF, this state shall check if the address frame is valid OPEN address frame.

This state shall accept an address frame a valid OPEN address frame if:

a) the ADDRESS FRAME TYPE field is set to Open;
b) the number of data dwords between the SOAF and EOAF is 8; and
c) the CRC field contains a valid CRC.

Otherwise, this state shall discard the address frame. If the frame is not discarded then this state machine shall send a OPEN Received message to the SL_CC0:Idle state and the SL_CC1:ArbSel state with an argument that contains all the data dwords received in the OPEN address frame.

0.7 XL (link layer for expander phys) state machine

0.7.1 XL state machine overview

The XL state machine controls the flow of dwords on the physical link and establishes and maintains connections with another XL state machine as facilitated by the expander function - specifically the ECM and ECR.

This state machine consists of the following states:

a) XL0:Idle (see 0.7.3)(initial state);
b) XL1:Request_Path (see 0.7.4);
c) XL2:Request_Open (see 0.7.5);
d) XL3:Open_Confirm_Wait (see 0.7.6);
e) XL4:Open_Reject (see 0.7.7);
f) XL5:Forward_Open (see 0.7.8);
g) XL6:Open_Response_Wait (see 0.7.9);
h) XL7:Connected (see 0.7.10);
i) XL8:Close_Wait(see 0.7.11);
j) XL9:Break (see 0.7.12); and
k) XL10:Break_Wait (see 0.7.13).

The XL state machine shall start in the XL0:Idle state. The XL state machine shall transition to the XL0:Idle state from any other state after receiving an Enable Disable SAS Link (Disable) message from the SL_IR state machines (see 0.1.5).

The XL state machine receives the following messages from the SL_IR state machine:

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

Unless otherwise stated within a state description, all running disparity errors, illegal characters, and unexpected primitives received within any XL state shall be ignored.

The XL state machine shall maintain the timers listed in table 8.

Table 9 — XL timers

Timer	Initial value
Partial Pathway Timeout timer	partial pathway timeout value (see 0.4.4.3)
Break Timeout timer	1 ms

Figure 8 shows several states in the XL state machine.

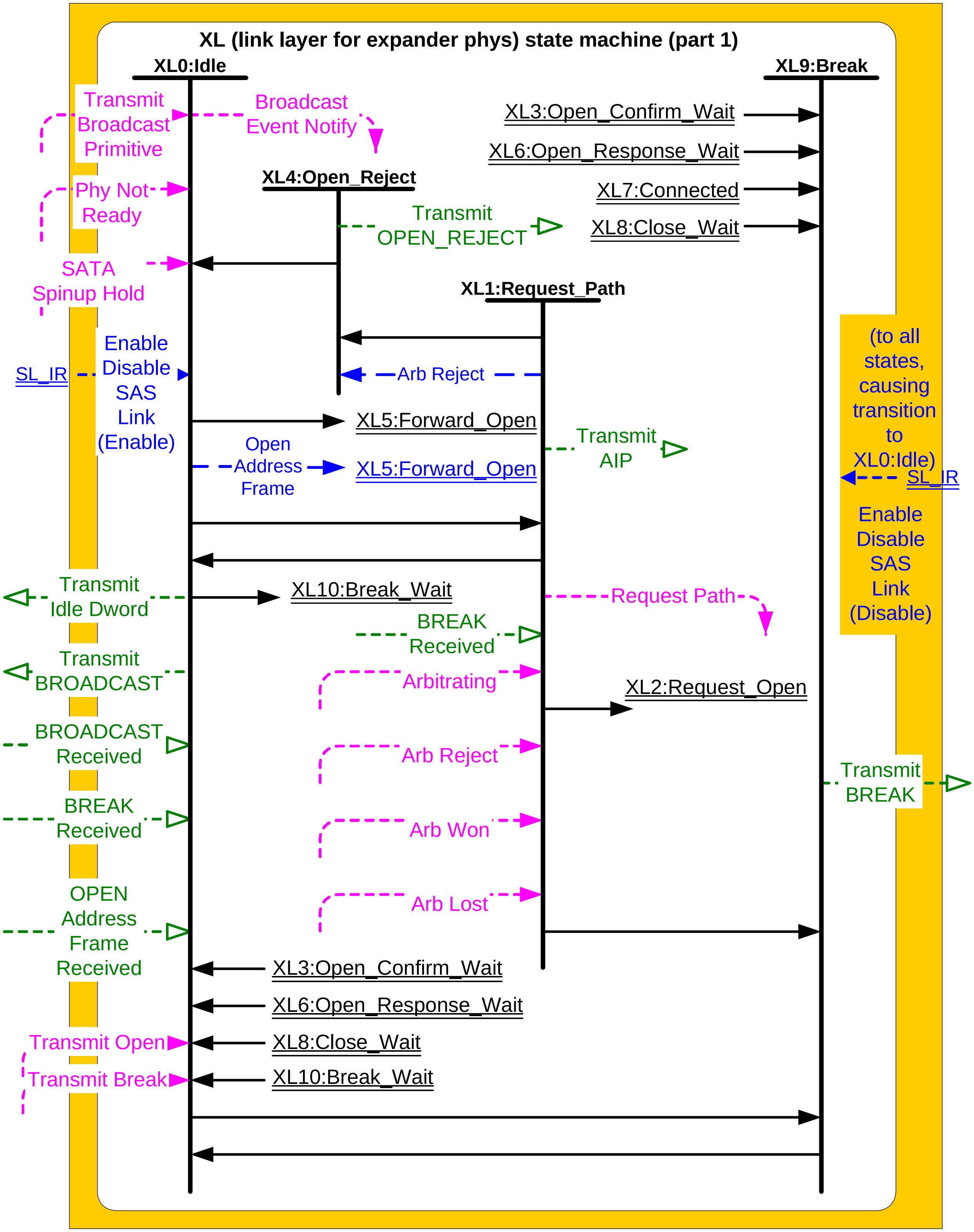

Figure 8 — XL (link layer for expander phys) state machine (part 1)

Figure 9 shows additional states in the XL state machine.

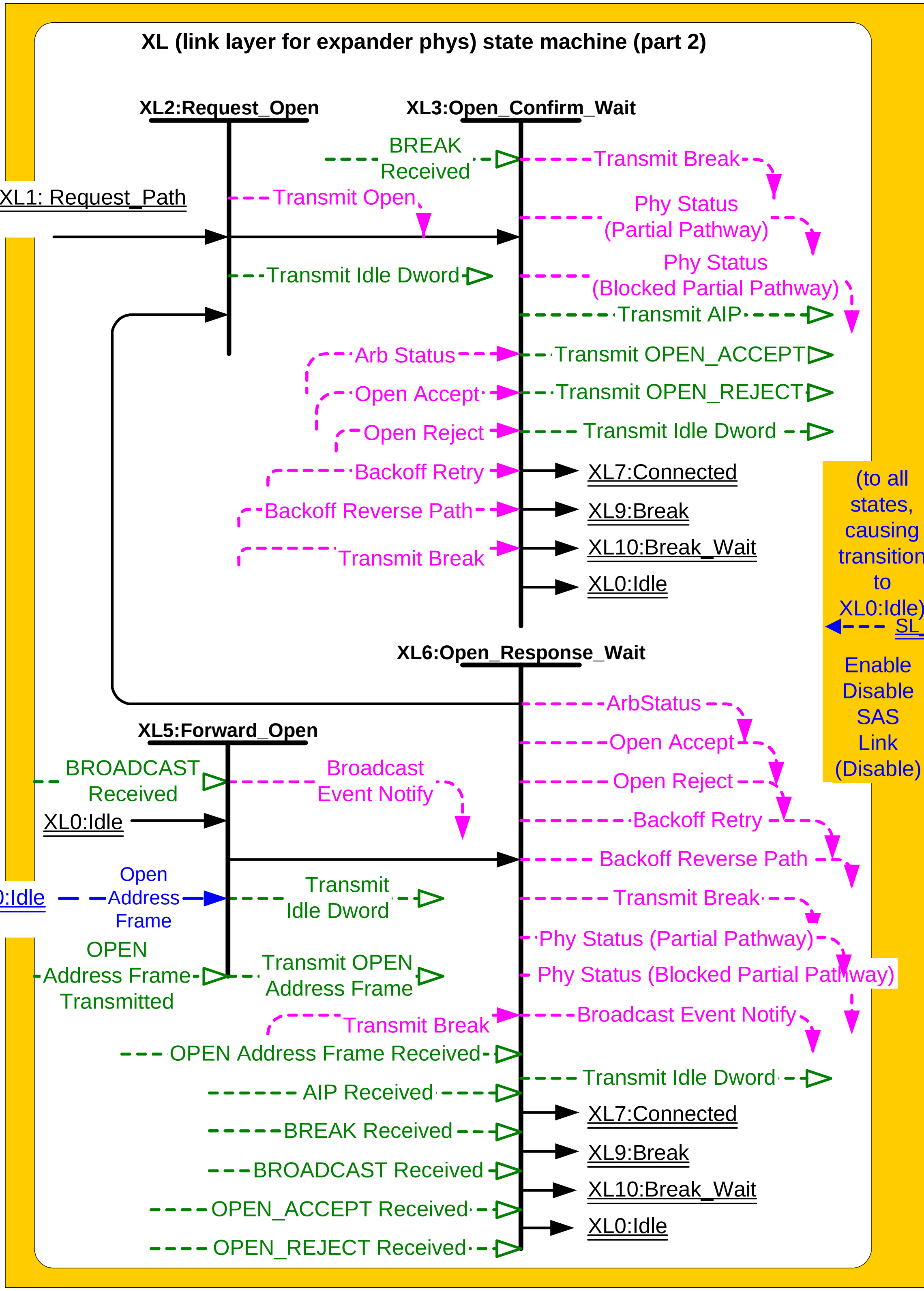

Figure 9 — XL (link layer for expander phys) state machine (part 2)

Figure 10 shows additional states in the XL state machine.

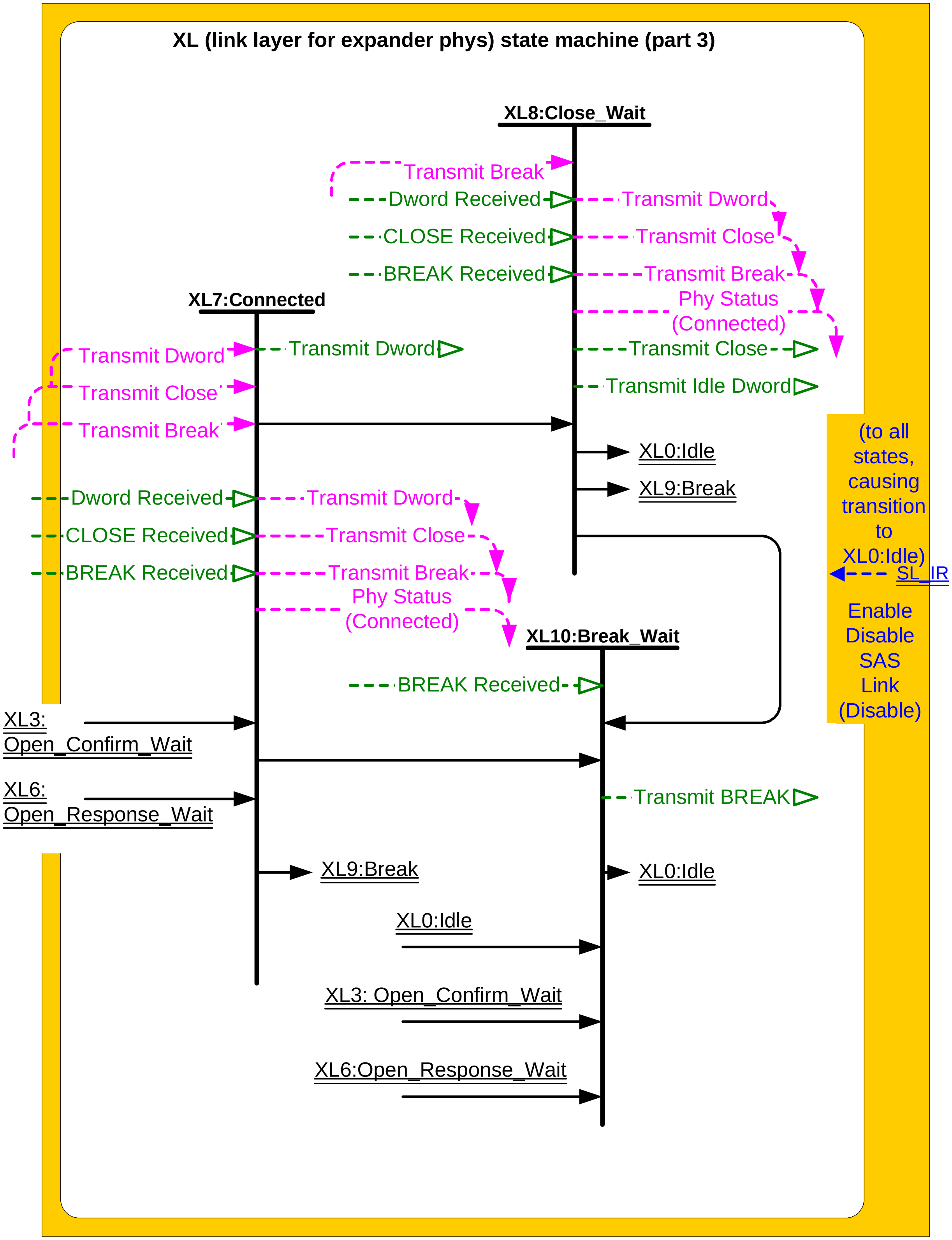

Figure 10 — XL (link layer for expander phys) state machine (part 3)

0.7.2 XL transmitter and receiver

The XL transmitter receives the following messages from the XL state machine:

a) Transmit Idle Dword;
b) Transmit AIP with an argument indicating the specific type (e.g., Transmit AIP (Normal));
c) Transmit BREAK
d) Transmit BROADCAST with an argument indicating the specific type (e.g., Transmit BROADCAST (Change));
e) Transmit CLOSE with an argument indicating the specific type (e.g., Transmit CLOSE (Normal));
f) Transmit OPEN_ACCEPT;
g) Transmit OPEN_REJECT, with an argument indicating the specific type (e.g., Transmit OPEN_REJECT (No Destination));
h) Transmit OPEN Address Frame; and
i) Transmit Dword.

