To: T10 Technical Committee
From: Timothy Hoglund, LSI Logic
Date: 28 February 2006
Subject: Serial Attached SCSI - 2 (SAS-2)

Revision History

Revision 0 (28 February 2008) first revision

Related Documents

sas2r02	Serial Attached SCSI 2.0 revision 2
T10/05-040r0	SAS-1.1: Break_Wait handling
T10/05-086r0	SAS-1.1: Link layer timeout race conditions
T10/05-093r1	SAS-1.1: Responding to an OPEN address frame in the BreakWait state
T10/05-145r0	SAS-1.1: BreakWait issue resolution

Overview

This proposal seeks to remove link layer race conditions present within the SAS-1.1 specification, namely the potential for BREAKs to cross on the wire with either OPEN_REJECTs or CLOSEs leading to potential livelock scenarios involving Open and Break Timeouts.

These link layer race conditions stem from the fact that the SL and XL state machines use the BREAK primitive sequence indiscriminately, i.e. the BREAK primitive sequence is used by SL_CC5:BreakWait and XL10:Break_Wait to originate the abandoning of a connection request (or breaking of a connection) and by SL_CC6:Break and XL9:Break to respond to a received BREAK primitive sequence.

Because the same primitive sequence is used to originate and respond, the SL and XL idle states have been made insensitive to BREAK primitive sequence received to avoid deadlock loops. This insensitivity (in the SL or XL idle state) to a received BREAK primitive sequence however leads to link layer race conditions because a received BREAK primitive sequence can be ignored causing the two link state machines on either end of a physical link to become out-of-sync with each other.

The specific link layer race conditions that this proposal seeks to remove are shown in Figure 1 and Figure 2.

Proposal T10/05-145 provides a description of various other approaches that have been suggested to resolve this race condition.

In May, 2005 the SAS Protocol Working Group reviewed T10/145r0 and provided the following direction:

1) remove potentially counterproductive approaches to addressing these race conditions; and
2) wait for SAS-2 to incorporate a request/response model for BREAKs.

It is the intention of this proposal to provide a suitable request/response model for BREAKs into SAS-2.

Figure 1 below (from proposal T10/05-086r0) demonstrates the potential of BREAK and OPEN_REJECT crossing leading to two phys ping-ponging OPENs and BREAKs forever.

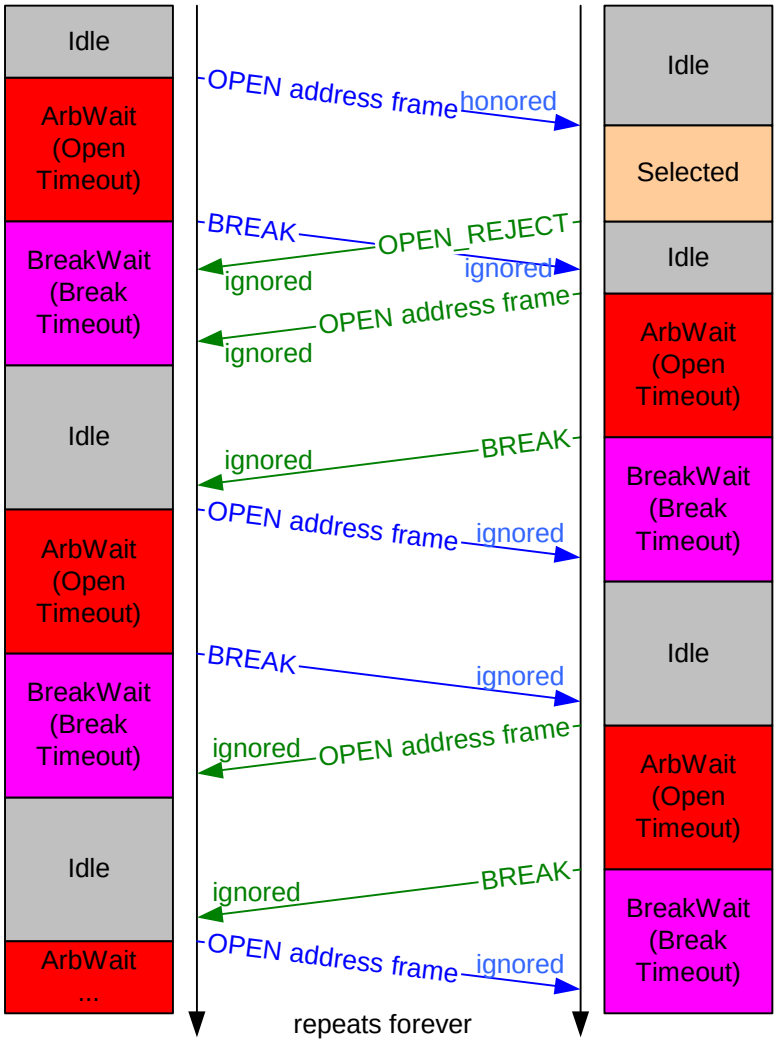

Figure 1 — BREAK crossing OPEN_REJECT

Figure 2 below (from proposal T10/05-086r0) demonstrates the potential of BREAK and OPEN_REJECT crossing leading to two phys ping-ponging OPENs and BREAKs forever.

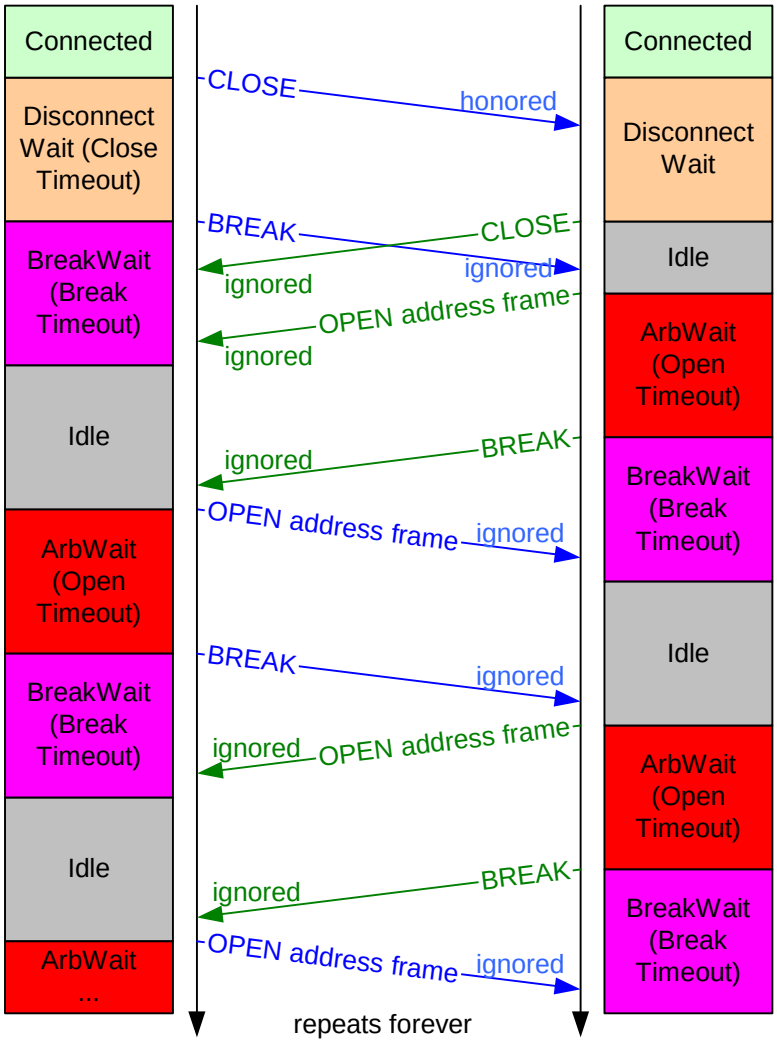

Figure 2 — BREAK crossing OPEN_REJECT

[Start of proposed modifications to sas2r02 text, tables, diagrams]

4 General

4.1 Architecture

4.1.1 Architecture overview

...

4.1.2 Physical links and phys

...

Each phy has:

a) a SAS address (see 4.2.2), inherited from the SAS port (see 4.1.3) or expander device;
b) a phy identifier (see 4.2.7) which is unique within the device;
c) optionally, support for being an SSP initiator phy;
d) optionally, support for being an STP initiator phy;
e) optionally, support for being an SMP initiator phy;
f) optionally, support for being an SSP target phy;
g) optionally, support for being an STP target phy; and
h) optionally, support for being an SMP target phy.

During the identification sequence (see 7.9), a phy:

a) transmits an IDENTIFY address frame including the device type (i.e., end device, edge expander device, or fanout expander device) of the device containing the phy, the SAS address of the SAS port or expander device containing the phy, the device name of the SAS device or expander device containing the phy, the phy identifier, the bit specifying phy BREAK_REPLY capability, and bits specifying the SSP initiator phy capability, STP initiator phy capability, SMP initiator phy capability, SSP target phy capability, STP target phy capability, and SMP target phy capability.
b) receives an IDENTIFY address frame containing the same set of information from the attached phy, including the attached device type, the attached SAS address, the attached device name, the attached phy identifier, the bit specifying the attached phy BREAK_REPLY capability, and bits indicating the attached SSP initiator phy capability, attached STP initiator phy capability, attached SMP initiator phy capability, attached SSP target phy capability, attached STP target phy capability, and attached SMP target phy capability.

...

4.1.3 Ports (narrow ports and wide ports)

...

Phys that are able to become part of the same wide port shall set the DEVICE TYPE field, PHY BREAK_REPLY CAPABLE bit, SSP INITIATOR PORT bit, STP INITIATOR PORT bit, SMP INITIATOR PORT bit, SSP TARGET PORT bit, STP TARGET PORT bit, SMP TARGET PORT bit, and SAS ADDRESS field in the IDENTIFY address frame (see 7.8.2) transmitted during the identification sequence to the same set of values on each phy in the wide port. Recipient wide ports are not required to check the consistency of these fields across their phys.

4.3 State machines

4.3.1 State machine overview

4.3.2 Transmit data path

Figure 32 shows the transmit data path in a SAS phy.

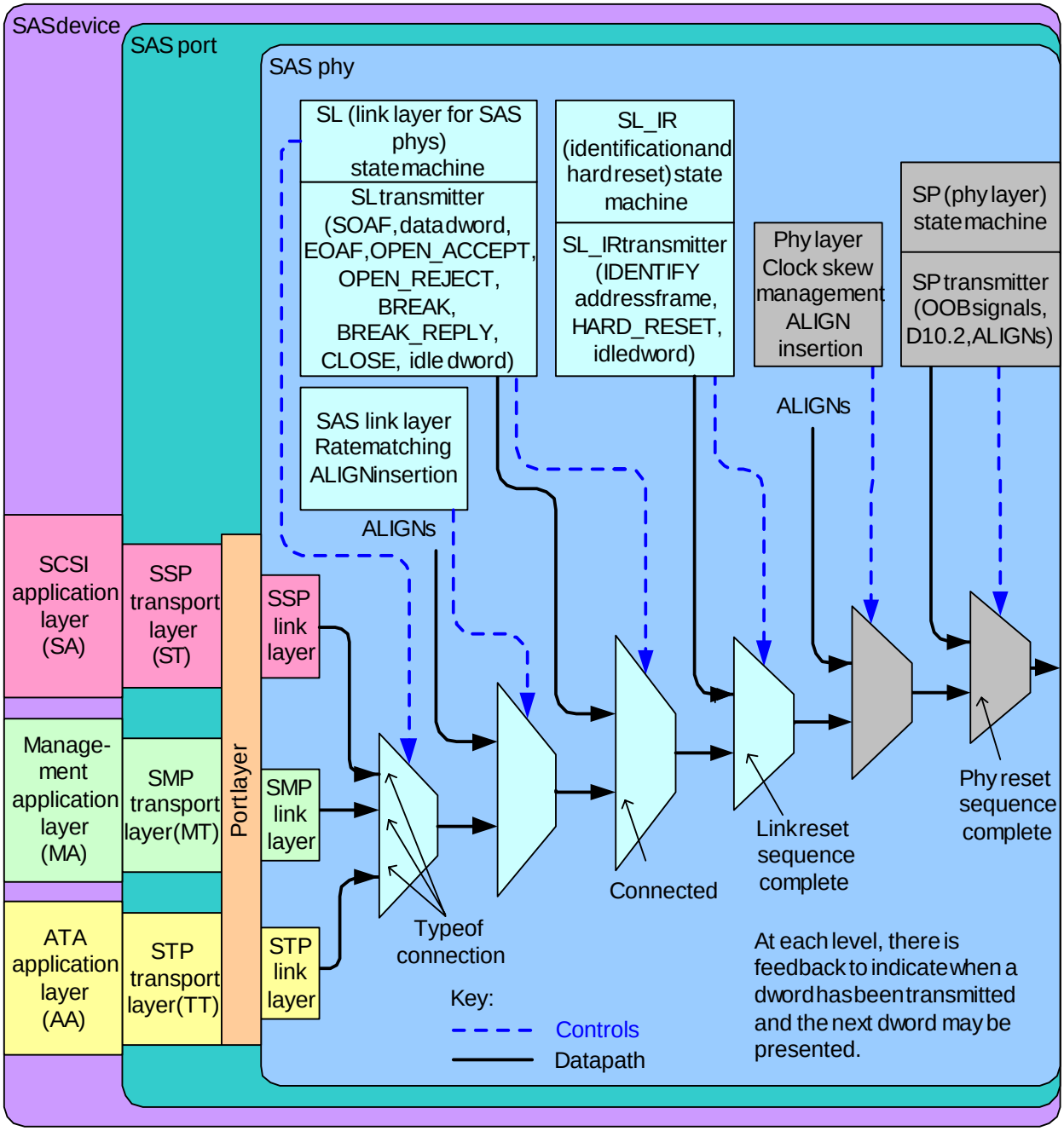

Figure 32 — Transmit data path in a SAS phy

Figure 36 shows the transmit data path in an expander phy.

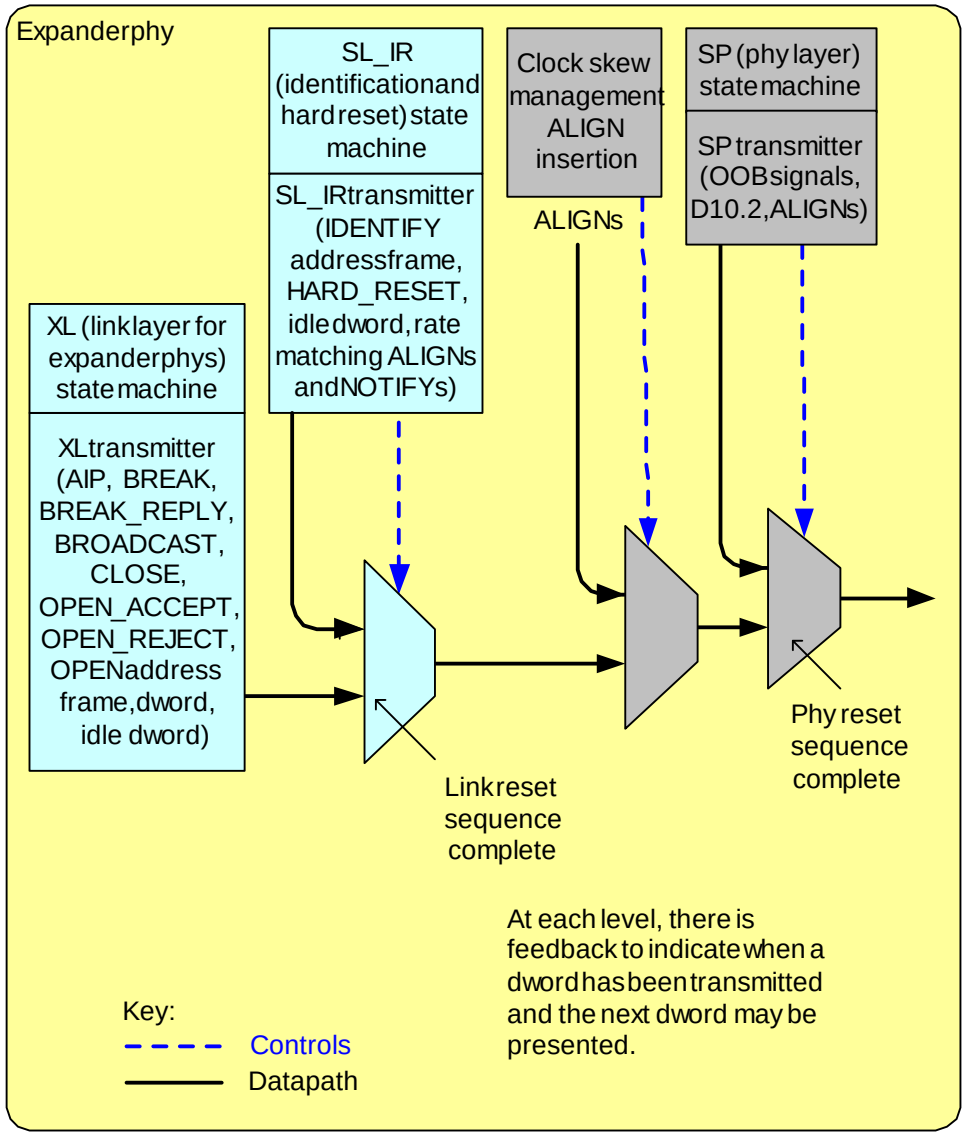

Figure 36 — Transmit data path and state machines in an expander phy

4.9 Phy event information

...

The PHY EVENT INFORMATION SOURCE field, defined in table 22, is used in the Protocol-Specific Port log page (see 10.2.8.1), the REPORT PHY EVENT INFORMATION function (see 10.4.3.9), and the CONFIGURE PHY EVENT INFORMATION function (see 10.4.3.14) indicates the type of phy event information being reported in the accompanying PHY EVENT INFORMATION field.

Table 22 — PHY EVENT INFORMATION SOURCE **field** (part 1 of 4)

Code	Name	Type [a]	Description
00h	No event	N/A	No event. The PHY EVENT INFORMATION field is not valid.
01h	Invalid dword count	WC	Number of invalid dwords (see 3.1.101) that have been received outside of phy reset sequences (i.e., between when the SP state machine (see 6.8) sends a Phy Layer Ready (SAS) confirmation or Phy Layer Ready (SATA) confirmation and when it sends a Phy Layer Not Ready confirmation to the link layer)
02h	Running disparity error count	WC	Number of dwords containing running disparity errors (see 6.2) that have been received outside of phy reset sequences
03h	Loss of dword synchronization count	WC	Number of times the phy has restarted the link reset sequence because it lost dword synchronization (i.e., the SP state machine transitioned from SP15:SAS_PHY_Ready or SP22:SATA_PHY_Ready to SP0:OOB_COMINIT (see 6.8))
04h	Phy reset problem count	WC	Number of times the phy did not obtain dword synchronization during the final SAS speed negotiation window (see 6.7.4.2)
05h	Elasticity buffer overflow count	WC	Number of times the phy's receive elasticity buffer (see 7.3) has overflowed (e.g., because it did receive a sufficient number of ALIGNs and/or NOTIFYs)
06h	Received ERROR count	WC	Number of times the phy received an ERROR primitive
07h - 1Fh	Reserved for phy layer-based phy event information		
20h	Received address frame error count	WC	Number of times the phy detected an invalid address frame (see 7.8) (e.g., because of a CRC error)
21h	Received OPEN_REJECT abandon count	WC	Number of times the phy transmitted an OPEN address frame and received an abandon-class OPEN_REJECT (see 7.2.5.11). In expander devices, forwarded OPEN_REJECTs shall not be counted.
22h	Received OPEN_REJECT retry count	WC	Number of times the phy transmitted an OPEN address frame and received a retry-class OPEN_REJECT (see 7.2.5.11). In expander devices, forwarded OPEN_REJECTs shall not be counted.

[a] The Type column indicates the source type:
a) WC = wrapping counter
b) PVD = peak value detector
c) N/A = not applicable

Table 22 — PHY EVENT INFORMATION SOURCE **field** (part 2 of 4)

Code	Name	Type [a]	Description
23h	Transmitted OPEN_REJECT abandon count	WC	Number of times the phy received an OPEN address frame and transmitted an abandon-class OPEN_REJECT (see 7.2.5.11). In expander devices, forwarded OPEN_REJECTs shall not be counted.
24h	Transmitted OPEN_REJECT retry count	WC	Number of times the phy received an OPEN address frame and transmitted a retry-class OPEN_REJECT (see 7.2.5.11). In expander devices, forwarded OPEN_REJECTs shall not be counted.
25h	Received AIP (WAITING ON PARTIAL) count	WC	Number of times the phy received an AIP (WAITING ON PARTIAL) or AIP (RESERVED WAITING ON PARTIAL). In expander devices, forwarded AIPs shall be counted.
26h	Received AIP (WAITING ON CONNECTION) count	WC	Number of times the phy received an AIP (WAITING ON CONNECTION). In expander devices, forwarded AIPs shall be counted.
27h	Received BREAK count	WC	Number of times the phy received a BREAK or BREAK_REPLY that was not a response to a BREAK that it transmitted
28h	Transmitted BREAK count	WC	Number of times the phy transmitted a BREAK that was not a response to a BREAK it received (e.g., a Close Timeout was detected by the SL state machine interfacing to the SMP target port).
29h	Break Timeout count	WC	Number of times the phy transmitted a BREAK and did not receive a BREAK or BREAK_REPLY in response (e.g., as detected by the XL state machine and/or the SL state machine interfacing to the SMP target port).
2Ah	Connection count	WC	Number of connections in which the phy was involved
2Bh	Peak transmitted pathway blocked count	PVD	Peak value of a PATHWAY BLOCKED field in an OPEN address frame transmitted by the phy. Since the maximum value of the PATHWAY BLOCKED field is FFh, only byte 3 of the PHY EVENT INFORMATION field is used.
2Ch	Peak transmitted arbitration wait time	PVD	Peak value of an ARBITRATION WAIT TIME field in an OPEN address frame transmitted by the phy. Since the maximum value of the PATHWAY BLOCKED field is FFFFh, only bytes 2 and 3 of the PHY EVENT INFORMATION field are used.
2Dh	Peak arbitration time	PVD	Peak time in microseconds after transmitting an OPEN address frame that the phy has waited for connection response (e.g., OPEN_ACCEPT or OPEN_REJECT).
2Eh	Peak connection time	PVD	The peak duration, in microseconds, of any connection in which the phy was involved.
2Fh - 3Fh	Reserved for SAS arbitration-related phy information		

[a] The Type column indicates the source type:
a) WC = wrapping counter
b) PVD = peak value detector
c) N/A = not applicable

Table 22 — PHY EVENT INFORMATION SOURCE **field** (part 3 of 4)

Code	Name	Type [a]	Description
40h	Transmitted SSP frame count	WC	Number of SSP frames transmitted
41h	Received SSP frame count	WC	Number of SSP frames received
42h	Received SSP frame error count	WC	Number of times the phy was used in a connection involving the SSP target port, detected an invalid frame, and transmitted a NAK (CRC ERROR) (e.g., because of a CRC error)
43h	Transmitted SSP frame error count	WC	Number of times the phy was used in a connection involving the SSP target port, transmitted a frame, and received a NAK or an ACK/NAK timeout
44h	Transmitted CREDIT_BLOCKED count	WC	Number of times the phy transmitted a CREDIT_BLOCKED
45h	Received CREDIT_BLOCKED count	WC	Number of times the phy received a CREDIT_BLOCKED
46h - 4Fh	Reserved for SSP-related phy event information		
50h	Transmitted SATA frame count	WC	Number of STP or SATA frames transmitted
51h	Received SATA frame count	WC	Number of STP or SATA frames received
52h	SATA flow control buffer overflow count	WC	Number of times the phy's STP flow control buffer (see 7.17.3) has overflowed (e.g., because it received more data dwords than allowed after transmitting HOLD during an STP connection). This count should be maintained in the phy transmitting the HOLD and receiving the data dwords, but may be maintained in the phy receiving the HOLD and transmitting the data dwords.
53h - 5Fh	Reserved for STP and SATA-related phy event information		
60h	Transmitted SMP frame count	WC	Number of SMP frames transmitted
61h	Received SMP frame count	WC	Number of SMP frames received
62h	Receive SMP frame error count	WC	Number of times the phy was used for to access the SMP target port and the SMP target port detected an invalid frame and transmitted a BREAK (e.g., because of a CRC error).