The XL receiver sends the following messages to the XL state machine:

a) OPEN Address Frame Transmitted.

The XL transmitter shall insert ALIGNs and NOTIFYs needed for rate matching (see 0.5).

The XL receiver sends the following messages to the XL state machine:

a) AIP Received with an argument indicating the specific type (e.g., AIP Received (Normal));
b) BREAK Received;
c) BROADCAST Received;
d) CLOSE Received;
e) OPEN Address Frame Received; and
f) Dword Received.

0.7.3 XL0:Idle state

0.7.3.1 State description

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

If a Phy Not Ready confirmation is received, this state shall send a Broadcast Event Notify (Phy Not Ready) request to the BPP.

If a SATA Spinup Hold confirmation is received, this state shall send a Broadcast Event Notify (SATA Spinup Hold) request to the BPP.

If a BROADCAST Received message is received, this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive received (e.g., CHANGE Received).

If a Transmit Broadcast indication is received, this state shall send a Transmit BROADCAST message to the XL transmitter with an argument specifying the specific type from the Transmit Broadcast indication. Otherwise, this state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

0.7.3.2 Transition XL0:Idle to XL1:Request_Path

This transition shall occur when the following conditions are met:

a) an Enable Disable SAS Link (Enable) message has been received;
b) a Transmit Open indication is not being received;
c) a Transmit Break indication is not being received; and
d) an OPEN Address Frame Received message is received.

0.7.3.3 Transition XL0:Idle to XL5:Forward_Open

This transition shall occur when the following conditions are met:

a) an Enable Disable SAS Link (Enable) message has been received;

b) a Transmit Break indication is not being received; and
c) a Transmit Open indication is received.

This transition shall include a set of arguments containing the arguments received in the Transmit Open indication.

0.7.3.4 Transition XL0:Idle to XL9:Break

This transition shall occur after receiving a BREAK Received message if an Enable Disable SAS Link (Enable) message has been received.

0.7.3.5 Transition XL0:Idle to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication if an Enable Disable SAS Link (Enable) message has been received.

0.7.4 XL1:Request_Path state

0.7.4.1 State description

This state is used to arbitrate for connection resources and to specify the destination of the connection.

Upon entry to this state, this state shall repeatedly send a Transmit AIP (Normal) parameter to the XL transmitter.

If an Arbitrating (Waiting On Partial) or Arbitrating (Blocked On Partial) confirmation is received, this state shall repeatedly send a Transmit AIP (Waiting On Partial) parameter to the XL transmitter.

If an Arbitrating (Waiting On Connection) confirmation is received, this state shall repeatedly send a Transmit AIP (Waiting On Connection) parameter to the XL transmitter.

Upon entry to this state, this state shall send a Request Path request to the ECM with the following arguments:

a) destination SAS address;
b) source SAS address;
c) protocol;
d) connection rate;
e) arbitration wait time;
f) initiator port bit;
g) initiator connection tag;
h) pathway blocked count; and
i) partial pathway timeout status.

This state maintains the Partial Pathway Timeout timer.

If the Partial Pathway Timeout timer is not already running, the Partial Pathway Timeout timer shall be initialized and started when an Arbitrating (Blocked On Partial) confirmation is received.

If the Partial Pathway Timeout timer is already running, the Partial Pathway Timeout timer shall continue to run if an Arbitrating (Blocked On Partial) confirmation is received.

The Partial Pathway Timeout timer shall be stopped when one of the following confirmations is received:

a) Arbitrating (Waiting On Partial); or
b) Arbitrating (Waiting On Connection);

If the Partial Pathway Timeout timer expires, timeout status is conveyed to the expander connection manager via the partial pathway timeout status argument in the Request Path request.

0.7.4.2 Transition XL1:Request_Path to XL2:Request_Open

This transition shall occur after receiving an Arb Won confirmation.

0.7.4.3 Transition XL1:Request_Path to XL4:Open_Reject

This transition shall occur after receiving an Arb Reject confirmation. This transition shall include an Arb Reject argument corresponding to the Arb Reject confirmation.

0.7.4.4 Transition XL1:Request_Path to XL0:Idle

This transition shall occur after receiving an Arb Lost confirmation.

0.7.4.5 Transition XL1:Request_Path to XL9:Break

This transition shall occur receiving a BREAK Received message.

0.7.5 XL2:Request_Open state

0.7.5.1 State description

This state is used to forward an OPEN address frame through the ECR to a destination phy.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

Upon entry into this state, this state shall send a Transmit Open request to the ECR, received by the destination phy as a Transmit Open indication. The arguments to the Transmit Open request are:

a) destination SAS address;
b) source SAS address;
c) protocol;
d) connection rate;
e) arbitration wait time;
f) initiator port bit;
g) initiator connection tag;
h) features; and
i) pathway blocked count.

0.7.5.2 Transition XL2:Request_Open to XL3:Open_Confirm_Wait

This transition shall occur after sending a Transmit Open request.

0.7.6 XL3:Open_Confirm_Wait state

0.7.6.1 State description

This state waits for confirmation to an OPEN address frame sent on a destination phy.

This state shall send the following messages to the XL transmitter:

a) Transmit AIP (Normal) when an Arb Status (Normal) confirmation is received;
b) Transmit AIP (Waiting On Partial) when an Arb Status (Waiting On Partial) confirmation is received;
c) Transmit AIP (Waiting On Connection) when an Arb Status (Waiting On Connection) confirmation is received;
d) Transmit AIP (Waiting On Device) when an Arb Status (Waiting On Device) confirmation is received;
e) Transmit OPEN_ACCEPT when an Open Accept confirmation is received;
f) Transmit OPEN_REJECT when an Open Reject confirmation is received with the argument from the Open Reject confirmation, after releasing path resources; or
g) request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages when none of the previous conditions are present.

If a Backoff Retry confirmation is received, this state shall release path resources.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM. After an Arb Status (Waiting on Partial) confirmation is received, this state shall repeatedly send a Phy Status (Blocked Partial Pathway) response to the ECM.

0.7.6.2 Transition XL3:Open_Confirm_Wait to XL0:Idle

This transition shall occur after:

a) sending a Transmit OPEN_REJECT message;
b) receiving a Backoff Retry confirmation, after releasing path resources; or
c) receiving a Backoff Reverse Path confirmation.

0.7.6.3 Transition XL3:Open_Confirm_Wait to XL7:Connected

This transition shall occur after sending a Transmit OPEN_ACCEPT message.

0.7.6.4 Transition XL3:Open_Confirm_Wait to XL9:Break

This transition shall occur after sending a Transmit Break request.

0.7.6.5 Transition XL3:Open_Confirm_Wait to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication indication.

0.7.7 XL4:Open_Reject state

0.7.7.1 State description

This state is used to reject a connection request.

This state shall send one of the following messages to the XL transmitter:

a) a Transmit OPEN_REJECT (No Destination) message when an Arb Reject (No Destination) argument is received with the transition into this state;
b) a Transmit OPEN_REJECT (Bad Destination) message when an Arb Reject (Bad Destination) argument is received with the transition into this state;
c) a Transmit OPEN_REJECT (Connection Rate Not Supported) message when an Arb Reject (Bad Connection Rate) argument is received with the transition into this state; or
d) a Transmit OPEN_REJECT (Pathway Blocked) message when an Arb Reject (Pathway Blocked) argument is received with the transition into this state.

0.7.7.2 Transition XL4:Open_Reject to XL0:Idle

This transition shall occur after OPEN_REJECT has been transmitted.

0.7.8 XL5:Forward_Open state

0.7.8.1 State description

This state is used to transmit an OPEN address frame passed with the transition into this state.

If a BROADCAST Received message is received, this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive received (e.g., CHANGE Received).

Upon entry into this state, this state shall send a Transmit OPEN Address Frame message to the XL transmitter with the fields set to the values specified with the transition into this state.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

0.7.8.2 Transition XL5:Forward_Open to XL6:Open_Response_Wait

This transition shall occur after receiving an OPEN Address Frame Transmitted message.

0.7.9 XL6:Open_Response_Wait state

0.7.9.1 State description

This state waits for a response to a transmitted OPEN address frame and determines the appropriate action to take based on the response.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

If a BROADCAST Received message is received before an AIP Received message is received, this state shall send a Broadcast Event Notify request to the BPP with the argument indicating the specific BROADCAST primitive received (e.g., CHANGE Received).

This state shall send the following responses through the ECR to a source phy, received by the source phy as confirmations:

a) an Open Accept response when an OPEN_ACCEPT Received message is received;
b) an Open Reject response when an OPEN_REJECT Received message is received, after releasing any path resources;
c) a Backoff Retry response when an OPEN Address Frame Received message is received containing a higher priority OPEN address frame (see 0.4.3) and the source SAS address and connection rate of the received OPEN address frame are not equal to the destination SAS address and connection rate of the transmitted OPEN address frame, after releasing path resources; or
d) a Backoff Reverse Path response when an OPEN Address Frame Received message is received containing a higher priority OPEN address frame (see 0.4.3) and the source SAS address and connection rate of the received OPEN address frame are equal to the destination SAS address and connection rate of the transmitted OPEN address frame.

This state shall send the following responses through the ECR to a source phy, received by the source phy as confirmations:

a) an Arb Status (Waiting On Device) response when an AIP Received message has not been received;
b) an Arb Status (Normal) response when an AIP Received (Normal) message is received;
c) an Arb Status (Waiting On Partial) response when an AIP Received (Waiting On Partial) message is received;
d) an Arb Status (Waiting On Connection) response when an AIP Received (Waiting On Connection) message is received; and
e) an Arb Status (Waiting On Device) response when an AIP Received (Waiting On Device) message is received.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Partial Pathway) response to the ECM. After an AIP Received (Waiting On Partial) message is received, this state shall repeatedly send a Phy Status (Blocked Partial Pathway) response to the ECM.

0.7.9.2 Transition XL6:Open_Response_Wait to XL0:Idle

This transition shall occur after sending an Open Reject response or a Backoff Retry response.

0.7.9.3 Transition XL6:Open_Response_Wait to XL2:Request_Open

This transition shall occur after sending a Backoff Reverse Path response.

0.7.9.4 Transition XL6:Open_Response_Wait to XL7:Connected

This transition shall occur after sending an Open Accept response.

0.7.9.5 Transition XL6:Open_Response_Wait to XL9:Break

This transition shall occur after sending a Transmit Break response.

0.7.9.6 Transition XL6:Open_Response_Wait to XL10:Break_Wait

This transition shall occur after receiving a Transmit Break indication.

0.7.10 XL7:Connected state

0.7.10.1 State description

This state provides a full-duplex circuit between two phys within an expander device.

This state shall send Transmit Dword messages to the XL transmitter to transmit all dwords received with Transmit Dword indications.

This state shall send Transmit Dword requests to the ECR containing each valid dword except BREAK and CLOSE primitives received with Dword Received messages.

If an invalid dword is received with the Dword Received message and the expander phy is forwarding to an expander phy attached to a SAS physical link, the expander phy shall send an ERROR primitive with the Transmit Dword request instead of the invalid dword.

If an invalid dword or an ERROR primitive is received with Dword Received message and the expander phy is forwarding to an expander phy attached to a SATA physical link, the expander phy shall send a SATA_ERROR primitive with the Transmit Dword request instead of the invalid dword or ERROR primitive.

If a CLOSE Received message is received, this state shall send a Transmit Close request to the ECR with the argument from the CLOSE Received message.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Connected) response to the ECM.

0.7.10.2 Transition XL7:Connected to XL8:Close_Wait

Tis transition shall occur when a Transmit Close indication is received.

0.7.10.3 Transition XL7:Connected to XL9:Break

This transition shall occur after sending a Transmit Break request.

0.7.10.4 Transition XL7:Connected to XL10:Break_Wait

This transition shall occur when a Transmit Break indication is received.

0.7.11 XL8:Close_Wait state

0.7.11.1 State description

This state closes a connection and releases path resources.

This state shall send a Transmit CLOSE message to the XL transmitter with the argument from the Transmit Close indication, then shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the XL transmitter.

This state shall send Transmit Dword requests to the ECR containing each valid dword except BREAK and CLOSE primitives received with Dword Received messages.

If an invalid dword is received with the Dword Received message and the expander phy is forwarding to an expander phy attached to a SAS physical link, the expander phy shall send an ERROR primitive with the Transmit Dword request instead of the invalid dword.

If an invalid dword or an ERROR primitive is received with Dword Received message and the expander phy is forwarding to an expander phy attached to a SATA physical link, the expander phy shall send a SATA_ERROR primitive with the Transmit Dword request instead of the invalid dword or ERROR primitive.

If a CLOSE Received message is received, this state shall release path resources and send a Transmit Close request to the ECR with the argument from the CLOSE Received ~~mesage~~message.

If a BREAK Received message is received, this state shall send a Transmit Break request to the ECR.