[a] The Type column indicates the source type:
a) WC = wrapping counter
b) PVD = peak value detector
c) N/A = not applicable

Table 22 — PHY EVENT INFORMATION SOURCE **field** (part 4 of 4)

Code	Name	Type [a]	Description
63h - 6Fh	Reserved for STP-related phy event information		
70h - CFh	Reserved		
D0h - FFh	Vendor specific		

[a] The Type column indicates the source type:
a) WC = wrapping counter
b) PVD = peak value detector
c) N/A = not applicable

7 Link layer

7.1 Link layer overview

7.2 Primitives

7.2.1 Primitives overview

7.2.2 Primitive summary

Table 71defines the primitives not specific to the type of connection.

Table 71— Primitives not specific to type of connection (part 1 of 2)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
AIP (NORMAL)	NoConn		E		I	E	T	Single
AIP (RESERVED 0)	NoConn				I	E	T	Single
AIP (RESERVED 1)	NoConn				I	E	T	Single
AIP (RESERVED 2)	NoConn				I	E	T	Single
AIP (RESERVED WAITING ON PARTIAL)	NoConn				I	E	T	Single
AIP (WAITING ON CONNECTION)	NoConn		E		I	E	T	Single
AIP (WAITING ON DEVICE)	NoConn		E		I	E	T	Single
AIP (WAITING ON PARTIAL)	NoConn		E		I	E	T	Single
ALIGN (0)	All	I	E	T	I	E	T	Single
ALIGN (1)	All	I	E	T	I	E	T	Single
ALIGN (2)	All	I	E	T	I	E	T	Single
ALIGN (3)	All	I	E	T	I	E	T	Single
BREAK	All	I	E	T	I	E	T	Redundant
BREAK_REPLY	All	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I	E	T	Redundant
BROADCAST (SES)	NoConn			T	I	E	T	Redundant
BROADCAST (EXPANDER)	NoConn		E		I	E	T	Redundant
BROADCAST (RESERVED 2)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 3)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 4)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 0)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 1)	NoConn				I	E	T	Redundant
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T	I		T	Triple
CLOSE (RESERVED 0)	Conn				I		T	Triple
CLOSE (RESERVED 1)	Conn				I		T	Triple
EOAF	NoConn	I	E	T	I	E	T	Single
ERROR	All		E		I	E	T	Single
HARD_RESET	NoConn	I	E		I	E	T	Redundant
NOTIFY (ENABLE SPINUP)	All	I	E				T	Single
NOTIFY (POWER LOSS EXPECTED)	All	I	E				T	Single
NOTIFY (RESERVED 1)	All				I	E	T	Single
NOTIFY (RESERVED 2)	All				I	E	T	Single
OPEN_ACCEPT	NoConn	I		T	I		T	Single

Table 71— Primitives not specific to type of connection (part 2 of 2)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [c]
		I	E	T	I	E	T	
OPEN_REJECT (BAD DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	NoConn	I	E	T	I		T	Single
OPEN_REJECT (NO DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (PATHWAY BLOCKED)	NoConn		E		I		T	Single
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	NoConn	I		T	I		T	Single
OPEN_REJECT (RESERVED ABANDON 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 2)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 3)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 1)	NoConn				I		T	Single
OPEN_REJECT (RETRY)	NoConn	I		T	I		T	Single
OPEN_REJECT (STP RESOURCES BUSY)	NoConn		E	T	I			Single
OPEN_REJECT (WRONG DESTINATION)	NoConn	I		T	I		T	Single
SOAF	NoConn	I	E	T	I	E	T	Single

[a] The Use column indicates when the primitive is used:
a) NoConn: SAS physical links, outside connections;
b) Conn: SAS physical links, inside connections;
c) All: SAS physical links, both outside connections or inside any type of connection; or
d) STP: SAS physical links, inside STP connections.

[b] The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:
a) I for SAS initiator ports;
b) E for expander ports; and
c) T for SAS target ports.
Expander ports are not considered originators of primitives that are passing through from expander port to expander port.

[c] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a continued primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 7.2.4).

7.2.3 Primitive encodings

Table 74 defines the primitive encoding for primitives not specific to type of connection.

Table 74 — Primitive encoding for primitives not specific to type of connection (part 1 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
AIP (NORMAL)	K28.5	D27.4	D27.4	D27.4
AIP (RESERVED 0)	K28.5	D27.4	D31.4	D16.7
AIP (RESERVED 1)	K28.5	D27.4	D16.7	D30.0
AIP (RESERVED 2)	K28.5	D27.4	D29.7	D01.4
AIP (RESERVED WAITING ON PARTIAL)	K28.5	D27.4	D01.4	D07.3
AIP (WAITING ON CONNECTION)	K28.5	D27.4	D07.3	D24.0
AIP (WAITING ON DEVICE)	K28.5	D27.4	D30.0	D29.7
AIP (WAITING ON PARTIAL)	K28.5	D27.4	D24.0	D04.7
ALIGN (0)	K28.5	D10.2	D10.2	D27.3
ALIGN (1)	K28.5	D07.0	D07.0	D07.0
ALIGN (2)	K28.5	D01.3	D01.3	D01.3
ALIGN (3)	K28.5	D27.3	D27.3	D27.3
BREAK	K28.5	D02.0	D24.0	D07.3
BREAK_REPLY	K28.5	D02.0	D29.7	D16.7
BROADCAST (CHANGE)	K28.5	D04.7	D02.0	D01.4
BROADCAST (SES)	K28.5	D04.7	D07.3	D29.7
BROADCAST (EXPANDER)	K28.5	D04.7	D01.4	D24.0
BROADCAST (RESERVED 2)	K28.5	D04.7	D04.7	D04.7
BROADCAST (RESERVED 3)	K28.5	D04.7	D16.7	D02.0
BROADCAST (RESERVED 4)	K28.5	D04.7	D29.7	D30.0
BROADCAST (RESERVED CHANGE 0)	K28.5	D04.7	D24.0	D31.4
BROADCAST (RESERVED CHANGE 1)	K28.5	D04.7	D27.4	D07.3
CLOSE (CLEAR AFFILIATION)	K28.5	D02.0	D07.3	D04.7
CLOSE (NORMAL)	K28.5	D02.0	D30.0	D27.4
CLOSE (RESERVED 0)	K28.5	D02.0	D31.4	D30.0
CLOSE (RESERVED 1)	K28.5	D02.0	D04.7	D01.4
EOAF	K28.5	D24.0	D07.3	D31.4
ERROR	K28.5	D02.0	D01.4	D29.7
HARD_RESET	K28.5	D02.0	D02.0	D02.0

Table 74 — Primitive encoding for primitives not specific to type of connection (part 2 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
NOTIFY (ENABLE SPINUP)	K28.5	D31.3	D31.3	D31.3
NOTIFY (POWER LOSS EXPECTED)	K28.5	D31.3	D07.0	D01.3
NOTIFY (RESERVED 1)	K28.5	D31.3	D01.3	D07.0
NOTIFY (RESERVED 2)	K28.5	D31.3	D10.2	D10.2
OPEN_ACCEPT	K28.5	D16.7	D16.7	D16.7
OPEN_REJECT (BAD DESTINATION)	K28.5	D31.4	D31.4	D31.4
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	K28.5	D31.4	D04.7	D29.7
OPEN_REJECT (NO DESTINATION)	K28.5	D29.7	D29.7	D29.7
OPEN_REJECT (PATHWAY BLOCKED)	K28.5	D29.7	D16.7	D04.7
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	K28.5	D31.4	D29.7	D07.3
OPEN_REJECT (RESERVED ABANDON 0)	K28.5	D31.4	D02.0	D27.4
OPEN_REJECT (RESERVED ABANDON 1)	K28.5	D31.4	D30.0	D16.7
OPEN_REJECT (RESERVED ABANDON 2)	K28.5	D31.4	D07.3	D02.0
OPEN_REJECT (RESERVED ABANDON 3)	K28.5	D31.4	D01.4	D30.0
OPEN_REJECT (RESERVED CONTINUE 0)	K28.5	D29.7	D02.0	D30.0
OPEN_REJECT (RESERVED CONTINUE 1)	K28.5	D29.7	D24.0	D01.4
OPEN_REJECT (RESERVED INITIALIZE 0)	K28.5	D29.7	D30.0	D31.4
OPEN_REJECT (RESERVED INITIALIZE 1)	K28.5	D29.7	D07.3	D16.7
OPEN_REJECT (RESERVED STOP 0)	K28.5	D29.7	D31.4	D07.3
OPEN_REJECT (RESERVED STOP 1)	K28.5	D29.7	D04.7	D27.4
OPEN_REJECT (RETRY)	K28.5	D29.7	D27.4	D24.0
OPEN_REJECT (STP RESOURCES BUSY)	K28.5	D31.4	D27.4	D01.4
OPEN_REJECT (WRONG DESTINATION)	K28.5	D31.4	D16.7	D24.0
SOAF	K28.5	D24.0	D30.0	D01.4

7.2.4 Primitive sequences

7.2.5 Primitives not specific to type of connections

7.2.5.1 AIP (Arbitration in progress)

7.2.5.2 ALIGN

7.2.5.3 BREAK

BREAK is used to abort a connection request or break a connection.

See 7.12.5 and 7.12.7 for details on breaking connections.

7.2.5.XX BREAK_REPLY

BREAK_REPLY is used to confirm receipt of a BREAK primitive sequence.

See 7.12.5 and 7.12.7 for details on breaking connections.

7.2.5.11 OPEN_REJECT

...

All of the OPEN_REJECT versions defined in table 84 shall result in the originating port retrying the connection request.

Table 84 — OPEN_REJECT retry primitives

Primitive	Originator	Description
OPEN_REJECT (NO DESTINATION) [a]	Expander phy	Either: a) No such destination phy; b) the expander device determines the connection request would have to be routed to the same expander port as the expander port through which the connection request arrived (e.g., the destination SAS address equals the source SAS address) and the expander device has not chosen to return OPEN_REJECT (BAD DESTINATION) (see 7.12.4.3 and 7.12.4.4); or c) the SAS address is valid for an STP target port in an STP/SATA bridge, but the initial Register - Device to Host FIS has not been successfully received (see 10.4.3.7).
OPEN_REJECT (PATHWAY BLOCKED) [b]	Expander phy	An expander device determined the pathway was blocked by higher priority connection requests.
OPEN_REJECT (RESERVED CONTINUE 0) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED CONTINUE 1) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED INITIALIZE 0) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED INITIALIZE 1) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED STOP 0) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED).
OPEN_REJECT (RESERVED STOP 1) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED).
OPEN_REJECT (RETRY) [c]	Destination phy	Phy with destination SAS address exists but is not able to accept connections.

[a] If the I_T Nexus Loss timer is already running, it continues running; if it is not already running, it is initialized and started. Stop retrying the connection request if the I_T Nexus Loss timer expires.
[b] If the I_T Nexus Loss timer is already running, it continues running. Stop retrying the connection request if the I_T Nexus Loss timer expires.
[c] If the I_T Nexus Loss timer (see 8.2.2) is already running, it is stopped.

NOTE 21 - Some SAS phys also transmit OPEN_REJECT (RETRY) if they receive an OPEN address frame while their SL_CC state machines are in the SL_CC5:BreakWait state (see 7.14.4.7).

Editor's Note 1: WG discuss - should NOTE 21 be removed or altered?

7.8 Address frames

7.8.1 Address frames overview

7.8.2 IDENTIFY address frame

Table 93 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is sent after the phy reset sequence completes if the physical link is a SAS physical link.

Table 93 — IDENTIFY address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	Restricted (for OPEN address frame)	DEVICE TYPE			ADDRESS FRAME TYPE (0h)			
1	Restricted (for OPEN address frame)							
2	Reserved				SSP INITIATOR PORT	STP INITIATOR PORT	SMP INITIATOR PORT	Restricted (for OPEN address frame)
3	Reserved				SSP TARGET PORT	STP TARGET PORT	SMP TARGET PORT	Restricted (for OPEN address frame)
4	DEVICE NAME							
11								
12	SAS ADDRESS							
19								
20	PHY IDENTIFIER							
21	Reserved							PHY BREAK_REPLY CAPABLE
22	Reserved							
27								
28	(MSB) CRC							
31								(LSB)

...