This state shall repeatedly send a Phy Status (Connected) response to the ECM.

0.7.11.2 Transition XL8:Close_Wait to XL0:Idle

This transition shall occur after sending a Transmit Close request.

0.7.11.3 Transition XL8:Close_Wait to XL9:Break

This transition shall occur after sending a Transmit Break request.

0.7.11.4 Transition XL8:Close_Wait to XL10:Break_Wait

This transition shall occur after a Transmit Break indication is received.

0.7.12 XL9:Break state

0.7.12.1 State description

This state closes any connection and releases path resources.

This state shall send a Transmit BREAK message to the XL transmitter.

0.7.12.2 Transition XL9:Break to XL0:Idle

This transition shall occur after sending a Transmit BREAK message to the XL transmitter.

0.7.13 XL10:Break_Wait state

0.7.13.1 State description

This state closes any connection and releases path resources.

This state shall send a Transmit BREAK message to the XL transmitter. After transmitting the BREAK this state shall initialize and start the Break Timeout timer.

0.7.13.2 Transition XL10:Break_Wait to XL0:Idle

This transition shall occur after a BREAK Received message is received or after the Break Timeout timer expires.

0.8 SSP link layer

0.8.1 Opening an SSP connection

An SSP phy that accepts an OPEN address frame shall transmit at least one RRDY in that connection within 1 ms of transmitting an OPEN_ACCEPT. If the SSP phy is not able to grant credit, it shall respond with OPEN_REJECT (RETRY) and not accept the connection request.

> Editor’s Note 1: new paragraph below created in 2/25 meeting. Put this in the port layer “SSP wide port rules” section.

To prevent livelocks, a wide SSP port shall not reject an incoming connection request on one phy because it has an outgoing connection request on another phy. If multiple connections are established and the wide SSP port still has a frame to transmit, the wide SSP port shall transmit at least one frame on any of the connections before preparing to close one of the connections to prevent livelocks.

0.8.2 Full duplex

SSP is a full duplex protocol. An SSP phy may receive an SSP frame or primitive in a connection the same time it is transmitting an SSP frame or primitive in the same connection. A wide SSP port may send and/or receive SSP frames or primitives concurrently on different connections (i.e., on different phys).

When a connection is open and an SSP phy has no more SSP frames to transmit on that connection, it shall transmit DONE to start closing the connection. The other direction may still be active, so the DONE may be followed by one or more CREDIT_BLOCKED, RRDY, ACK, or NAKs.

0.8.3 SSP frame transmission and reception

During an SSP connection, SSP frames are preceded by SOF and followed by EOF as shown in figure 11.

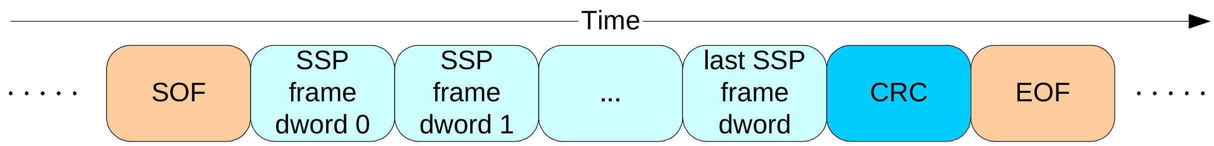

Figure 11 — SSP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5). The SSP link layer state machine checks that the frame is not too short and that the CRC is valid (see 0.8.7.7).

Receiving SSP phys shall acknowledge SSP frames within 1 ms if not discarded as described in 0.8.7.7 with either a positive acknowledgement (ACK) or a negative acknowledgement (NAK). ACK means the SSP frame was received into a frame buffer without errors. NAK (CRC ERROR) means the SSP frame was received with a CRC error.

Some interlocked frames which are NAKed may be retried by the transport layer (see ~~9.2.4~~9.2.4). Non-interlocked frames which are NAKed shall not be retried by any layer. The SCSI command associated with a NAKed frame that is not successfully retried shall be terminated with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA PHASE CRC ERROR DETECTED (see SPC-3).

0.8.4 SSP flow control

SSP phys grant credit for permission to transmit frames with RRDY. Each RRDY increments credit by one frame. Frame transmission decrements credit by one frame. Credit of zero frames is established at the beginning of each connection.

SSP phys shall not increment credit past 255 frames.

To prevent deadlocks where an SSP initiator port and SSP target port are both waiting on each other to provide credit, an SSP initiator port shall never refuse to provide credit by withholding RRDY because it needs to transmit a frame itself. It may refuse to provide credit for other reasons (e.g., temporary buffer full conditions).

An SSP target port may refuse to provide credit for any reason, including because it needs to transmit a frame itself.

0.8.5 Interlocked frames

Table 10 shows which SSP frames shall be interlocked and which are non-interlocked.

Table 10 — SSP frame interlock requirements

SSP frame type	Interlock requirement
COMMAND	Interlocked
TASK	Interlocked
XFER_RDY	Interlocked
DATA	Non-interlocked
RESPONSE	Interlocked
See ~~9.2~~9.2 for SSP frame type definitions.	

Before transmitting an interlocked frame, an SSP phy shall wait for all SSP frames to be acknowledged with ACK or NAK, even if credit is available. After transmitting an interlocked frame, an SSP phy shall not transmit another SSP frame until it has been acknowledged with ACK or NAK, even if credit is available.

Before transmitting a non-interlocked frame, an SSP phy shall wait for:

a) all non-interlocked frames with different tags; and
b) all interlocked frames;

to be acknowledged with ACK or NAK, even if credit is available.

After transmitting a non-interlocked frame, an SSP phy may transmit another non-interlocked frame with the same tag if credit is available. The phy shall not transmit:

a) a non-interlocked frame with a different tag; or
b) an interlocked frame;

until all SSP frames have been acknowledged with ACK or NAK, even if credit is available.

Interlocking does not prevent transmitting and receiving interlocked frames simultaneously (e.g., an SSP initiator phy could be transmitting a COMMAND frame while receiving XFER_RDY, DATA, or RESPONSE frames for a different command).

An SSP phy may transmit primitives responding to traffic it is receiving (e.g. an ACK or NAK to acknowledge an SSP frame, an RRDY to grant more receive credit, or a CREDIT_BLOCKED to indicate credit is not forthcoming) while waiting for an interlocked frame it transmitted to be acknowledged. These primitives may also be interspersed within an SSP frame.

Figure 12 shows an example of interlocked frame transmission.

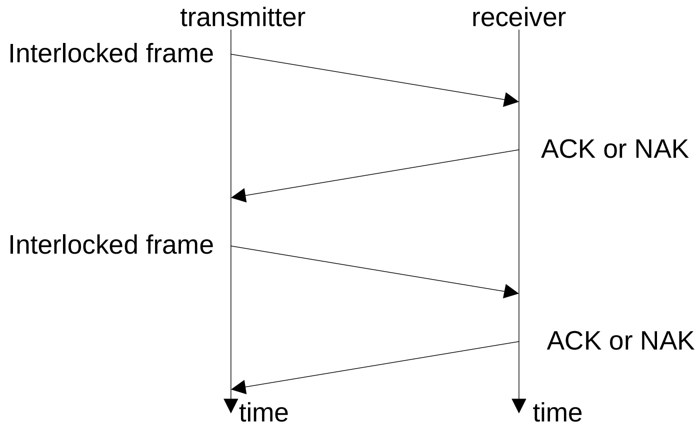

Figure 12 — Interlocked frames

Figure 13 shows an example of non-interlocked frame transmission with the same tags

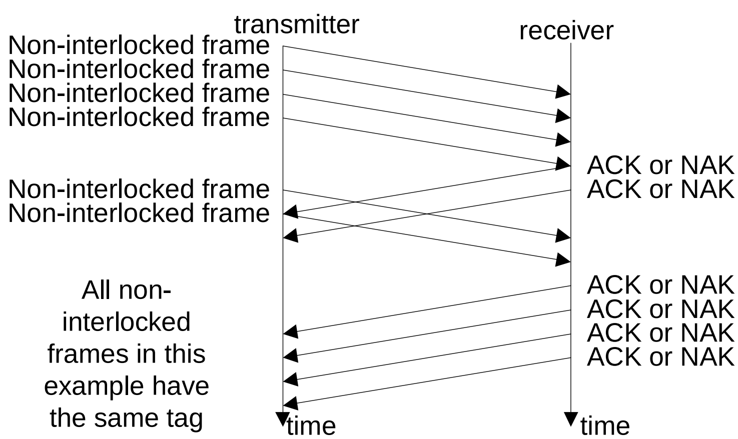

Figure 13 — Non-interlocked frames with the same tag

Figure 14 shows an example of non-interlocked frame transmission with different tags.

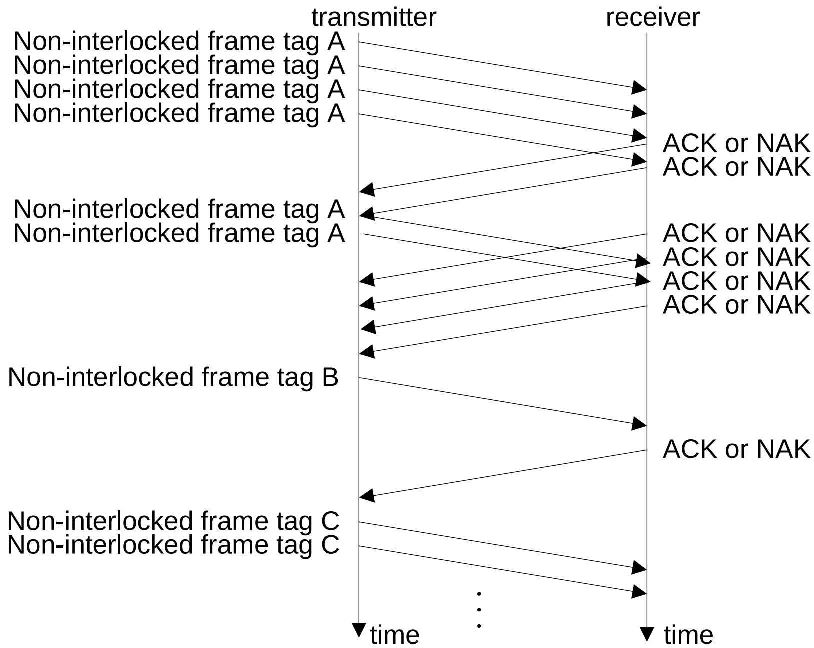

Figure 14 — Non-interlocked frames with different tags

0.8.6 Closing an SSP connection

DONE shall be exchanged prior to closing an SSP connection.

When a source SSP phy has no SSP frames to transmit, it should transmit DONE (NORMAL). When a destination SSP phy has no SSP frames to transmit, it may wait for a vendor-specific period of time, and then shall transmit DONE (NORMAL). There are several versions of the DONE primitive indicating additional information about why the SSP connection is being closed:

a) DONE (NORMAL) indicates normal completion; the transmitter has no more SSP frames to transmit;
b) DONE (CREDIT TIMEOUT) indicates the transmitter still has SSP frames to transmit, but did not receive an RRDY granting frame credit within 1 ms; and
c) DONE (ACK/NAK TIMEOUT) indicates the transmitter transmitted an SSP frame but did not receive the corresponding ACK or NAK within 1 ms. As a result, the ACK/NAK count is not balanced and the transmitter is going to transmit a BREAK in 1 ms unless the recipient replies with DONE and the connection is closed.

After transmission of DONE, a device shall not transmit any more SSP frames during this connection. However, a device may transmit ACK, NAK, RRDY, and CREDIT_BLOCKED after transmitting DONE if the other device is still transmitting SSP frames in the reverse direction. Once an SSP phy has both transmitted and received DONE, it shall close the connection by transmitting CLOSE (NORMAL) (see 0.4.8).

Figure 15 shows the sequence for a closing an SSP connection.

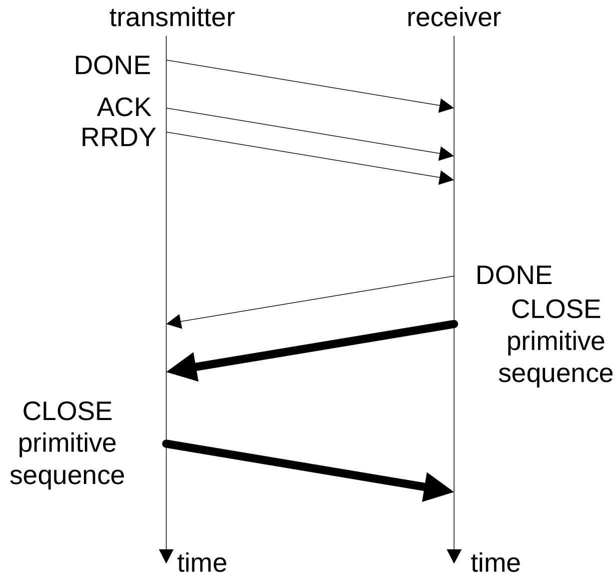

Figure 15 — Closing an SSP connection example

0.8.7 SSP (link layer for SSP phys) state machines

0.8.7.1 SSP state machines overview

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

a) SSP_TIM (transmit interlocked frame monitor) state machine (see 0.8.7.3);
b) SSP_TCM (transmit frame credit monitor) state machine (see 0.8.7.4);
c) SSP_D (DONE control) state machine (see 0.8.7.5);
d) SSP_TF (transmit frame control) state machine (see 0.8.7.6);
e) SSP_RF (receive frame control) state machine (see 0.8.7.7);
f) SSP_RCM (receive frame credit monitor) state machine (see 0.8.7.8);
g) SSP_RIM (receive interlocked frame monitor) state machine (see 0.8.7.9);
h) SSP_TC (transmit credit control) state machine (see 0.8.7.10); and
i) SSP_TAN (transmit ACK/NAK control) state machine (see 0.8.7.11).