The PHY IDENTIFIER field specifies the phy identifier of the phy transmitting the IDENTIFY address frame.

The PHY BREAK_REPLY CAPABLE bit specifies that the phy is capable of responding to received BREAK primitive sequences with a BREAK_REPLY primitive sequence.

SAS phys shall enable the BREAK_REPLY method of responding to received BREAK primitive sequences when:

1) the PHY BREAK_RESPONSE CAPABLE bit transmitted by the phy in the outgoing IDENTIFY address frame is set to one; and
2) the PHY BREAK_RESPONSE CAPABLE bit received by the phy in the incoming IDENTIFY address frame is set to one.

If either the PHY BREAK_REPLY CAPABLE bit transmitted or received in the IDENTIFY address frame are set to zero the SAS phy shall respond to received BREAK primitive sequences with a BREAK primitive sequence.

SAS-2 phys shall set the PHY BREAK_RESPONSE CAPABLE bit to one in their outgoing IDENTIFY address frame.

...

7.12 Connections

7.12.1 Connections overview

7.12.2 Opening a connection

7.12.3 Arbitration fairness

7.12.4 Arbitration and resource management in an expander device

7.12.5 Aborting a connection request

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

After transmitting BREAK, the source phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

After a phy transmits a BREAK to abort a connection request, it shall expect one of the results listed in table 105.

Table 105 — Results of aborting a connection request

Result	Description
Receive BREAK or BREAK_REPLY	This confirms that the connection request has been aborted.
Break Timeout timer expires	The originating phy shall assume the connection request has been aborted.

When a phy sourcing a BREAK is attached to an expander device, the BREAK or BREAK_REPLY (see 7.8.2) response to the source phy is generated by the expander phy to which the source phy is attached, not the other SAS phy 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. After transmitting BREAK or BREAK_REPLY (see 7.8.2) back to the originating phy, the expander device shall ensure that an open response does not occur (i.e., the expander device shall not forward dwords from the destination any more). Figure 141 shows an example of BREAK usage.

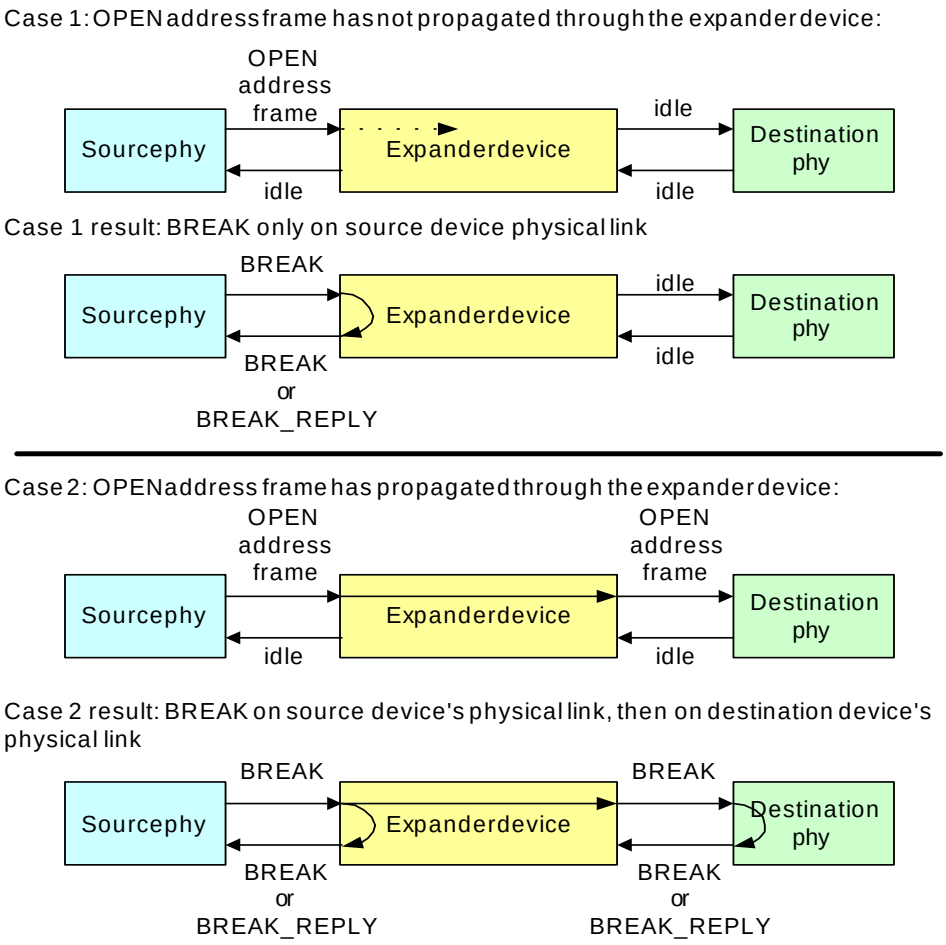

Figure 141 — Aborting a connection request with BREAK

Figure 142 shows the sequence for a connection request where the Open Timeout timer expires.

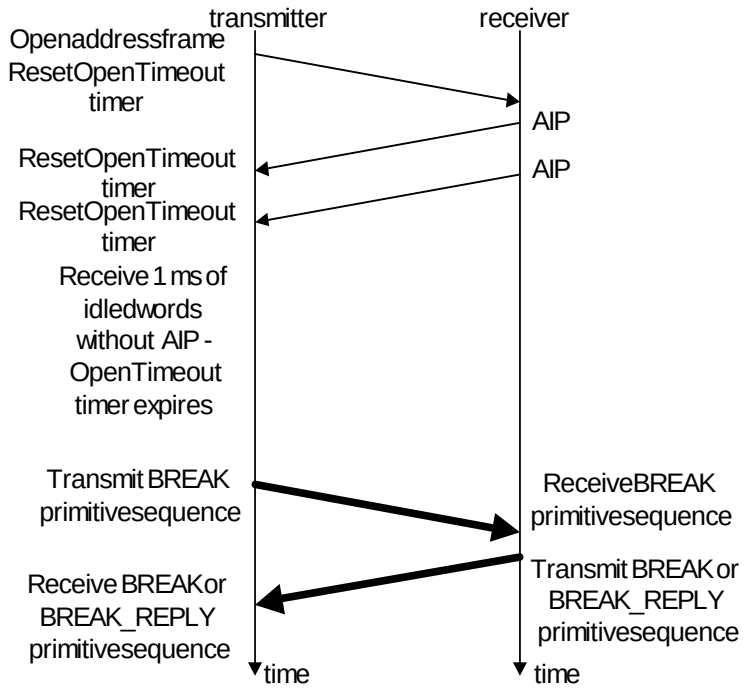

Figure 142 — Connection request timeout example

7.12.6 Closing a connection

7.12.7 Breaking a connection

In addition to aborting a connection request, BREAK may also be used to break a connection, in cases where CLOSE is not available. After transmitting BREAK, the originating phy shall ignore all incoming dwords except for BREAKs and BREAK_REPLYs.

After transmitting BREAK, the source phy shall initialize a Break Timeout timer to 1 ms and start the Break Timeout timer.

After a phy transmits a BREAK to break a connection, it shall expect one of the results listed in table 107.

Table 107 — Results of breaking a connection

Result	Description
Receive BREAK or BREAK_REPLY	This confirms that the connection has been broken.
Break Timeout timer expires	The originating phy shall assume the connection has been broken. The originating phy may perform a link reset sequence.

In addition to a BREAK, a connection is considered broken if a link reset sequence starts (i.e., the SP state machine transitions from SP15:SAS_PHY_Ready or SP22:SATA_PHY_Ready to SP0:OOB_COMINIT (see 6.8)).

See 7.16.6 for additional rules on breaking an SSP connection.

7.13 Rate matching

...

A phy shall stop inserting ALIGNs and/or NOTIFYs for rate matching after:

a) transmitting the first dword in a CLOSE;
b) transmitting the first dword in a BREAK;
c) transmitting the first dword in a BREAK_REPLY;
d) receiving an OPEN_REJECT for a connection request; or
e) losing arbitration to a received OPEN address frame.

7.14 SL (link layer for SAS phys) state machines

7.14.1 SL state machines overview

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

a) SL_RA (receive OPEN address frame) state machine (see 7.14.3); and
b) SL_CC (connection control) state machine (see 7.14.4).

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

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

Figure 146 shows the messages sent to the SL transmitter and received from the SL receiver.

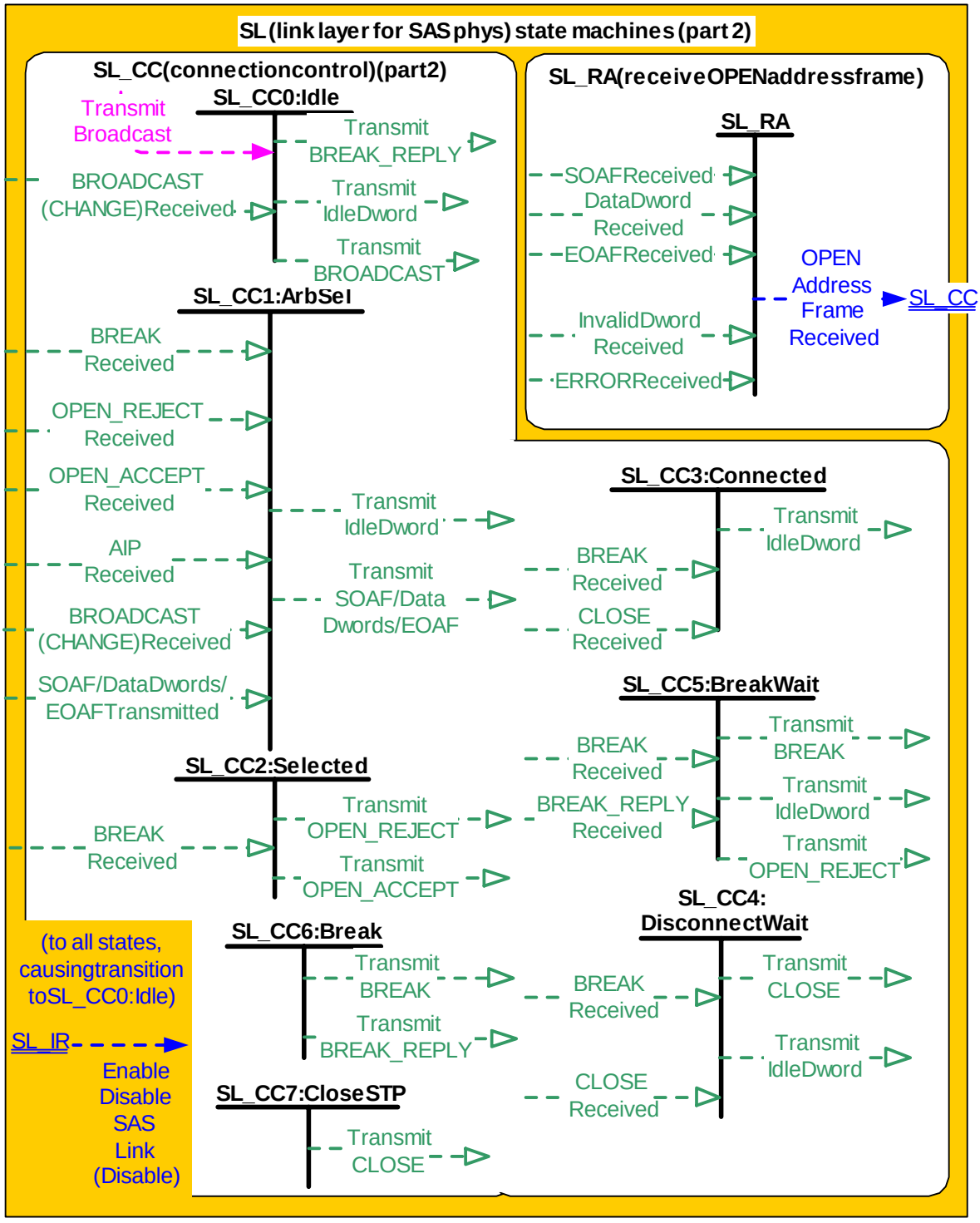

Figure 146 — SL (link layer for SAS phys) state machines (part 2)

7.14.2 SL transmitter and receiver

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

a) Transmit Idle Dword;
b) Transmit SOAF/Data Dwords/EOAF;

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

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

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

Editor's Note 2: WG discuss priority of BREAK_REPLY

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

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

a) SOAF/Data Dwords/EOAF Transmitted.