All the SSP state machines shall start after receiving an Enable Disable SSP (Enable) message from the ~~SL~~ SL_CC state machine (see 0.6).

All the SSP state machines shall terminate after:

a) receiving an Enable Disable SSP (Disable) message from the ~~SL~~ SL_CC state machine;
b) receiving a Request Close message from the SSP_D state machine indicating that the connection has been closed; or
c) receiving a Request Break message from the SSP_D state machine indicating that a BREAK has been transmitted.

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

The SSP state machines shall maintain the timers listed in table 11.

Table 11 — SSP link layer timers

Timer	Initial value
ACK/NAK Timeout timer	1 ms
DONE Timeout timer	1 ms
Credit Timeout timer	1 ms

~~Figure 16 shows the SSP state machines and states related to frame transmissionSSP (link layer for SSP phys) state machines (part 1 - frame transmission)~~

~~Figure 17~~Figure 16 shows the SSP state machines and states related to frame ~~reception~~transmission.

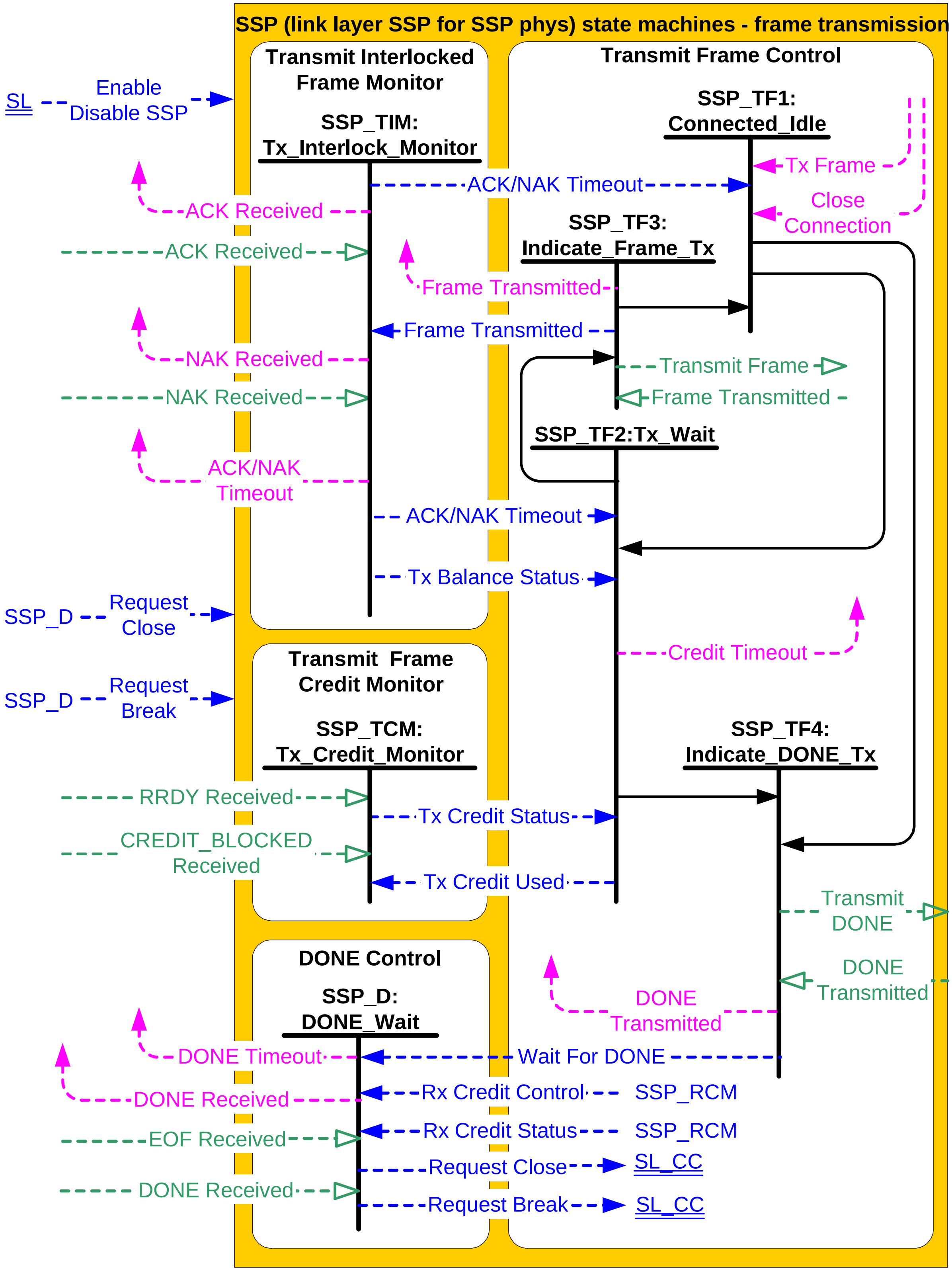

Figure 16 — SSP (link layer for SSP phys) state machines (part ~~2~~ 1 - frame ~~reception~~transmission)

~~Figure 17~~Figure 17 shows the SSP state machines and states related to ~~primitive transmission~~frame reception.

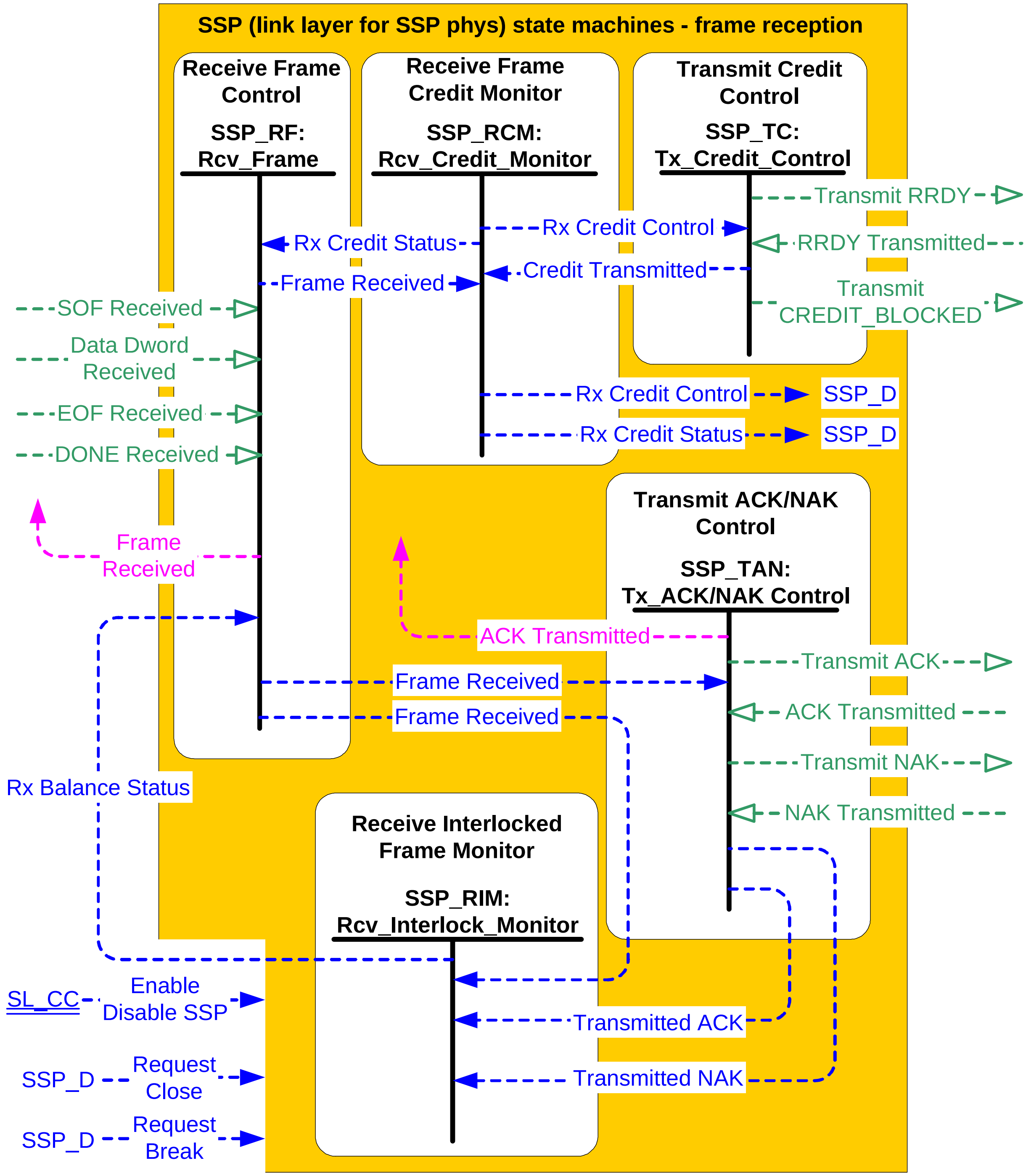

Figure 17 — SSP (link layer for SSP phys) state machines (part ~~3~~ 2 - ~~primitive transmission~~frame reception)

0.8.7.2 SSP transmitter and receiver

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

a) Transmit RRDY with an argument indicating the specific type (e.g., Transmit RRDY (Normal));
b) Transmit CREDIT_BLOCKED;
c) Transmit ACK;

d) Transmit NAK with an argument indicating the specific type (e.g., Transmit NAK (CRC Error));
e) Transmit Frame; and
f) Transmit DONE with an argument indicating the specific type (e.g., Transmit DONE (Normal)).

The SSP transmitter sends the following messages to the SSP state machines:

a) DONE Transmitted;
b) RRDY Transmitted;
c) CREDIT_BLOCKED Transmitted;
d) ACK Transmitted;
e) NAK Transmitted; and
f) Frame Transmitted.

When the SSP transmitter is not processing a message to transmit, it shall transmit idle dwords.

The SSP receiver sends the following messages to the SSP state machines:

a) ACK Received;
b) NAK Received;
c) RRDY Received;
d) CREDIT_BLOCKED Received;
e) EOF Received;
f) DONE Received with an argument indicating the specific type (e.g., DONE Received (Normal));
g) SOF Received;
h) Data Dword Received; and
i) EOF Received.

0.8.7.3 SSP_TIM (transmit interlocked frame monitor) state machine

The SSP_TIM state machine's function is to ensure that ACKs or NAKs are received for each transmitted frame before the ACK/NAK timeout. This state machine consists of one state.

This state machine monitors the number of frames transmitted and monitors the number of ACKs and NAKs received. This state machine ensures that an ACK or NAK is received for each frame transmitted and indicates a timeout if they are not.

The Frame Transmitted message shall be used by this state machine to count the number of frames transmitted.

When the number of Frame Transmitted messages received equals the number of ACK Received and NAK Received messages received then the ACK/NAK count is balanced and this state machine shall send the Tx Balance Status (Balanced) message to the SSP_TF2:Tx_Wait state. When the number of Frame Transmitted messages received does not equal the number of ACK Received and NAK Received messages received then this the ACK/NAK count is not balanced and this state machine shall send the Tx Balance Status (Not Balanced) message to the SSP_TF2:Tx_Wait state.

If the ACK/NAK count is not balanced and an ACK Received message is received this state machine shall:

a) use the ACK Received message to count the number of ACKs and NAKs received; and
b) send an ACK Received confirmation to the port layer each time the ACK Received message is received.

If the ACK/NAK count is not balanced and an NAK Received message is received this state machine shall:

a) use the NAK Received message to count the number of ACKs and NAKs received; and
b) send an NAK Received confirmation to the port layer each time the NAK Received message is received.

If the ACK/NAK count is balanced, the ACK Received message and NAK Received message shall be ignored and the ACK/NAK Timeout timer shall be stopped.

Each time the ACK/NAK count is not balanced, the ACK/NAK Timeout timer shall be initialized and started. The ACK/NAK Timeout timer shall be re-initialized each time an ACK or NAK is counted. If the ACK/NAK

Timeout timer expires, this state machine shall send the ACK/NAK Timeout confirmation to the port layer and to the following states:

a) SSP_TF1:Connected_Idle; and
b) SSP_TF2:Tx_Wait state.

When this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message the number of frames transmitted shall be set to the number of ACKs and NAKs received.

0.8.7.4 SSP_TCM (transmit frame credit monitor) state machine

The SSP_TCM state machine's function is to ensure that credit is available from the originator before a frame is transmitted. This state machine consists of one state.

This state machine shall keep track of the number of transmit frame credits received versus the number of transmit frame credits used. This state machine adds transmit frame credit for each RRDY Received message received and subtracts transmit frame credit for each Tx Credit Used message received. This state machine shall remember any CREDIT_BLOCKED Received message that is received.

When transmit frame credit is available, this state machine shall send the Tx Credit Status (Credit Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and credit is not blocked, this state machine shall send the Tx Credit Status (Credit Not Available) message to the SSP_TF2:Tx_Wait state.

When transmit frame credit is not available and credit is blocked, this state machine shall send the Tx Credit Status (Credit Blocked) message to the SSP_TF2:Tx_Wait state.

When this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message transmit frame credit shall be set to not available and credit shall not be blocked.

0.8.7.5 SSP_D (DONE control) state machine

The SSP_D state machine's function is to ensure a DONE has been received and transmitted before the ~~SL~~ SL_CC state machine disables the SSP state machines. This state machine consists of one state.

This state machine ensures that a DONE is received and transmitted before the connection is closed. The DONE may be transmitted and received in any order.

If the DONE Received message has been received before the Wait For DONE message is received, this state machine shall send the Request Close message to the ~~SL~~ SL_CC state machine (see 0.6) and all the SSP state machines.

If the Rx Credit Status (Extended) message or the Rx Credit Control (Blocked) message was received before the Wait For Done DONE (NORMAL) message or the Wait For DONE (CREDIT TIMEOUT) message is received, then this state shall initialize and start the DONE Timeout timer after receiving the Wait For DONE (NORMAL) message or the Wait For DONE (CREDIT TIMEOUT) message if a DONE Received message has not been received.

If the DONE Received message has not been received and the Wait For DONE (NORMAL) message or the Wait For DONE (CREDIT TIMEOUT) message has been received, this state machine shall initialize and start the DONE Timeout timer each time:

a) the Rx Credit Status (Extended) message is received; or
b) the Rx Credit Control (Blocked) message is received.

If the Wait For DONE (NORMAL) message or the Wait for DONE (CREDIT TIMEOUT) message was received, the DONE timeout timer shall be reinitialized each time the EOF Received message is received.

If the Wait For DONE (NORMAL) message or the Wait for DONE (CREDIT TIMEOUT) message was received, the DONE timeout timer shall be stopped after:

a) the Rx Credit Status (Exhausted) message is received; and
b) the Rx Credit Control (Blocked) message has not been received.

If the DONE Received message has not been received ~~when~~ and the Wait For DONE (ACK/NAK TIMEOUT) message ~~is~~ has been received, this state machine shall initialize and start the DONE Timeout ~~timer. If the Wait For DONE~~ timer (~~NORMAL) message or the Wait for DONE (CREDIT TIMEOUT) message was received~~i.e., ~~the DONE timeout timer~~ this state shall ~~be reinitialized each time the EOF Received message is received. If the Wait For DONE (ACK/NAK TIMEOUT) message was received~~ not reinitialize the DONE Timeout ~~timer shall not be reinitialized.~~timer).

If the DONE Received message is received before the DONE Timeout timer expires, this state machine shall send the Request Close message to the ~~SL~~ SL_CC state machine and all the SSP state machines.

If the DONE Received message is not received before the DONE Timeout timer expires, this state machine shall:

a) send a DONE Timeout confirmation to the port layer; and
b) send a Request Break message to the ~~SL~~ SL_CC state machine and all the SSP state machines.

Any time a DONE Received message is received this state machine shall send a DONE Received confirmation to the port layer. A DONE Received (ACK/NAK Timeout) confirmation informs the port layer that the SSP transmitter is going to close the connection within 1 ms; other DONE Received confirmations (e.g., DONE Received (Close Connection) and DONE Received (Credit Timeout)) may be used by the application layer to decide when to reuse tags.

0.8.7.6 SSP_TF (transmit frame control) state machine

0.8.7.6.1 SSP_TF state machine overview

The SSP_TF state machine's function is to control when the SSP transmitter transmits SOF, frame dwords, EOF, and DONE. This state machine consists of the following states:

a) SSP_TF1:Connected_Idle (see 0.8.7.6.2)(initial state);
b) SSP_TF2:Tx_Wait (see 0.8.7.6.3);
c) SSP_TF3:Indicate_Frame_Tx (see 0.8.7.6.4); and
d) SSP_TF4:Indicate_DONE_Tx (see 0.8.7.6.5).

0.8.7.6.2 SSP_TF1:Connected_Idle state

0.8.7.6.2.1 State description

This state waits for a request to transmit a frame or to close the connection.

0.8.7.6.2.2 Transition SSP_TF1:Connected_Idle to SSP_TF2:Tx_Wait

This transition shall occur after a Tx Frame request is received or a Close Connection request is received.

If a Tx Frame (Balance Required) request was received this transition shall include a Transmit Frame Balance Required argument.

If a Tx Frame (Balance Not Required) request was received this transition shall include a Transmit Frame Balance Not Required argument.

If a Close Connection request was received this transition shall include a Close Connection argument.

0.8.7.6.2.3 Transition SSP_TF1:Connected_Idle to SSP_TF4:Indicate_DONE_Tx

This transition shall occur if an ACK/NAK Timeout message is received. This transition shall include an ACK/NAK Timeout argument.

0.8.7.6.3 SSP_TF2:Tx_Wait state

0.8.7.6.3.1 State description

This state monitors the Tx Balance Status message and the Tx Credit Status message to ensure that frames are transmitted and connections are closed at the proper time.

If this state is entered from the SSP_TF1:Connected_Idle state with a Transmit Frame Balance Required argument or a Transmit Frame Balance Not Required argument, and:

a) if the last Tx Credit Status message received had an argument of Not Available this state shall initialize and start the Credit Timeout timer; or
b) if the last Tx Credit Status message had an argument other than Not Available this state shall stop the Credit Timeout timer.

0.8.7.6.3.2 Transition SSP_TF2:Tx_Wait to SSP_TF3:Indicate_Frame_Tx

This transition shall occur if this state was entered from the SSP_TF1:Connected_Idle state with an argument of Transmit Frame Balance Required if:

a) the last Tx Balance Status message received had an argument of Balanced; and
b) the last Tx Credit Status message received had an argument of Credit Available.

This transition shall occur if this state was entered from the SSP_TF1:Connected_Idle state with an argument of Transmit Frame Balance Not Required and if the last Tx Credit Status message received had an argument of Credit Available.

This transition shall occur after sending a Tx Credit Used message to the SSP_TCM state machine.

0.8.7.6.3.3 Transition SSP_TF2:Tx_Wait to SSP_TF4:Indicate_DONE_Tx

This transition shall include a Close Connection argument if this state was entered from the SSP_TF1:Connected_Idle state with an argument of Close Connection and the last Tx Balance Status message received had an argument of Balanced.

This transition shall include a Credit Timeout argument if this state was entered from the SSP_TF1:Connected_Idle state with a Transmit Frame Balance Required argument or a Transmit Frame Balance Not Required argument and the last Tx Credit Status message received had an argument of Blocked.

This transition shall include a Credit Timeout argument if:

a) this state was entered from the SSP_TF1:Connected_Idle state with a Transmit Frame Balance Required argument or a Transmit Frame Balance Not Required argument;
b) the Credit Timeout timer expires before a Tx Credit Status (Available) message is received; and
c) the last Tx Balance Status message received had an argument of Balanced or a Tx Balance Status (Balanced) message is received before a ACK/NAK Timeout message is received.

This transition shall include an ACK/NAK Timeout argument if an ACK/NAK Timeout message is received.

0.8.7.6.4 SSP_TF3:Indicate_Frame_Tx state

0.8.7.6.4.1 State description

This state shall request a frame transmission by sending a Transmit Frame message to the SSP transmitter. Each time a Transmit Frame message is sent to the SSP transmitter, one SSP frame (i.e., SOF, frame contents, and EOF) is transmitted.

In this state receiving a Frame Transmitted message indicates that the frame has been transmitted.

0.8.7.6.4.2 Transition SSP_TF3:Indicate_Frame_Tx to SSP_TF1:Connected_Idle

This transition shall occur after:

a) receiving a Frame Transmitted message;
b) sending an Frame Transmitted message to the SSP_TIM state machine; and
c) sending a Frame Transmitted confirmation to the port layer.

0.8.7.6.5 SSP_TF4:Indicate_DONE_Tx state

This state shall send one of the following messages to an SSP transmitter:

a) a Transmit DONE (Normal) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Close Connection;
b) a Transmit DONE (ACK/NAK Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state or the Connected_Idle state with an argument of ACK/NAK Timeout; or
c) a Transmit DONE (CREDIT Timeout) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Credit Timeout.

After a DONE Transmitted message is received this state shall send the DONE Transmitted confirmation to the port layer and send one of the following messages:

a) Wait For DONE (NORMAL) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Close Connection;
b) Wait For DONE (ACK/NAK TIMEOUT) message if this state was entered from the SSP_TF2:Tx_Wait state or the Connected_Idle state with an argument of ACK/NAK Timeout; or
c) Wait For DONE (CREDIT TIMEOUT) message if this state was entered from the SSP_TF2:Tx_Wait state with an argument of Credit Timeout.

0.8.7.7 SSP_RF (receive frame control) state machine

The SSP_RF state machine's function is to receive frames and determine whether or not those frames were received successfully. This state machine consists of one state.

This state machine:

a) checks the frame to determine if the frame should be accepted or discarded:
b) checks the frame to determine if an ACK or NAK should be transmitted; and
c) sends a Frame Received confirmation to the port layer.

The frame (i.e., all the dwords between an SOF and EOF) shall be discarded if any of the following conditions are true:

a) the number of data dwords between the SOF and EOF is less than 7;
b) the number of data dwords after the SOF is greater than 263 data dwords;
c) the Rx Credit Status (Credit Exhausted) message is received; or
d) the DONE Received message is received.

If consecutive SOF Received messages are received without an intervening EOF Received message (i.e., SOF, data dwords, SOF, data dwords, and EOF instead of SOF, data dwords, EOF, SOF, data dwords, and EOF) then this state machine shall discard all dwords between those SOFs.

If the frame is discarded then no further action is taken by this state machine. If the frame is not discarded then this state machine shall:

a) send a Frame Received message to the SSP_RCM state machine; and
b) send a Frame Received message to the SSP_RIM state machine;

If the frame CRC is good and the frame contained no invalid data dwords, this state machine shall send a Frame Received (Successful) message to the SSP_TAN1:Idle state and:

a) if the last Rx Balance Status message received had an argument of Balanced, send a Frame Received (ACK/NAK Balanced) confirmation to the port layer; or
b) if the last Rx Balance Status message received had an argument of Not Balanced, send a Frame Received (ACK/NAK Not Balanced) confirmation to the port layer.

If the frame CRC is bad or the frame contained invalid data dwords, this state machine shall send a Frame Received (Unsuccessful) message to the SSP_TAN1:Idle state.

0.8.7.8 SSP_RCM (receive frame credit monitor) state machine

The SSP_RCM state machine's function is to ensure that there was credit given to the originator for every frame that is received. This state machine consists of one state.

This state machine monitors the receiver's resources and keeps track of the number of RRDYs transmitted versus the number of frames received.

Any time resources are released or become available this state machine shall send the Rx Credit ~~Status~~ Control (Available) message to the ~~SSP_TC1:Idle state~~SSP_TC state machine. This state machine shall only send the Rx Credit ~~Status~~ Control (Available) message to the ~~SSP_TC1:Idle~~ SSP_TC state machine after frame receive resources become available. The specifications for when or how resources become available is outside the scope of this standard.

This state machine may send the Rx Credit ~~Status~~ Control (Blocked) message to the ~~SSP_TC1:Idle~~ SSP_TC state machine and the SSP_D state machine indicating that no more credit is going to be sent during this connection. After sending the Rx Credit ~~Status~~ Control (Blocked) message to the SSP_TC state machine and the ~~SSP_TC1:Idle state~~SSP_D state machine, this state machine shall not send the Rx Credit ~~Status~~ Control (Available) message to the ~~SSP_TC1:Idle~~ SSP_TC state machine or the SSP_D state machine for the duration of the current connection. The Rx Credit ~~Status~~ Control (Blocked) message should be sent to the SSP_TC state machine and the ~~SSP_TC1:Idle~~ SSP_D state machine when no further credit is going to become available within a credit timeout (i.e., less than 1 ms).

This state machine shall indicate through the Rx Credit ~~Status~~ Control message only the amount of resources available to handle received frames (e.g., if this state machine has resources for 5 frames the maximum number of Rx Credit ~~Status~~ Control requests with the Available argument outstanding is 5).

This state machine shall use the ~~RRDY~~ Credit Transmitted message to keep track of the number of RRDYs transmitted. This state machine shall use the Frame Received message to keep a track of the number of frames received.

Any time the number of ~~RRDY~~ Credit Transmitted messages received exceeds the number of Frame Received messages received this state machine shall send a Rx Credit Status (Credit Extended) message to the SSP_RF state machine and the SSP_D state machine.

Any time the number of ~~RRDY~~ Credit Transmitted messages received equals the number of Frame Received messages received this state machine shall send a Rx Credit Status (Credit Exhausted) message to the SSP_RF state machine and the SSP_D state machine.

If this state machine receives an Enable Disable SSP (Enable) message, Request Close message, or Request Break message the frame receive resources shall be initialized to the no credit value for the current connection.

0.8.7.9 SSP_RIM (receive interlocked frame monitor) state machine

The SSP_RIM state machine's function is to inform the SSP_RF state machine when the number of ACKs and NAKs transmitted equals the number of the EOFs received. This state machine consists of one state.

This state machine monitors the number of frames received versus the number of ACKs and NAKs transmitted.

This state machine shall use the ACK Transmitted message and the NAK Transmitted message to keep track of the number of ACKs and NAKs transmitted. This state machine shall use the Frame Received message to keep a track of the number of frames received.

Any time the number of the ACK Transmitted messages and the number of NAK Transmitted messages received equals the number of Frame Received messages received this state machine shall send an Rx Balance Status (Balanced) message to the SSP_RF state machine.

Any time the number of the ACK Transmitted messages and the number of NAK Transmitted messages received does not equal the number of Frame Received messages received this state machine shall send an Rx Balance Status (Not Balanced) message to the SSP_RF state machine.

When the ~~SL~~ SL_CC state machine sends the Enable Disable SSP (Enable) message the number of the ACKs and NAKs transmitted shall be set to the number of frames received.