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

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

The SL receiver shall ignore all other dwords.

7.14.3 SL_RA (receive OPEN address frame) state machine

7.14.4 SL_CC (connection control) state machine

7.14.4.1 SL_CC state machine overview

7.14.4.2 SL_CC0:Idle state

7.14.4.2.1 State description

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

Upon entry into this state, this state shall send:

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

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

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

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

If a BREAK received message is received, this state shall send a Transmit BREAK_REPLY message to the SL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.8.2).

7.14.4.3 SL_CC1:ArbSel state

7.14.4.4 SL_CC2:Selected state

7.14.4.4.1 State description

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

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

1) If the OPEN address frame DESTINATION SAS ADDRESS field does not match the SAS address of this port, this state shall send a Transmit OPEN_REJECT (Wrong Destination) message to the SL transmitter (see 7.14.4.4.2);
2) If the OPEN address frame INITIATOR PORT bit, PROTOCOL field, FEATURES field, and/or INITIATOR CONNECTION TAG field are set to values that are not supported (e.g., a connection request from an SMP target port), this state shall send a Transmit OPEN_REJECT (Protocol Not Supported) message to the SL transmitter (see 7.14.4.4.2);
3) If the OPEN address frame CONNECTION RATE field is set to a connection rate that is not supported, this state shall send a Transmit OPEN_REJECT (Connection Rate Not Supported) message to the SL transmitter (see 7.14.4.4.2);
4) If the OPEN address frame PROTOCOL field is set to STP, the source SAS address is not that of the STP initiator port with an affiliation established or the source SAS address is not that of an STP initiator port with task file register set resources (see 7.17.5), this state shall send a Transmit OPEN_REJECT (STP Resources Busy) message to the SL transmitter (see 7.14.4.4.2);
5) If an Accept_Reject Opens (Reject SSP) request, Accept_Reject Opens (Reject SMP) request, or Accept_Reject Opens (Reject STP) request is received and the requested protocol is the corresponding protocol, this state shall send a Transmit OPEN_REJECT (Retry) message to the SL transmitter (see 7.14.4.4.2);

6) If the requested protocol is SSP and this state has not received an Accept_Reject Opens (Reject SSP) request then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SSP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3);
7) If the requested protocol is SMP and this state has not received an Accept_Reject Opens (Reject SMP) request then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (SMP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3); or
8) If the requested protocol is STP and this state has not received an Accept_Reject Opens (Reject STP) request then this state shall send a Transmit OPEN_ACCEPT message to the SL transmitter and send a Connection Opened (STP, Destination Opened) confirmation to the port layer (see 7.14.4.4.3).

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

NOTE 36 - Possible livelock scenarios can occur when a SAS phy transmits BREAK to abort a connection request (e.g., if its Open Timeout timer expires). SAS phys should respond to OPEN Address frames faster than 1 ms to reduce susceptibility to this problem.

Editor's Note 3: WG discuss - should NOTE 36 be removed or altered?

7.14.4.5 SL_CC3:Connected state

7.14.4.6 SL_CC4:DisconnectWait state

7.14.4.6.1 State description

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

Upon entry into this state, this state shall:

1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter (see 7.17.7); and
2) initialize and start the Close Timeout timer.

A CLOSE Received message may be received at any time while in this state. If a CLOSE Received (Clear Affiliation) is received during an STP connection, this state shall clear any affiliation (see 7.17.5).

NOTE 37 - Possible livelock scenarios can occur when a SAS phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). SAS phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

Editor's Note 4: WG discuss - should NOTE 37 be removed or altered?

7.14.4.7 SL_CC5:BreakWait state

7.14.4.7.1 State description

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

Upon entry into this state, this state shall:

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

NOTE 38 - Some SAS phys send a Transmit OPEN_REJECT (Retry) message to the SL transmitter in response to each OPEN Address Frame Received message received while in this state.

Editor's Note 5: WG discuss - should NOTE 38 be removed or altered?

7.14.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle

This transition shall occur after:

a) receiving a BREAK Received message;
b) receiving a BREAK_REPLY Received message; or
c) the Break Timeout timer expires.

7.14.4.8 SL_CC6:Break state

7.14.4.8.1 State description

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

Upon entry into this state, this state shall:

a) send a Transmit BREAK message to the SL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is not enabled (see 7.8.2); or
b) send a Transmit BREAK_REPLY message to the SL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.8.2).

7.14.4.8.2 Transition SL_CC6:Break to SL_CC0:Idle

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

7.14.4.9 SL_CC7:CloseSTP state

7.14.4.9.1 State description

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

Upon entry into this state, this state shall:

1) send a Transmit CLOSE (Normal) message or Transmit CLOSE (Clear Affiliation) message to the SL transmitter (see 7.17.7); and
2) send a Connection Closed (Normal) confirmation to the port layer (see 7.14.4.9.2).

NOTE 39 - Possible livelock scenarios can occur when a SAS phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). SAS phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

Editor's Note 6: WG discuss - should NOTE 39 be removed or altered?

7.15 XL (link layer for expander phys) state machine

7.15.1 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 (e.g., the ECM and ECR).

This state machine consists of the following states:

a) XL0:Idle (see 7.15.3)(initial state);

b) XL1:Request_Path (see 7.15.4);
c) XL2:Request_Open (see 7.15.5);
d) XL3:Open_Confirm_Wait (see 7.15.6);
e) XL4:Open_Reject (see 7.15.7);
f) XL5:Forward_Open (see 7.15.8);
g) XL6:Open_Response_Wait (see 7.15.9);
h) XL7:Connected (see 7.15.10);
i) XL8:Close_Wait (see 7.15.11);
j) XL9:Break (see 7.15.12); and
k) XL10:Break_Wait (see 7.15.13).

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 7.9.5).

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).

Any message received by a state that is not referred to in the description of that state shall be ignored.

The XL state machine shall maintain the timers listed in table 108.

Table 108 — XL timers

Timer	Initial value
Partial Pathway Timeout timer	Partial pathway timeout value (see 7.12.4.5)
Break Timeout timer	1 ms

Figure 147 shows several states in the XL state machine.

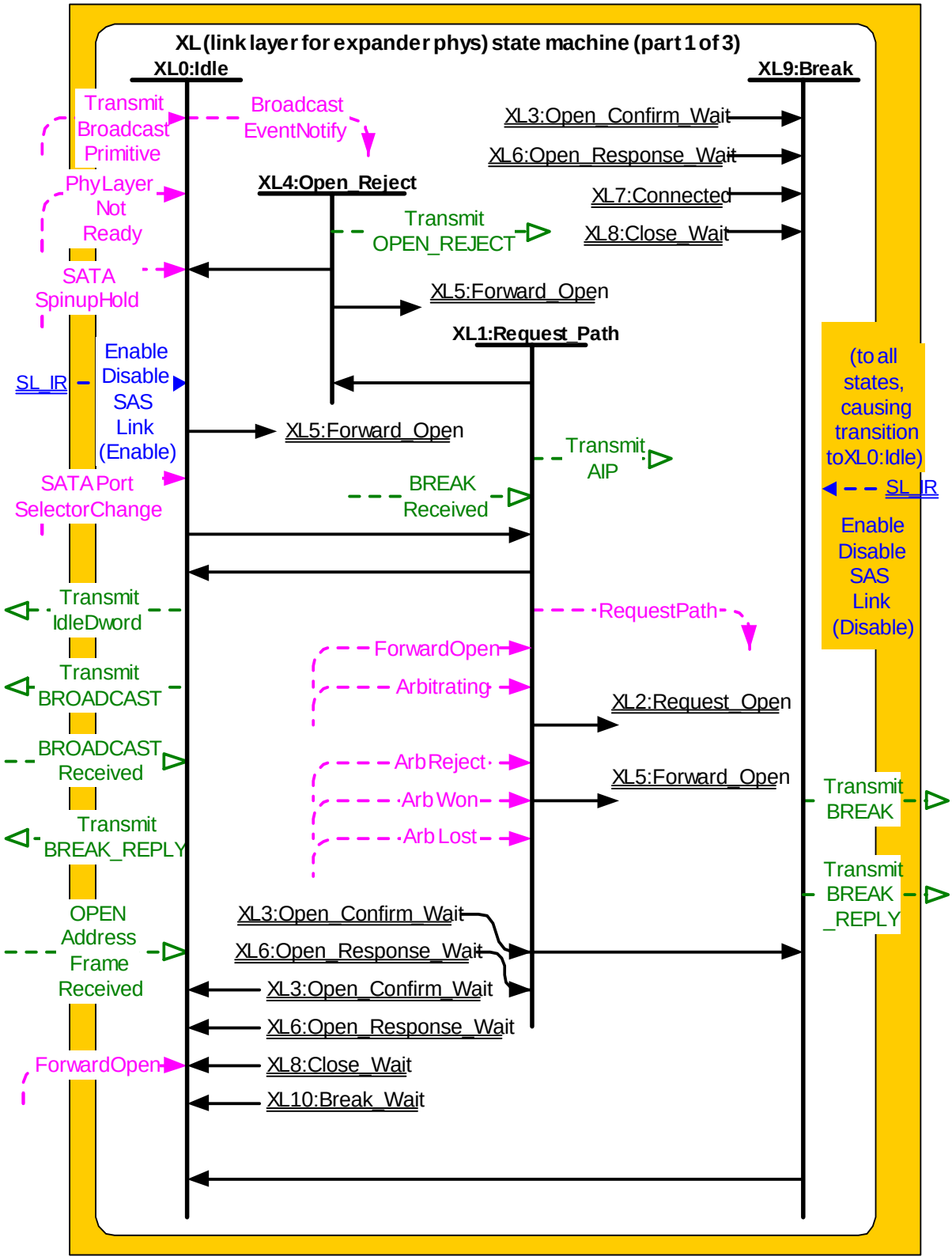

Figure 147 — XL (link layer for expander phys) state machine (part 1)

Figure 148 shows additional states in the XL state machine.

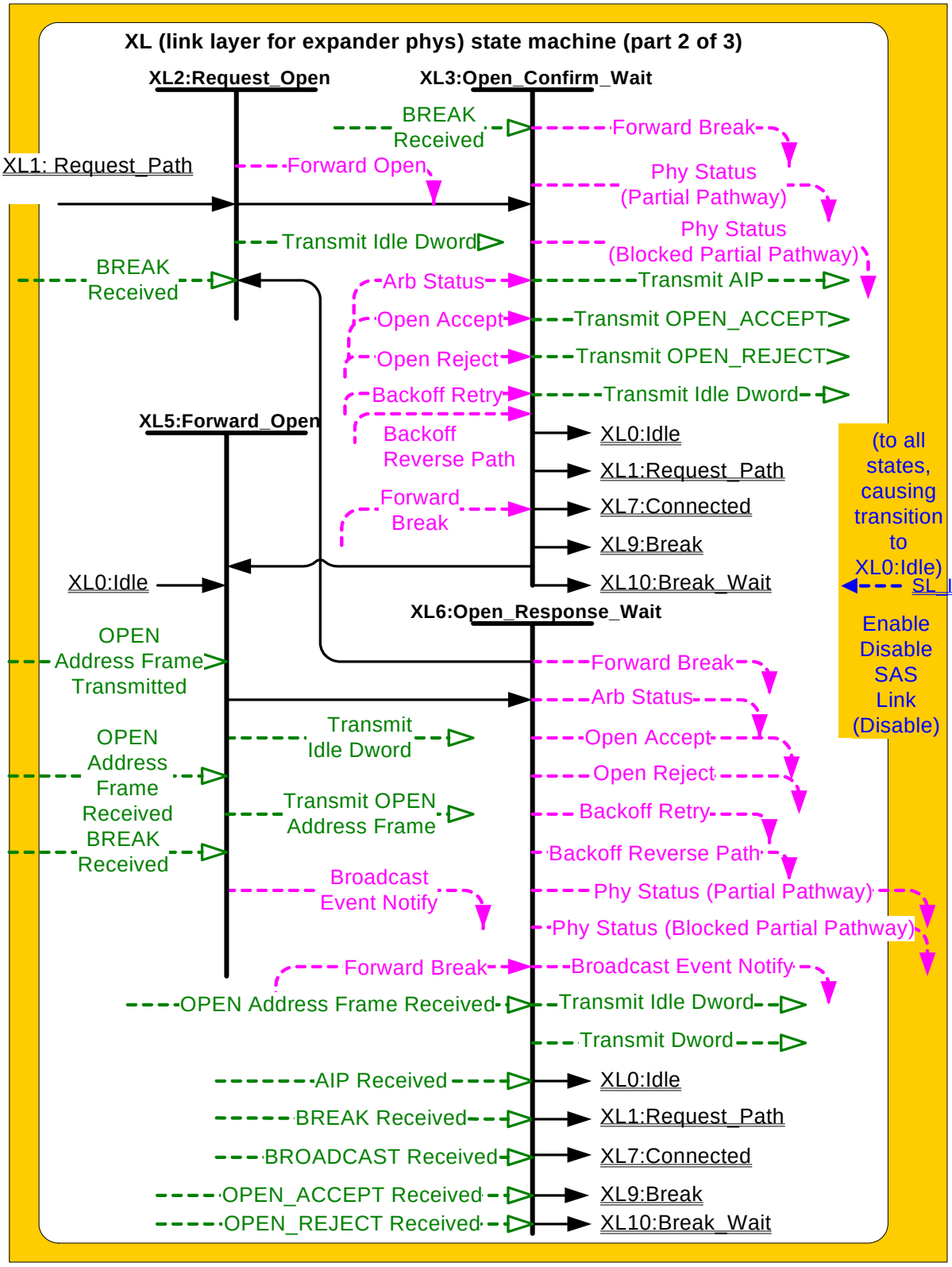

Figure 148 — XL (link layer for expander phys) state machine (part 2)

Figure 149 shows additional states in the XL state machine.

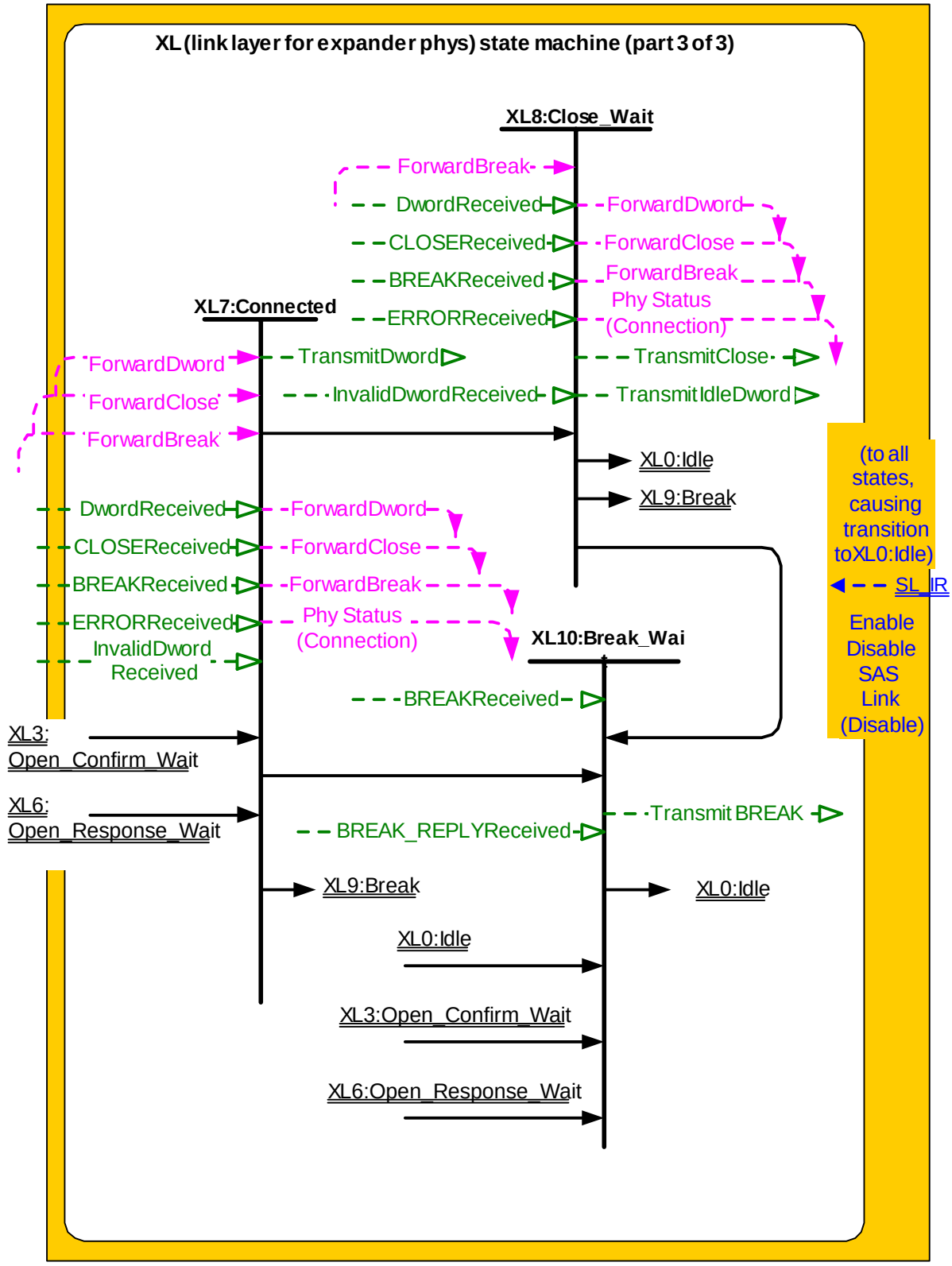

Figure 149 — XL (link layer for expander phys) state machine (part 3)

7.15.2 XL transmitter and receiver

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

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

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

a) OPEN Address Frame Transmitted.

The XL transmitter shall ensure clock skew management requirements are met (see 7.8) while originating dwords.

The XL transmitter shall ensure clock skew management requirements are met (see 7.8) during and after switching from forwarding dwords to originating dwords, including, for example:

a) when transmitting BREAK;
b) when transmitting BREAK_REPLY;
c) when transmitting CLOSE;
d) when transmitting an idle dword after closing a connection (i.e., after receiving BREAK, BREAK_REPLY, or CLOSE);
e) while transmitting a SATA frame to a SAS physical link, when transmitting the first SATA_HOLDA in response to detection of SATA_HOLD; and
f) while receiving dwords of a SATA frame from a SAS physical link, when transmitting SATA_HOLD.

NOTE 40 - The XL transmitter may always insert an ALIGN or NOTIFY before transmitting a BREAK, BREAK_REPLY, CLOSE, or SATA_HOLDA to meet clock skew management requirements.

The XL transmitter shall insert an ALIGN or NOTIFY before switching from originating dwords to forwarding dwords, including, for example:

a) when transmitting OPEN_ACCEPT;
b) when transmitting the last idle dword before a connection is established (i.e., after receiving OPEN_ACCEPT);
c) while transmitting a SATA frame to a SAS physical link, when transmitting the last dword from the SATA flow control buffer in response to release of SATA_HOLD;
d) while transmitting a SATA frame to a SAS physical link, when transmitting the last SATA_HOLDA in response to release of SATA_HOLD (e.g., if the SATA flow control buffer is empty); and
e) while receiving dwords of a SATA frame from a SAS physical link, when transmitting the last SATA_HOLD.

NOTE 41 - This ensures that clock skew management requirements are met, even if the forwarded dword stream does not include an ALIGN or NOTIFY until the last possible dword.

The XL transmitter shall ensure rate matching requirements are met during a connection (see 7.13).

The XL transmitter shall ensure STP initiator phy throttling requirements are met (see 7.17.2) when:

a) transmitting dwords in the direction of an STP target port while originating dwords (e.g., while transmitting SATA_HOLD, SATA_HOLDA, or unloading the SATA flow control buffer);
b) switching from forwarding dwords to originating dwords; and
c) switching from originating dwords to forwarding dwords.

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

The XL receiver sends the following messages to the XL state machine indicating primitive sequences, frames, and dwords received from the SP_DWS receiver (see 6.9.2):

a) AIP Received with an argument indicating the specific type (e.g., AIP Received (Normal));
b) BREAK Received;
c) BREAK_REPLY Received;
d) BROADCAST Received;
e) CLOSE Received;
f) OPEN_ACCEPT Received;
g) OPEN_REJECT Received;
h) OPEN Address Frame Received;
i) Dword Received with an argument indicating the data dword or primitive received; and
j) Invalid Dword Received.

The XL receiver shall ignore all other dwords.

While receiving an address frame, if the XL receiver receives an invalid dword or ERROR, then the XL receiver shall:

a) ignore the invalid dword or ERROR; or
b) discard the address frame.

7.15.3 XL0:Idle state

7.15.3.1 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 Layer 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 an Enable Disable SAS Link (Enable) message is received, this state shall send a Broadcast Event Notify (Identification Sequence Complete) request to the BPP.

If a SATA Port Selector Change confirmation is received, this state shall send a Broadcast Event Notify (SATA Port Selector Change) 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.

If a BREAK received message is received, this state shall send a Transmit BREAK_REPLY message to the XL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.8.2).

7.15.4 XL1:Request_Path state

7.15.5 XL2:Request_Open state

7.15.6 XL3:Open_Confirm_Wait state

7.15.7 XL4:Open_Reject state

7.15.8 XL5:Forward_Open state

7.15.9 XL6:Open_Response_Wait state

7.15.10 XL7:Connected state

7.15.11 XL8:Close_Wait state

7.15.11.1 State description

This state closes a connection and releases path resources.

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

NOTE 42 - Possible livelock scenarios can occur when a SAS phy transmits BREAK to break a connection (e.g., if its Close Timeout timer expires). SAS phys should respond to CLOSE faster than 1 ms to reduce susceptibility to this problem.

Editor's Note 7: WG discuss - should NOTE 42 be removed or altered?

...

7.15.12 XL9:Break state

7.15.12.1 State description

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

This state shall:

a) send a Transmit BREAK message to the XL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is not enabled (see 7.8.2); or
b) send a Transmit BREAK_REPLY message to the XL transmitter if the BREAK_REPLY method of responding to received BREAK primitive sequences is enabled (see 7.8.2).

7.15.12.2 Transition XL9:Break to XL0:Idle

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

7.15.13 XL10:Break_Wait state

7.15.13.1 State description

This state closes the connection if there is one and releases path resources associated with the connection.

Upon entry into this state, this state shall:

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

7.15.13.2 Transition XL10:Break_Wait to XL0:Idle

This transition shall occur after:

a) a BREAK Received message is received;
b) a BREAK_REPLY Received message is received; or
c) the Break Timeout timer expires.

7.16 SSP link layer

7.16.8 SSP (link layer for SSP phys) state machines

7.16.8.6.5 SSP_TF4:Transmit_DONE 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 SSP_TF1: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.

NOTE 46 - Possible livelock scenarios can occur when a SAS phy transmits BREAK to break a connection (e.g., if its Done Timeout timer expires). SAS phys should respond to DONE faster than 1 ms to reduce susceptibility to this problem.

Editor's Note 8: WG discuss - should NOTE 46 be removed or altered?

7.17 STP link layer

7.17.1 STP frame transmission and reception

7.17.2 STP initiator phy throttling

...

ALIGNs and NOTIFYs inserted for STP initiator phy throttling are in addition to ALIGNs and NOTIFYs inserted for clock skew management (see 7.8) and rate matching (see 7.13). See Annex H for a summary of their combined requirements.

A phy shall start inserting ALIGNs and NOTIFYs for STP initiator phy throttling after:

a) transmitting an OPEN_ACCEPT; or
b) sending the first SATA primitive after receiving an OPEN_ACCEPT.

A phy shall stop inserting ALIGNs and NOTIFYs for STP initiator phy throttling after:

a) transmitting the first dword in a CLOSE;
b) transmitting the first dword in a BREAK; or
c) transmitting the first dword in a BREAK_REPLY.

8 Application layer

8.4 Management application layer

8.4.3.5 DISCOVER function

Table 198 defines the response format.