0.8.7.10 SSP_TC (transmit credit control) state machine

~~0.8.7.10.1 SSP_TC state machine overview~~

The SSP_TC state machine's function ~~it~~ is to control the sending of requests to transmit an RRDY or CREDIT_BLOCKED. This state machine consists of ~~the following states:~~one state.

~~a) SSP_TC1:Idle (see 0.8.7.10.2)(initial state); and~~
~~b) SSP_TC2:Indicate_Credit_Tx (see 0.8.7.10.3).~~

~~0.8.7.10.2 SSP_TC1:Idle state~~

~~0.8.7.10.2.1 State description~~

This state machine waits for a Rx Credit ~~Status~~ Control message to be received.

Any time this state machine receives a Rx Credit Control (Available) message it shall send a number of Transmit RRDY (Normal) messages to the SSP transmitter as indicated by the amount of resources available to handle received frames (e.g., if the Available argument indicates 5 RRDYs are to be transmitted this state machine sends 5 Transmit RRDY (Normal) messages to the SSP transmitter).

~~When~~ Any time this state ~~is entered from the SSP_TC2:Indicate_Credit_Tx state with an argument of~~ machine receives a RRDY Transmitted message it shall send ~~an RRDY~~ a Credit Transmitted message to the SSP_RCM state machine.

~~0.8.7.10.2.2 Transition SSP_TC1:Idle to SSP_TC2:Indicate_Credit_Tx~~

~~This transition shall include a Transmit RRDY argument if a Rx Credit Status (Available) message was received.~~

~~This transition shall include a CREDIT_BLOCKED argument if a Rx Credit Status (Blocked) message was received.~~

~~0.8.7.10.3 SSP_TC2:Indicate_Credit_Tx state~~

~~0.8.7.10.3.1 State description~~

~~If this state is entered into from the SSP_TC1 state with an argument of Transmit RRDY, this state shall send a Transmit RRDY (Normal) message to the SSP transmitter.~~

~~If~~ Any time this state ~~is entered into from the SSP_TC1 state with an argument of CREDIT_BLOCKED, this state~~ machine receives a Rx Credit Control (Blocked) message it shall send a Transmit CREDIT_BLOCKED message to the SSP transmitter.

~~0.8.7.10.3.2 Transition SSP_TC2:Indicate_Credit_Tx to SSP_TC1:Idle~~

~~This transition shall occur after receiving an RRDY Transmitted message or the CREDIT_BLOCKED Transmitted message.~~

0.8.7.11 SSP_TAN (transmit ACK/NAK control) state machine

~~0.8.7.11.1 SSP_TAN state machine overview~~

The SSP_TAN state machine's function is to control the sending of requests to transmit an ACK or ~~NAK~~NAK to the SSP transmitter. This state machine consists of ~~the following states:~~one state.

~~a) SSP_TAN1:Idle (see 0.8.7.11.2)(initial state); and~~
~~b) SSP_TAN2:Indicate_ACK/NAK_Tx (see 0.8.7.11.3).~~

~~0.8.7.11.2 SSP_TAN1:Idle state~~

~~0.8.7.11.2.1 State description~~

This state waits for a Frame Received ~~message~~message to be received.

Any time this state receives a Frame Received (Successful) message this state shall send a Transmit ACK message to the SSP transmitter.

Any time this state receives a Frame Received (Unsuccessful) message it shall send a Transmit NAK (CRC Error) message to the SSP transmitter.

If multiple Frame Received (Unsuccessful) messages and Frame Received (Successful) messages are received, then the order in which the Transmit ACK messages and Transmit NAK messages are sent to the SSP transmitter shall be the same order as the Frame Received (Unsuccessful) messages and Frame Received (Successful) messages were received.

~~When~~ Any time this state ~~is entered into from the SSP_TAN2:Indicate_ACK/NAK_Tx state with~~ receives an ACK Transmitted ~~argument~~ message this state shall:

a) send ~~an~~ a Transmitted ACK ~~Transmitted~~ message to the SSP_RIM state machine~~:~~; and
b) send an ACK Transmitted confirmation to the port layer.

~~When this state is entered into from the SSP_TAN2:Indicate_ACK/NAK_Tx state with a NAK Transmitted argument it shall send a NAK Transmitted message to the SSP_RIM state machine.~~

~~0.8.7.11.2.2 Transition SSP_TAN1:Idle to SSP_TAN2:Indicate_ACK/NAK_Tx~~

~~This transition shall include a Transmit ACK argument if a Frame Received (Successful) message was received.~~

~~This transition shall include a Transmit NAK argument if a Frame Received (Unsuccessful) message was received.~~

~~If multiple Frame Received (Unsuccessful) messages and Frame Received (Successful) messages are received, then the order in which the Transmit ACK arguments and Transmit NAK arguments are passed to the SSP_TAN2:Indicate_ACK/NAK_Tx state shall be the same order as the Frame Received (Unsuccessful) messages and Frame Received (Successful) messages were received.~~

~~0.8.7.11.3 SSP_TAN2:Indicate_ACK/NAK_Tx state~~

~~0.8.7.11.3.1 State description~~

~~If this state is entered into from the SSP_TAN1 state with an argument of Transmit ACK, this state shall send a Transmit ACK message to the SSP transmitter.~~

~~If~~ Any time this state ~~is entered into from the SSP_TAN1 state with an~~ receives a NAK Transmitted argument ~~of Transmit NAK, this state~~ it shall send a ~~Transmit~~ Transmitted NAK ~~(CRC Error)~~ message to the ~~SSP transmitter~~SSP_RIM state machine.

~~0.8.7.11.3.2 Transition SSP_TAN2:Indicate_ACK/NAK_Tx to SSP_TAN1:Idle~~

~~This transition shall occur after receiving an ACK Transmitted message or the NAK Transmitted message.~~

0.9 STP link layer

0.9.1 STP frame transmission and reception

STP frame transmission is defined by SATA (see ATA/ATAPI-7 V3). During an STP connection, frames are preceded by SATA_SOF and followed by SATA_EOF as shown in figure 18.

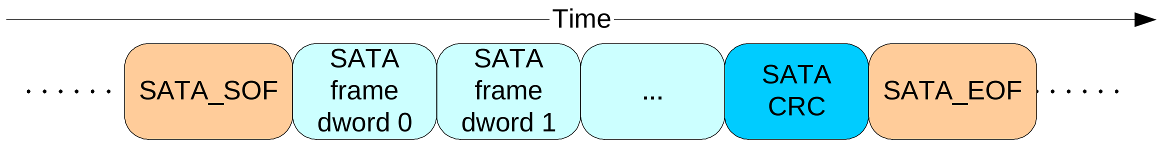

Figure 18 — STP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5).

Other primitives may be interspersed during the connection as defined by SATA.

STP encapsulates SATA with connection management.

0.9.2 STP flow control

Each STP port (i.e., STP initiator port and STP target port) and expander port through which the STP connection is routed shall implement the SATA flow control protocol on each physical link. The flow control primitives are not forwarded through expander devices like other dwords.

When an STP port is receiving a frame and its buffer begins to fill up, it shall transmit SATA_HOLD. After transmitting SATA_HOLD, it shall accept the following number of data dwords for the frame:

a) 24 dwords at 1,5 Gbps; or
b) 28 dwords at 3,0 Gbps.

When an STP port is transmitting a frame and receives SATA_HOLD, it shall transmit no more than 20 data dwords for the frame and respond with SATA_HOLDA.

NOTE 2 - The receive buffer requirements are based on (20 + (4 $\times$ n)) where n is 1 for 1,5 Gbps and 2 for 3,0 Gbps. The 20 portion of this equation is based on the frame transmitter requirements (see ATA/ATAPI-7 V3). The (4 $\times$ n) portion of this equation is based on:
a) One-way propagation time on a 10 m cable = (5 ns/m propagation delay) $\times$ (10 m cable) = 50 ns;
b) Round-trip propagation time on a 10 m cable = 100 ns (e.g., time to send SATA_HOLD and receive SATA_HOLDA);
c) Time to transmit a 1,5 Gbps dword = (0,666 ns/bit unit interval) $\times$ (40 bits/dword) = 26,667 ns; and
d) Number of 1,5 Gbps dwords on the wire during round-trip propagation time = (100 ns / 26,667 ns) = 3,75.
Receivers may support longer cables by providing larger buffer sizes.

When a SATA host port in an STP/SATA bridge is receiving a frame from a SATA physical link, it shall transmit a SATA_HOLD when it is only capable of receiving 21 more dwords.

NOTE 3 - SATA requires that frame transmission cease and SATA_HOLDA be transmitted within 20 dwords of receiving SATA_HOLD. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

When a SATA host port in an STP/SATA bridge is transmitting a frame to a SATA physical link, it shall transmit no more than 19 data dwords after receiving SATA_HOLD.

NOTE 4 - SATA assumes that once a SATA_HOLD is transmitted, frame transmission ceases and SATA_HOLDA arrives within 20 dwords. Since the SATA physical link has non-zero propagation time, one dword of margin is included.

Figure 19 shows STP flow control between:

a) an STP initiator port receiving a frame;

b) an expander device (the first expander device);
c) an expander device with an STP/SATA bridge (the second expander device); and
d) a SATA device port transmitting a frame.

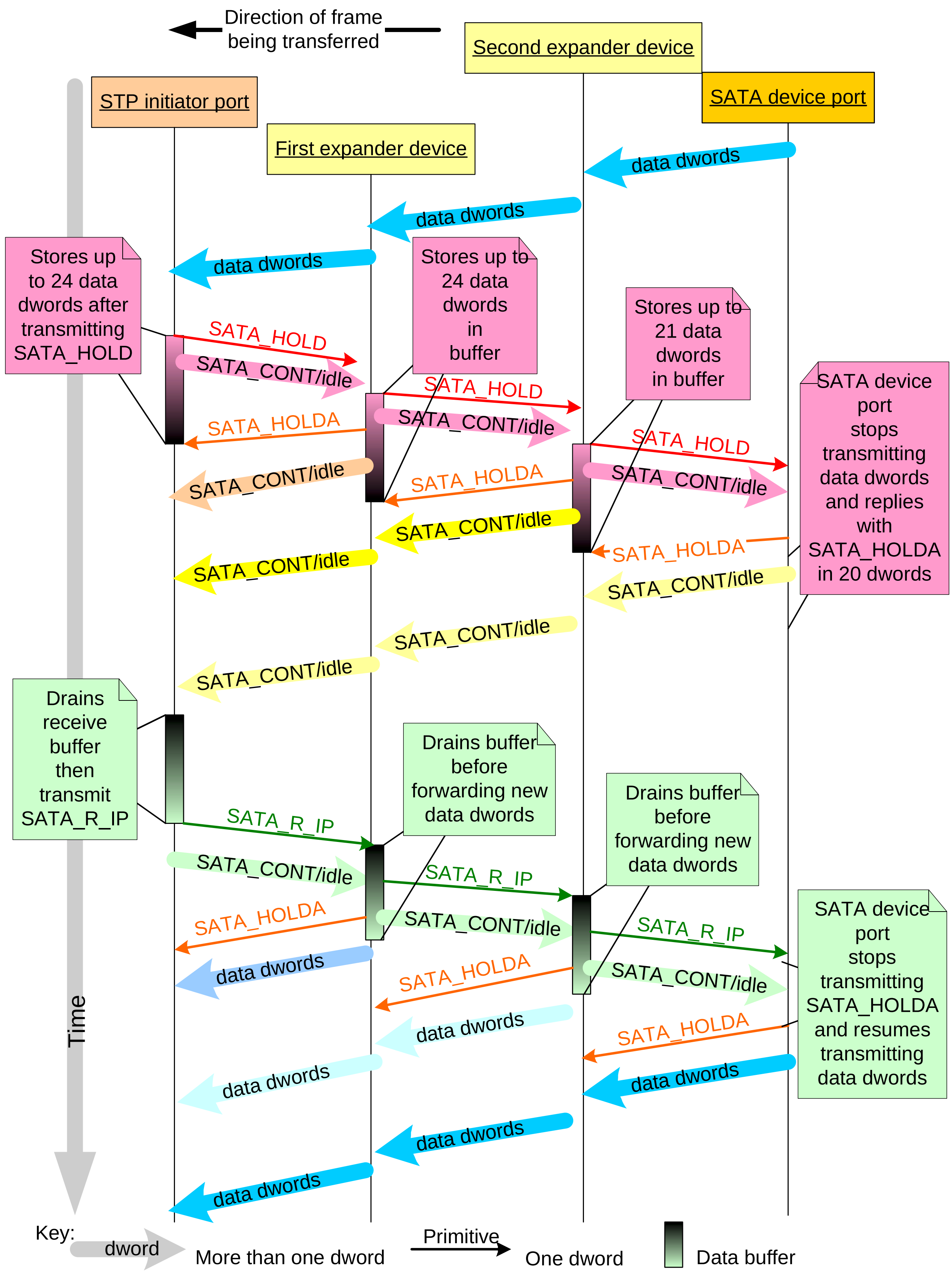

Figure 19 — STP flow control

After the STP initiator port transmits SATA_HOLD, it receives a SATA_HOLDA reply from the first expander device within 24 dwords (for a 1,5 Gbps physical link). The first expander device transmits SATA_HOLD to the second expander device and receives SATA_HOLDA within 24 dwords (for a 1,5 Gbps physical link), buffering data dwords it is no longer able to forward to the STP initiator port. The second expander device transmits SATA_HOLD to the SATA device port and receives SATA_HOLDA within 21 dwords (for a SATA physical link), buffering data dwords it is no longer able to forward to the first expander device. When the SATA device port stops transmitting data dwords, its previous data dwords are stored in the buffers in both expander devices and the STP initiator port.

After the STP initiator port drains its buffer and transmits SATA_R_IP, it receives data dwords from the first expander device's buffer, followed by data dwords from the second expander device's buffer, followed by data dwords from the SATA device port.

0.9.3 Affiliations

Coherent access to the SATA task file registers shall be provided for each STP initiator port. If the STP target port provides one set of task file registers for each STP initiator port then it shall not implement affiliations. If the STP target port provides access to one set of task file registers for all STP initiator ports, then it shall implement affiliations to provide coherency.

An affiliation is a state entered by an STP target port where it refuses to accept connection requests from STP initiator ports other than the one that has established an affiliation.

An STP target port that supports affiliations shall establish an affiliation whenever it accepts a connection request. When an affiliation is established, the STP target port shall reject all subsequent connection requests from other STP initiator ports with OPEN REJECT (STP RESOURCES BUSY).

An STP target port shall maintain an affiliation until any of the following occurs:

a) Power on;
b) the SAS target device receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of HARD RESET ~~(see 10.4.3.9)~~(see 10.4.3.9) from any SMP initiator port;
c) the SAS target device receives an SMP PHY CONTROL request specifying the phy with the affiliation and specifying a phy operation of CLEAR AFFILIATION from the same SAS initiator port that has the affiliation;
d) A connection to the phy with the affiliation is closed with CLOSE (CLEAR AFFILIATION); or
e) the STP target port is part of a STP/SATA bridge and a link reset sequence is begun on the SATA physical link.

An affiliation established when the command is transmitted should be kept until all frames for the command have been delivered. This avoids confusing the SATA device, which only knows about one SATA host. STP initiator ports may keep affiliations for longer tenures, but this is discouraged.

An STP target that implements affiliations shall implement one affiliation per port. Multiple phys on the same STP target port shall use the same affiliation. Support for affiliations is indicated in the SMP REPORT SATA PHY function response ~~(see 10.4.3.6)~~(see 10.4.3.6).

Only one connection between a wide STP initiator port and a wide STP target port shall be allowed at one time. The STP target port shall reject a second connection request from the same STP initiator port with OPEN_REJECT (STP RESOURCES BUSY).

0.9.4 Opening an STP connection

If no STP connection exists when the SATA host port in an STP/SATA bridge receives a SATA_X_RDY from the attached SATA device port, the STP target port in the STP/SATA bridge shall establish an STP connection to the appropriate STP initiator port before it transmits a SATA_R_RDY to the SATA device.

0.9.5 Closing an STP connection

Either STP port (i.e., either the STP initiator port or the STP target port) may originate closing an STP connection. An STP port shall not originate closing an STP connection after sending a SATA_X_RDY or

SATA_R_RDY until after both sending and receiving SATA_SYNC. An STP port shall transmit CLOSE after receiving a CLOSE if it has not already transmitted CLOSE.

When an STP initiator port closes an STP connection, it shall transmit a CLOSE (NORMAL) or CLOSE (CLEAR AFFILIATION~~)~~). When an STP target port closes an STP connection, it shall transmit a CLOSE (NORMAL).

An STP initiator port may issue CLOSE (CLEAR AFFILIATION) in place of a CLOSE (NORMAL) to cause the STP target port to clear the affiliation (see 0.9.3) along with closing the connection. If an STP target port receives CLOSE (CLEAR AFFILIATION), the STP target port shall clear the affiliation for the STP initiator port that sent the CLOSE (CLEAR AFFILIATION).

See 0.4.8 for additional details on closing connections.

An STP/SATA bridge shall break an STP connection if its SATA host phy loses dword synchronization (see 0.4.7).

0.9.6 STP connection management examples

The STP/SATA bridge adds the outgoing OPEN address frames and CLOSEs so the STP initiator port sees an STP target port. The STP/SATA bridge removes incoming OPEN address frame and CLOSEs so the SATA device port sees only a SATA host port. While the connection is open, the STP/SATA bridge passes through all dwords without modification. Both STP initiator port and STP target port use SATA, with SATA flow control (see 0.9.2), while the connection is open.

Figure 20 shows an STP initiator port opening a connection, transmitting a single SATA frame, and closing the connection.

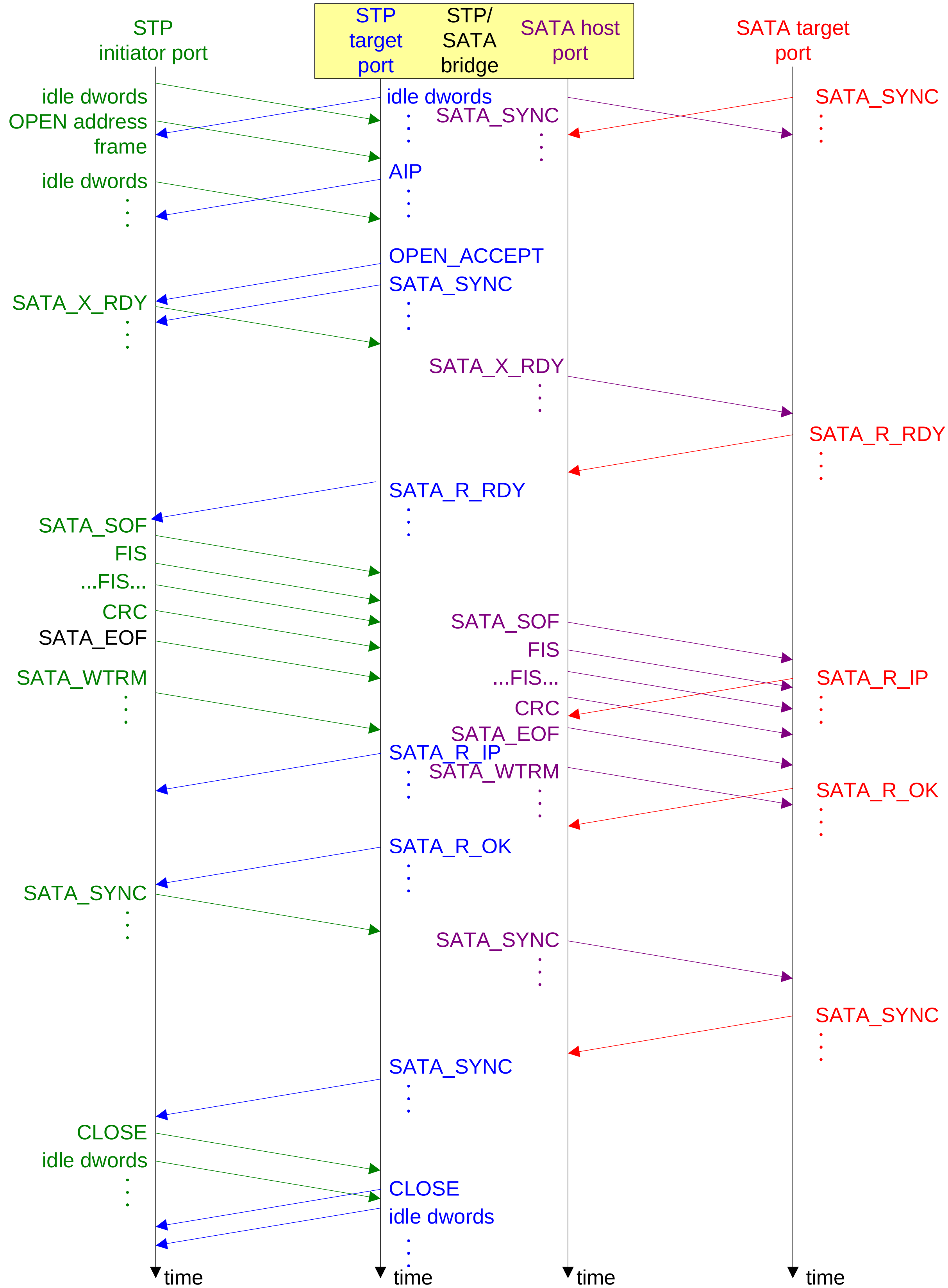

Figure 20 — STP initiator port opening an STP connection

Figure 21 shows a SATA device transmitting a SATA frame. In this example, the STP target port in the STP/SATA bridge opens a connection to an STP initiator port to send just one frame, then closes the connection.

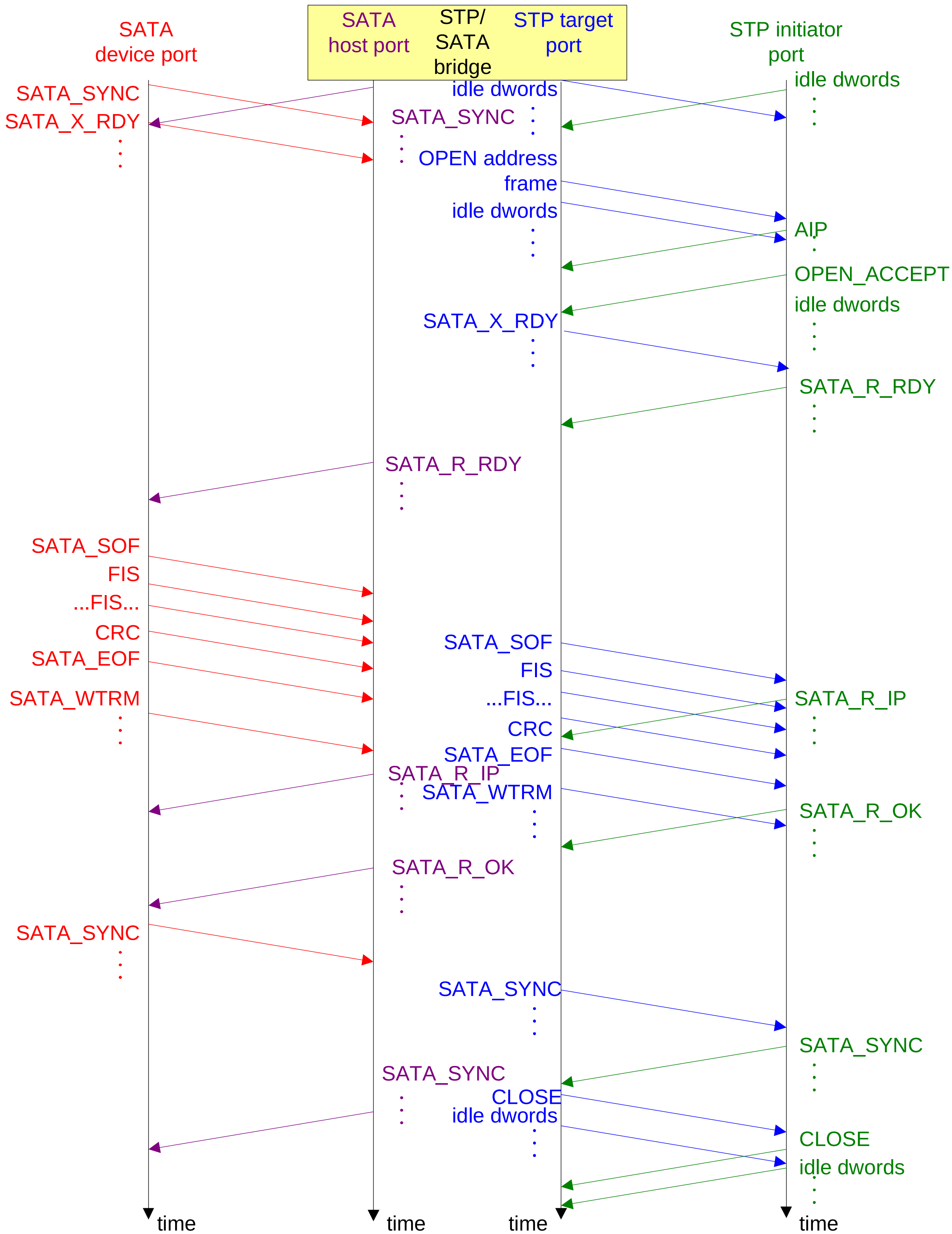

Figure 21 — STP target port opening an STP connection

0.9.7 STP (link layer for STP phys) state machines

The STP link layer uses the SATA link layer state machines (see ATA/ATAPI-7 V3), modified to:

a) communicate with the port layer rather than directly with the transport layer;
b) interface with the ~~SL~~ SL_CC state machine for connection management (e.g., to select when to open and close STP connections, and to tolerate idle dwords between an OPEN address frame or an OPEN_ACCEPT and the first SATA primitive); and
c) implement affiliations (see 0.9.3).

These modifications are not described in this standard.

0.9.8 SMP target port support

A SAS device that contains an STP target port shall also contain an SMP target port.

0.10 SMP link layer

0.10.1 SMP frame transmission and reception

Inside an SMP connection, the source device transmits a single SMP_REQUEST frame and the destination device responds with a single SMP_RESPONSE frame ~~(see 9.4)~~(see 9.4).

Frames are surrounded by SOF and EOF as shown in figure 22. There is no acknowledgement of SMP frames with ACK and NAK. There is no credit exchange with RRDY.

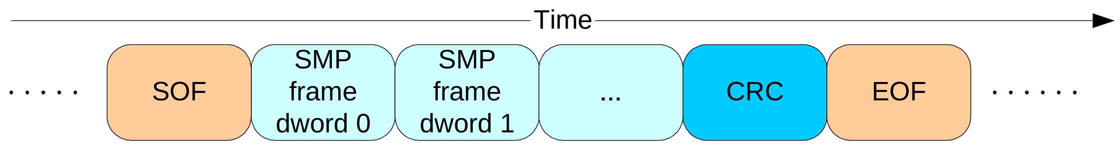

Figure 22 — SMP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 7.5). The SMP link layer state machine checks that the frame is not too short and that the CRC is valid (see 0.10.4).