Table 198 — DISCOVER response (part 1 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
0	SMP FRAME TYPE (41h)							
1	FUNCTION (10h)							
2	FUNCTION RESULT							
3	RESPONSE LENGTH (0Eh)							
4	Reserved							
8								
9	PHY IDENTIFIER							
10	Reserved							
11								
12	Reserved	ATTACHED DEVICE TYPE			Reserved			
13	Reserved				NEGOTIATED PHYSICAL LINK RATE			
14	Reserved				ATTACHED SSP INITIATOR	ATTACHED STP INITIATOR	ATTACHED SMP INITIATOR	ATTACHED SATA HOST
15	ATTACHED SATA PORT SELECTOR	Reserved			ATTACHED SSP TARGET	ATTACHED STP TARGET	ATTACHED SMP TARGET	ATTACHED SATA DEVICE
16	SAS ADDRESS							
23								
24	ATTACHED SAS ADDRESS							
31								
32	ATTACHED PHY IDENTIFIER							
33	Reserved							ATTACHED PHY BREAK_REPLY CAPABLE
34	Reserved							
39								
40	PROGRAMMED MINIMUM PHYSICAL LINK RATE				HARDWARE MINIMUM PHYSICAL LINK RATE			
41	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				HARDWARE MAXIMUM PHYSICAL LINK RATE			
42	PHY CHANGE COUNT							

Table 198 — DISCOVER response (part 2 of 2)

Byte\Bit	7	6	5	4	3	2	1	0
43	VIRTUAL PHY	PHY BREAK_REPLY CAPABLE	Reserved		PARTIAL PATHWAY TIMEOUT VALUE			
44	Reserved				ROUTING ATTRIBUTE			
45	Reserved	CONNECTOR TYPE						
46	CONNECTOR ELEMENT INDEX							
47	CONNECTOR PHYSICAL LINK							
48	Reserved							
49								
50	Vendor specific							
51								
52	ATTACHED DEVICE NAME							
59								
60	(MSB) CRC							
63								(LSB)

...

The ATTACHED PHY BREAK_REPLY CAPABLE bit indicates the value of the PHY BREAK_REPLY CAPABLE bit received in the IDENTIFY address frame (see 7.8.2) during the identification sequence. If a SAS phy reset sequence occurs (see 6.7.4) then the ATTACHED PHY BREAK_REPLY CAPABLE bit shall be set to zero.

...

A VIRTUAL PHY bit set to one indicates the phy is part of an internal port and the attached device is contained within the expander device. A VIRTUAL PHY bit set to zero indicates the phy is a physical phy and the attached device is not contained within the expander device.

A PHY BREAK_REPLY CAPABLE bit set to one indicates the phy is capable of responding to received BREAK primitive sequences with a BREAK_REPLY primitive sequence (see 7.8.2). A PHY BREAK_REPLY CAPABLE bit set to zero indicates the phy is not capable of responding to received BREAK primitive sequences with a BREAK_REPLY primitive sequence (see 7.8.2).

The PARTIAL PATHWAY TIMEOUT VALUE field indicates the partial pathway timeout value in microseconds (see 7.12.4.5) set by the PHY CONTROL function (see 10.4.3.12).

NOTE 47 - The recommended default value for PARTIAL PATHWAY TIMEOUT VALUE is 7 µs.

I.10 BREAK handling during path arbitration when BREAK_REPLY is not enabled

Figure I.10 shows an expander device responding to the reception of a BREAK during path arbitration when BREAK_REPLY is not enabled.

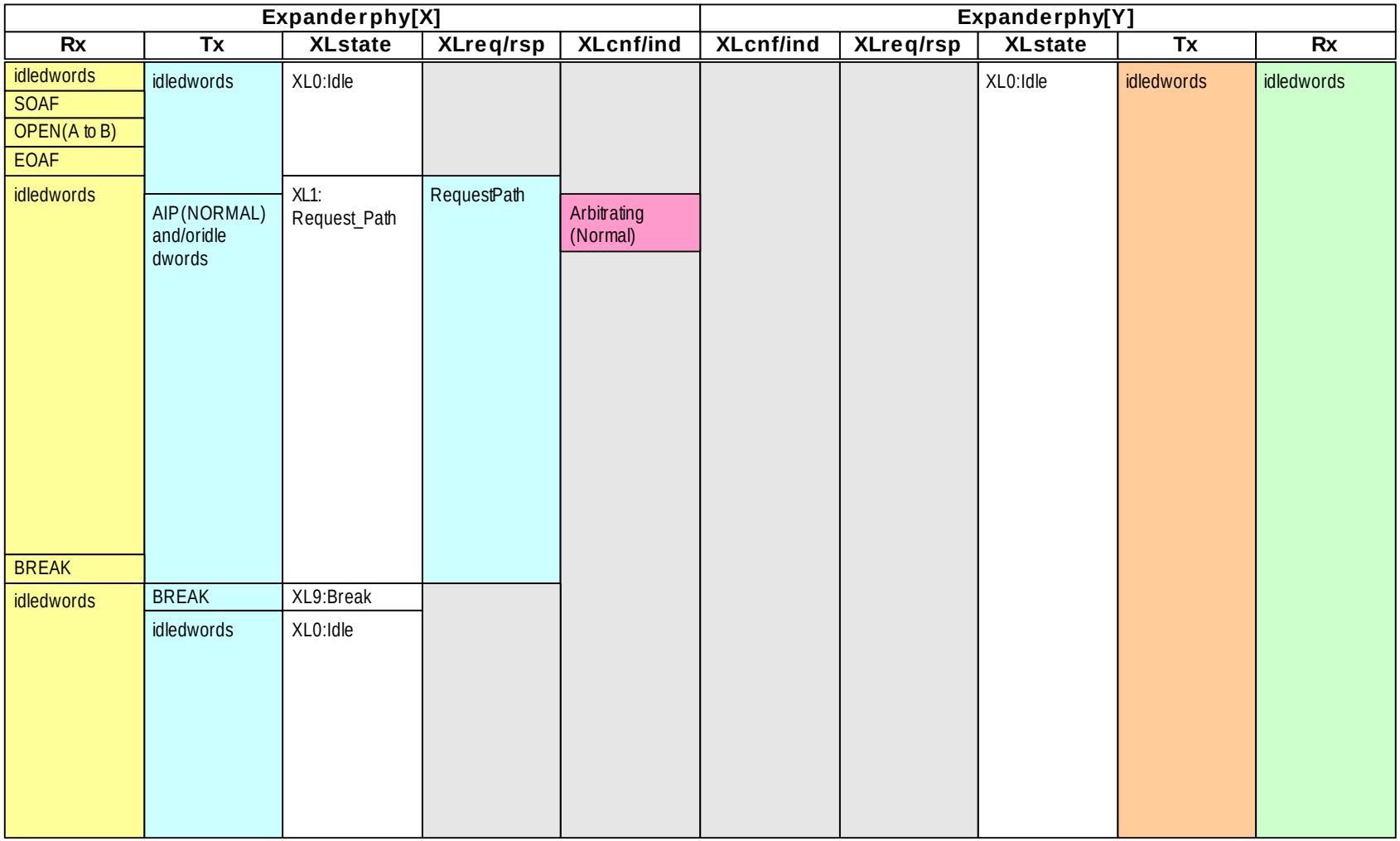

Figure I.10 — BREAK handling during path arbitration (BREAK_REPLY not enabled)

I.11 BREAK handling during connection when BREAK_REPLY is not enabled

Figure I.11 shows an expander device responding to the reception of a BREAK during a connection when BREAK_REPLY is not enabled.

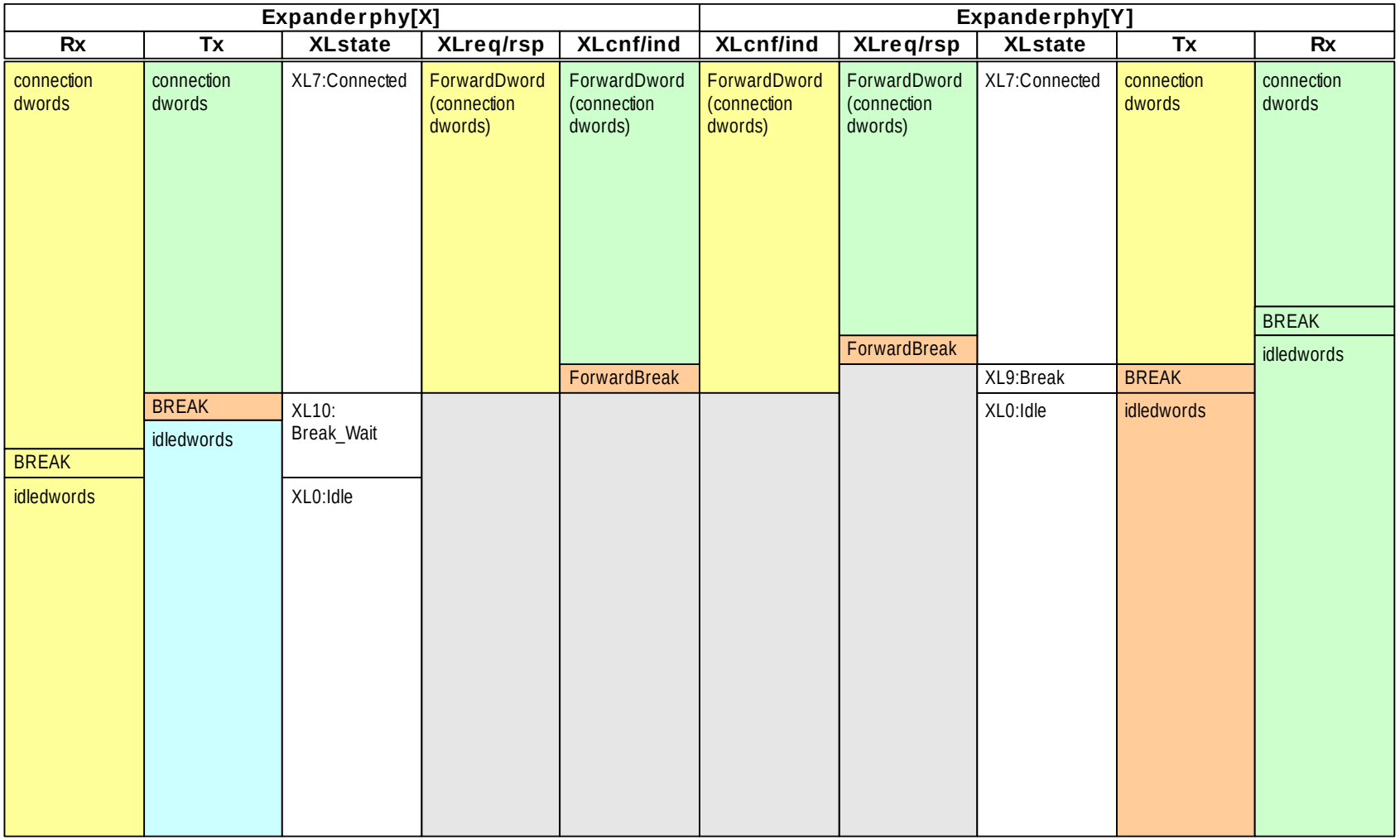

Figure I.11 — BREAK handling during a connection (BREAK_REPLY not enabled)

I.<X> BREAK handling during path arbitration when BREAK_REPLY is enabled

Figure I.<X> shows an expander device responding to the reception of a BREAK during path arbitration when BREAK_REPLY is enabled.

Expanderphy[X]					Expanderphy[Y]				
Rx	**Tx**	**XLstate**	**XLreq/rsp**	**XLcnf/ind**	**XLcnf/ind**	**XLreq/rsp**	**XLstate**	**Tx**	**Rx**
idledwords	idledwords	XL0:Idle					XL0:Idle	idledwords	idledwords
SOAF									
OPEN(A to B)									
EOAF									
idledwords		XL1: Request_Path	RequestPath						
	AIP(NORMAL) and/oridle dwords			Arbitrating (Normal)					
BREAK									
idledwords	BREAK_REPLY	XL9:Break							
	idledwords	XL0:Idle							

Figure I.<X> — BREAK handling during path arbitration (BREAK_REPLY enabled)

I.<Y> BREAK handling during connection when BREAK_REPLY is enabled

Figure I.<Y> shows an expander device responding to the reception of a BREAK during a connection when BREAK_REPLY is enabled.

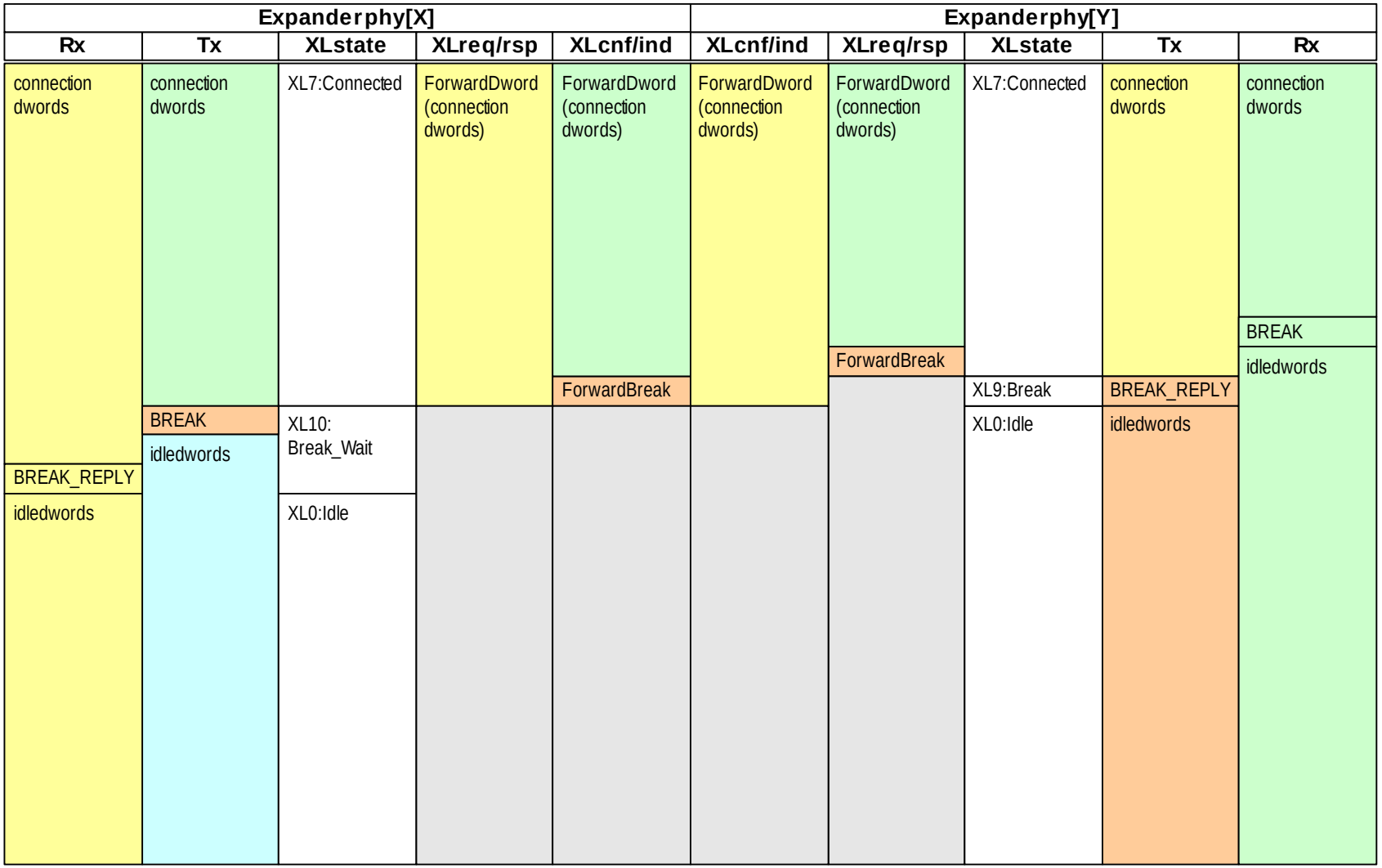

Figure I.<Y> — BREAK handling during a connection (BREAK_REPLY enabled)

Annex J
(informative)

Primitive encoding

This annex describes a set of the K28.5-based primitive encodings whose 40-bit values (after 8b10b encoding with either starting running disparity) have a Hamming distance (i.e., the number of bits different in two patterns) of at least 8. All the primitive encodings in 7.2 were selected from this list. Unassigned encodings may be used by future versions of this standard.

Table J.1 — Primitives with Hamming distance of 8 (part 1 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D01.3	D01.3	D01.3	ALIGN (2)
K28.5	D01.4	D01.4	D01.4	ACK
K28.5	D01.4	D02.0	D31.4	RRDY (RESERVED 0)
K28.5	D01.4	D04.7	D24.0	NAK (RESERVED 1)
K28.5	D01.4	D07.3	D30.0	CREDIT_BLOCKED
K28.5	D01.4	D16.7	D07.3	NAK (RESERVED 2)
K28.5	D01.4	D24.0	D16.7	RRDY (NORMAL)
K28.5	D01.4	D27.4	D04.7	NAK (CRC ERROR)
K28.5	D01.4	D30.0	D02.0	RRDY (RESERVED 1)
K28.5	D01.4	D31.4	D29.7	NAK (RESERVED 0)
K28.5	D02.0	D01.4	D29.7	ERROR
K28.5	D02.0	D02.0	D02.0	HARD_RESET
K28.5	D02.0	D04.7	D01.4	CLOSE (RESERVED 1)
K28.5	D02.0	D07.3	D04.7	CLOSE (CLEAR AFFILIATION)
K28.5	D02.0	D16.7	D31.4	
K28.5	D02.0	D24.0	D07.3	BREAK
K28.5	D02.0	D29.7	D16.7	BREAK_REPLY
K28.5	D02.0	D30.0	D27.4	CLOSE (NORMAL)
K28.5	D02.0	D31.4	D30.0	CLOSE (RESERVED 0)
K28.5	D04.7	D01.4	D24.0	BROADCAST (RESERVED 1)
K28.5	D04.7	D02.0	D01.4	BROADCAST (CHANGE)
K28.5	D04.7	D04.7	D04.7	BROADCAST (RESERVED 2)
K28.5	D04.7	D07.3	D29.7	BROADCAST (SES)
K28.5	D04.7	D16.7	D02.0	BROADCAST (RESERVED 3)
K28.5	D04.7	D24.0	D31.4	BROADCAST (RESERVED CHANGE 0)
K28.5	D04.7	D27.4	D07.3	BROADCAST (RESERVED CHANGE 1)
K28.5	D04.7	D29.7	D30.0	BROADCAST (RESERVED 4)
K28.5	D04.7	D31.4	D27.4	
K28.5	D07.0	D07.0	D07.0	ALIGN (1)
K28.5	D07.3	D01.4	D31.4	

Table J.1 — Primitives with Hamming distance of 8 (part 2 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D07.3	D02.0	D04.7	
K28.5	D07.3	D04.7	D30.0	
K28.5	D07.3	D07.3	D07.3	
K28.5	D07.3	D24.0	D29.7	
K28.5	D07.3	D27.4	D16.7	
K28.5	D07.3	D29.7	D27.4	
K28.5	D07.3	D30.0	D24.0	
K28.5	D07.3	D31.4	D02.0	
K28.5	D10.2	D10.2	D27.3	ALIGN (0)
K28.5	D16.7	D01.4	D02.0	
K28.5	D16.7	D02.0	D07.3	
K28.5	D16.7	D04.7	D31.4	
K28.5	D16.7	D16.7	D16.7	OPEN_ACCEPT
K28.5	D16.7	D24.0	D27.4	
K28.5	D16.7	D27.4	D30.0	
K28.5	D16.7	D29.7	D24.0	
K28.5	D16.7	D30.0	D04.7	
K28.5	D16.7	D31.4	D01.4	
K28.5	D24.0	D01.4	D16.7	
K28.5	D24.0	D02.0	D29.7	
K28.5	D24.0	D04.7	D07.3	SOF
K28.5	D24.0	D07.3	D31.4	EOAF
K28.5	D24.0	D16.7	D27.4	EOF
K28.5	D24.0	D24.0	D24.0	
K28.5	D24.0	D27.4	D02.0	
K28.5	D24.0	D29.7	D04.7	
K28.5	D24.0	D30.0	D01.4	SOAF
K28.5	D27.3	D27.3	D27.3	ALIGN (3)
K28.5	D27.4	D01.4	D07.3	AIP (RESERVED WAITING ON PARTIAL)
K28.5	D27.4	D04.7	D02.0	
K28.5	D27.4	D07.3	D24.0	AIP (WAITING ON CONNECTION)
K28.5	D27.4	D16.7	D30.0	AIP (RESERVED 1)
K28.5	D27.4	D24.0	D04.7	AIP (WAITING ON PARTIAL)
K28.5	D27.4	D27.4	D27.4	AIP (NORMAL)
K28.5	D27.4	D29.7	D01.4	AIP (RESERVED 2)
K28.5	D27.4	D30.0	D29.7	AIP (WAITING ON DEVICE)
K28.5	D27.4	D31.4	D16.7	AIP (RESERVED 0)
K28.5	D29.7	D02.0	D30.0	OPEN_REJECT (RESERVED CONTINUE 0)

Table J.1 — Primitives with Hamming distance of 8 (part 3 of 3)

1st	2nd	3rd	4th	Assignment
K28.5	D29.7	D04.7	D27.4	OPEN_REJECT (RESERVED STOP 1)
K28.5	D29.7	D07.3	D16.7	OPEN_REJECT (RESERVED INITIALIZE 1)
K28.5	D29.7	D16.7	D04.7	OPEN_REJECT (PATHWAY BLOCKED)
K28.5	D29.7	D24.0	D01.4	OPEN_REJECT (RESERVED CONTINUE 1)
K28.5	D29.7	D27.4	D24.0	OPEN_REJECT (RETRY)
K28.5	D29.7	D29.7	D29.7	OPEN_REJECT (NO DESTINATION)
K28.5	D29.7	D30.0	D31.4	OPEN_REJECT (RESERVED INITIALIZE 0)
K28.5	D29.7	D31.4	D07.3	OPEN_REJECT (RESERVED STOP 0)
K28.5	D30.0	D01.4	D04.7	DONE (ACK/NAK TIMEOUT)
K28.5	D30.0	D02.0	D16.7	
K28.5	D30.0	D07.3	D27.4	DONE (CREDIT TIMEOUT)
K28.5	D30.0	D16.7	D01.4	DONE (RESERVED 0)
K28.5	D30.0	D24.0	D02.0	
K28.5	D30.0	D27.4	D29.7	DONE (RESERVED TIMEOUT 0)
K28.5	D30.0	D29.7	D31.4	DONE (RESERVED 1)
K28.5	D30.0	D30.0	D30.0	DONE (NORMAL)
K28.5	D30.0	D31.4	D24.0	DONE (RESERVED TIMEOUT 1)
K28.5	D31.3	D01.3	D07.0	NOTIFY (RESERVED 1)
K28.5	D31.3	D07.0	D01.3	NOTIFY (POWER LOSS EXPECTED)
K28.5	D31.3	D10.2	D10.2	NOTIFY (RESERVED 2)
K28.5	D31.3	D31.3	D31.3	NOTIFY (ENABLE SPINUP)
K28.5	D31.4	D01.4	D30.0	OPEN_REJECT (RESERVED ABANDON 3)
K28.5	D31.4	D02.0	D27.4	OPEN_REJECT (RESERVED ABANDON 0)
K28.5	D31.4	D04.7	D29.7	OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)
K28.5	D31.4	D07.3	D02.0	OPEN_REJECT (RESERVED ABANDON 2)
K28.5	D31.4	D16.7	D24.0	OPEN_REJECT (WRONG DESTINATION)
K28.5	D31.4	D27.4	D01.4	OPEN_REJECT (STP RESOURCES BUSY)
K28.5	D31.4	D29.7	D07.3	OPEN_REJECT (PROTOCOL NOT SUPPORTED)
K28.5	D31.4	D30.0	D16.7	OPEN_REJECT (RESERVED ABANDON 1)
K28.5	D31.4	D31.4	D31.4	OPEN_REJECT (BAD DESTINATION)