0.10.2 SMP flow control

By accepting an SMP connection, the destination device indicates it is ready to receive one SMP_REQUEST frame.

After the source device transmits one SMP_REQUEST frame, it shall be ready to receive one SMP_RESPONSE frame.

0.10.3 Closing an SMP connection

After receiving the SMP_RESPONSE frame, the source device shall transmit a CLOSE (NORMAL) to close the connection.

After transmitting the SMP_RESPONSE frame, the destination device shall reply with a CLOSE (NORMAL).

See 0.4.8 for additional details on closing connections.

0.10.4 SMP (link layer for SMP phys) state machines

0.10.4.1 SMP state machines overview

The SMP state machines control the flow of dwords on the physical link during an SMP connection. The SMP state machines are as follows:

a) SMP_IP (link layer for SMP initiator phys) state machine (see 0.10.4.3); and
b) SMP_TP (link layer for SMP target phys) state machine (see 0.10.4.4).

0.10.4.2 SMP transmitter and receiver

The SMP transmitter receives the following messages from the SMP state machines:

a) Transmit Idle Dword; and
b) Transmit Frame.

The SMP transmitter sends the following messages to the SMP state machines:

a) Frame Transmitted.

The SMP receivers sends the following messages to the SMP state machines:

a) SOF Received;
b) Dword Received; and
c) EOF Received;

0.10.4.3 SMP_IP (link layer for SMP initiator phys) state machine

0.10.4.3.1 SMP_IP state machine overview

The SMP_IP state machine's function is to transmit an SMP request frame and then receive the corresponding response frame. This state machine consists of the following states:

a) SMP_IP1:Idle (see 0.10.4.3.2)(initial state);
b) SMP_IP2:Transmit_Frame (see 0.10.4.3.3);
c) SMP_IP3:Receive_Frame (see 0.10.4.3.4).

The SMP_IP state machine shall start in the SMP_IP1:Idle state on receipt of an Enable Disable SMP (Enable) message from the ~~SL~~ SL_CC state machine (see 0.6).

The SMP_IP state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the ~~SL~~ SL_CC state machine.

Figure 23 shows the SMP_IP state machine.

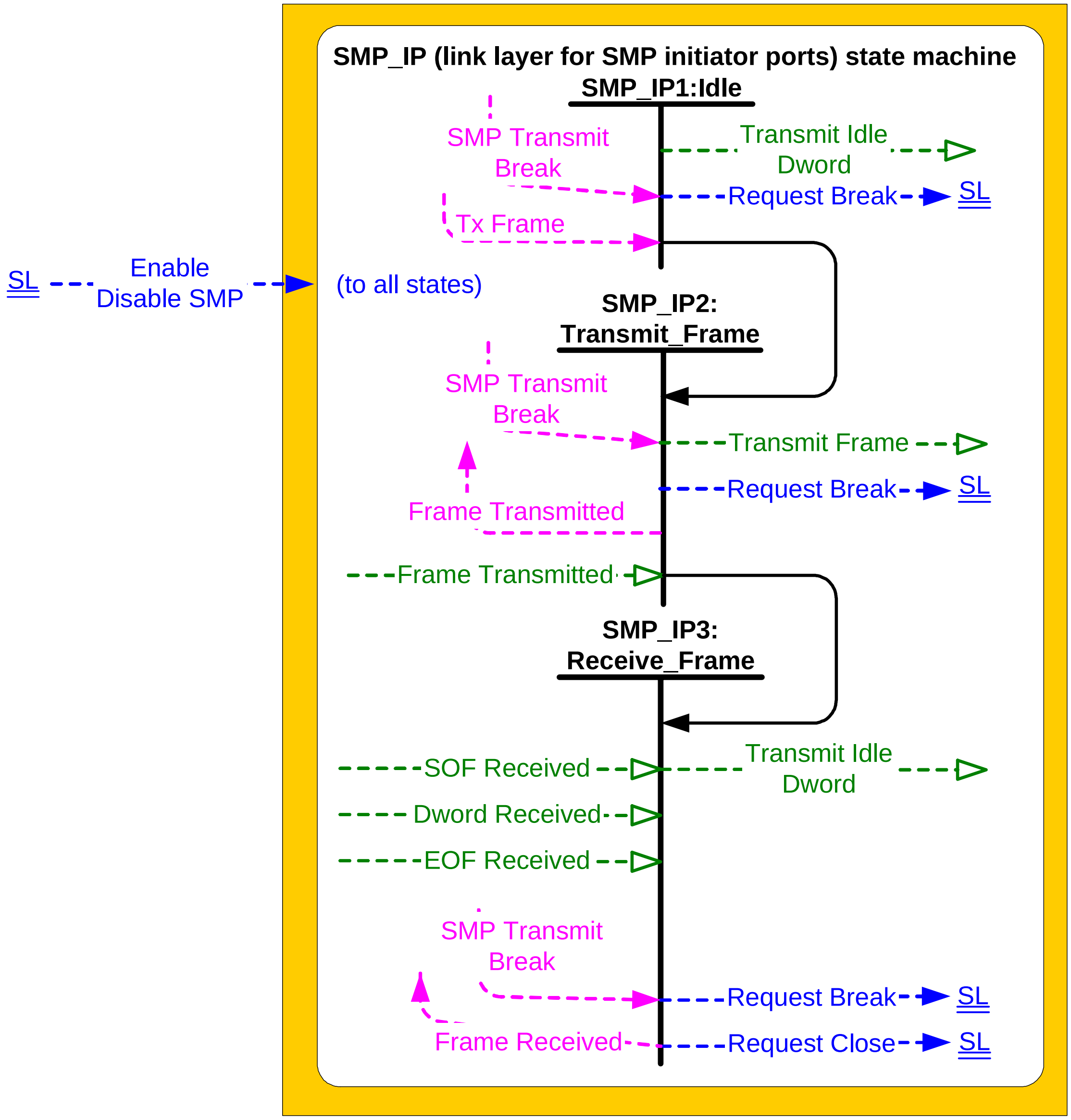

Figure 23 — SMP_IP (link layer for SMP initiator phys) state machine

0.10.4.3.2 SMP_IP1:Idle state

0.10.4.3.2.1 State description

This state is the initial state.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

If an SMP Transmit Break request is received, this state shall send a Request Break message to the ~~SL~~ SL_CC state machine (see 0.6).

0.10.4.3.2.2 Transition SMP_IP1:Idle to SMP_IP2:Transmit_Frame

This transition shall occur after a Tx Frame (SMP) request is received.

0.10.4.3.3 SMP_IP2:Transmit_Frame state

0.10.4.3.3.1 State description

This state shall send a Transmit Frame message to the SMP transmitter.

If an SMP Transmit Break request is received, this state shall send a Request Break message to the ~~SL~~ SL_CC state machine (see 0.6) and terminate.

After the Frame Transmitted message is received, this state shall send a Frame Transmitted confirmation to the port layer.

0.10.4.3.3.2 Transition SMP_IP2:Transmit_Frame to SMP_IP3:Receive_Frame

This transition shall occur after sending a Frame Transmitted confirmation to the port layer.

0.10.4.3.4 SMP_IP3:Receive_Frame state

This state checks the SMP response frame and determines if the SMP response frame was successfully received (e.g., no CRC error).

If the SMP response frame is received with a CRC error, this state shall send a Frame Received (SMP Failure) confirmation to the port layer.

If the number of dwords between the SOF and EOF of the SMP response frame is less than 2, or the number of dwords after an SOF is greater than 258, this state shall send a Frame Received (SMP Failure) confirmation to the port layer. If the SMP response frame is received with no CRC error and the SMP response frame is valid, this state shall:

a) send a Frame Received (SMP) confirmation to the port layer; and
b) send a Request Close message to the ~~SL~~ SL_CC state machine (see 0.6).

If an SMP Transmit Break request is received, this state shall send a Request Break message to the ~~SL~~ SL_CC state machine and terminate.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

0.10.4.4 SMP_TP (link layer for SMP target ports) state machine

0.10.4.4.1 SMP_TP state machine overview

The SMP_TP state machine's function is to receive an SMP request frame and then transmit the corresponding SMP response frame. The SMP_TP state machine consists of the following states:

a) SMP_TP1:Receive_Frame (see 0.10.4.4.2)(initial state); and
b) SMP_TP2:Transmit_Frame (see 0.10.4.4.3).

The SMP_TP state machine shall start in the SMP_TP1:Receive_Frame state after receiving an Enable Disable SMP (Enable) message from the ~~SL~~ SL_CC state machine (see 0.6).

The SMP_TP state machine shall terminate after receiving an Enable Disable SMP (Disable) message from the ~~SL~~ SL_CC state machine.

Figure 24 shows the SMP_TP state machine.

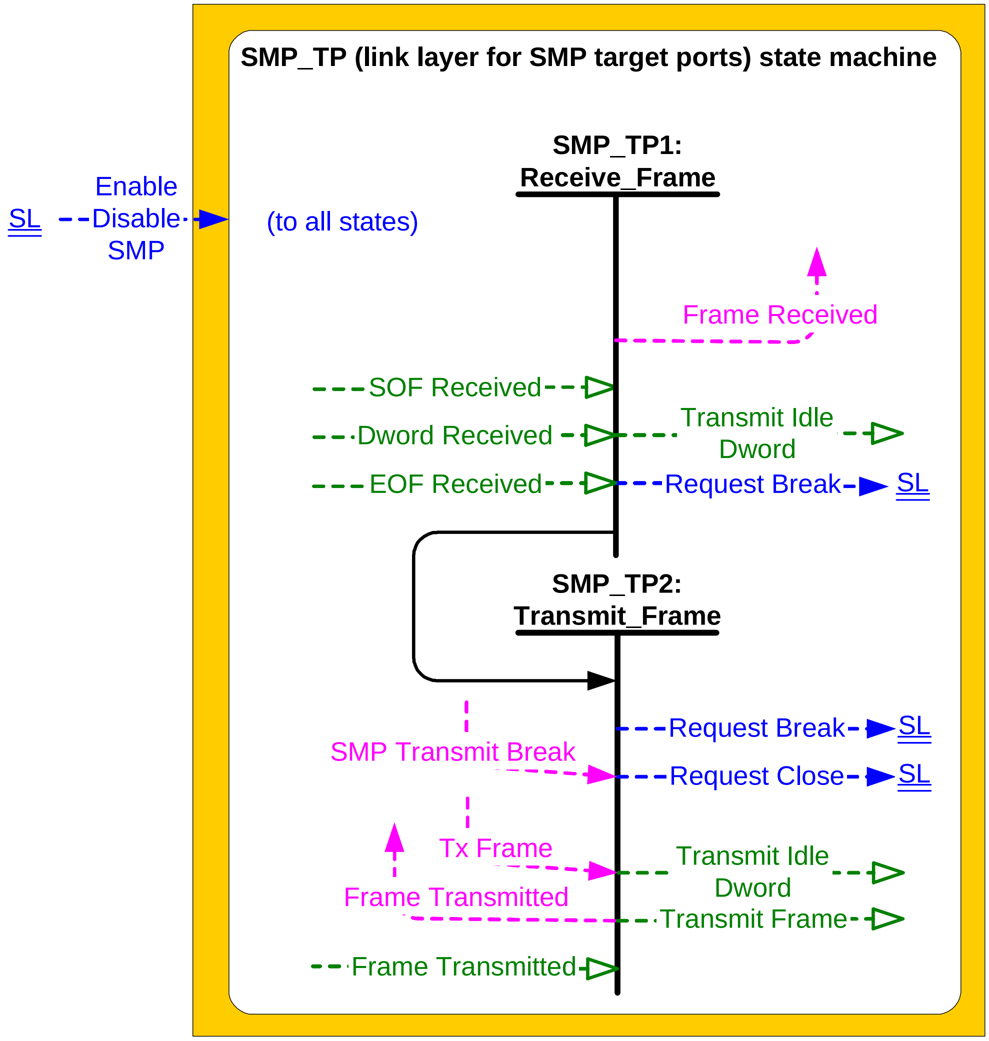

Figure 24 — SMP_TP (link layer for SMP target phys) state machine

0.10.4.4.2 SMP_TP1:Receive_Frame state

0.10.4.4.2.1 State description

This state waits for an SMP frame and determines if the SMP frame was successfully received (e.g., no CRC error).

If an SMP frame is received, this state shall send a Request Break message to the ~~SL~~ SL_CC state machine (see 0.6) and terminate if:

a) the SMP frame has a CRC error;
b) the number of data dwords between the SOF and EOF is less than 2; or
c) the number of data dwords after the SOF is greater than 258.

Otherwise, this state shall send a Frame Received (SMP) confirmation to the port layer.

This state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.

0.10.4.4.2.2 Transition SMP_TP1:Receive_Frame to SMP_TP2:Transmit_Frame

This transition shall occur after sending a Frame Received (SMP) confirmation to the port layer.

0.10.4.4.3 SMP_TP2:Transmit_Frame state

If this state receives an SMP Transmit Break request, this state shall send a Request Break message to the ~~SL~~ SL_CC state machine and terminate.

If this state receives a Tx Frame (SMP) request, this state shall send a Transmit Frame message to the SMP transmitter; then wait for a Frame Transmitted message. After receiving a Frame Transmitted message, this state shall send a Request Close message to the ~~SL~~ SL_CC state machine (see 0.6) and terminate.

After sending Transmit Frame message to the SMP transmitter, this state shall request idle dwords be transmitted by repeatedly sending Transmit Idle Dword messages to the SMP transmitter.