To: T10 Technical Committee
From: Timothy Hoglund, LSI Logic
Date: 19-April-2005
Subject: T10/05-145r0 SAS-1.1: BreakWait issue resolution

Revision History
Revision 0 (initial draft 19-April-2005)

Related Documents
sas1r09 – Serial Attached SCSI-1.1 revision 0.9
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: SAS-1.1, Responding to an OPEN address frame in the BreakWait state

Overview
Proposal 05-040r0 sought to correct a specific link layer race condition present when a SAS device sends a BREAK primitive while concurrently receiving an OPEN_REJECT primitive.

While the 05-040r0 proposal provided a viable solution, it did not comprehensively address other link layer race conditions subsequently identified.

Proposals 05-086r0 and 05-093r1 were offered as alternative remedies to identified link layer race conditions. Of these two proposals, 05-086r1 was recommended for inclusion into SAS-1.1 and is incorporated into the text of sas1r09.

However, subsequent analysis of the method offered by proposal 05-086r1 suggest that further link layer race conditions still exist.

A particularity of the SL and XL state machines is that the BREAK primitive sequence is used without differentiation, i.e. 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 several 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.

This proposal seeks to correct deficiencies in previously discussed methods by introducing a BREAK_RESPONSE primitive sequence that SAS-1.1 devices may choose to employ (with the expectation that SAS-2.0 devices shall be required to employ).

Related appoaches

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

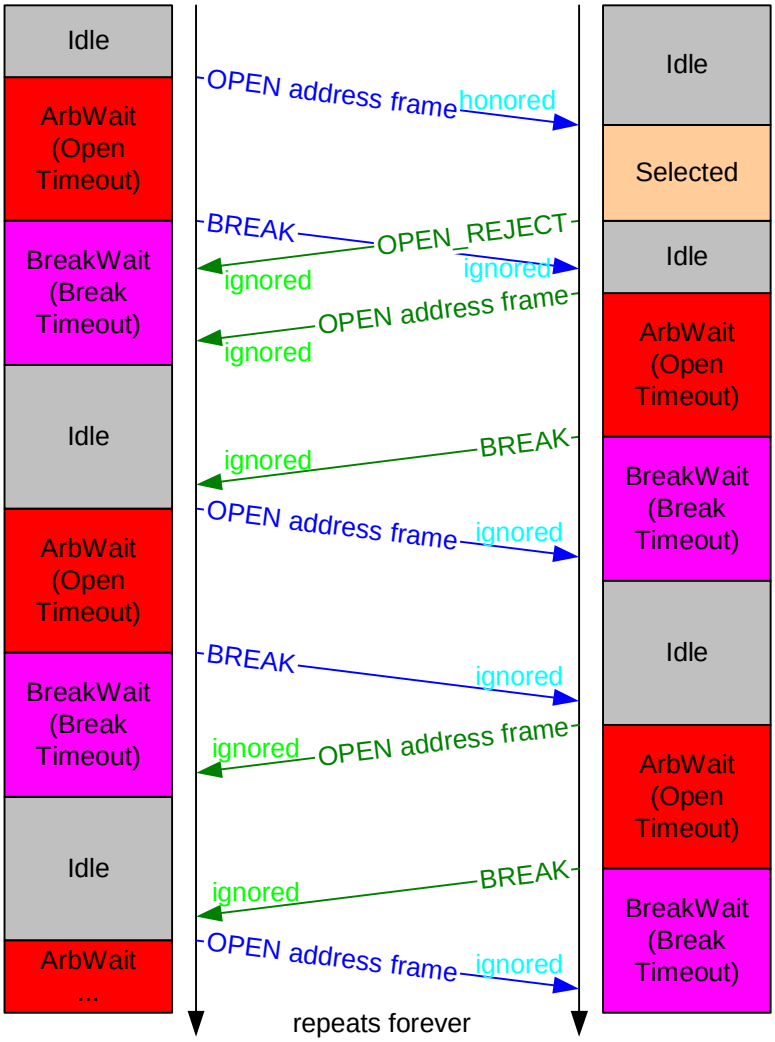

Figure 1: BREAK crossing OPEN_REJECT

Note this ping-pong scenario can also occur when BREAK and CLOSE cross on the wire as in Figure 2.

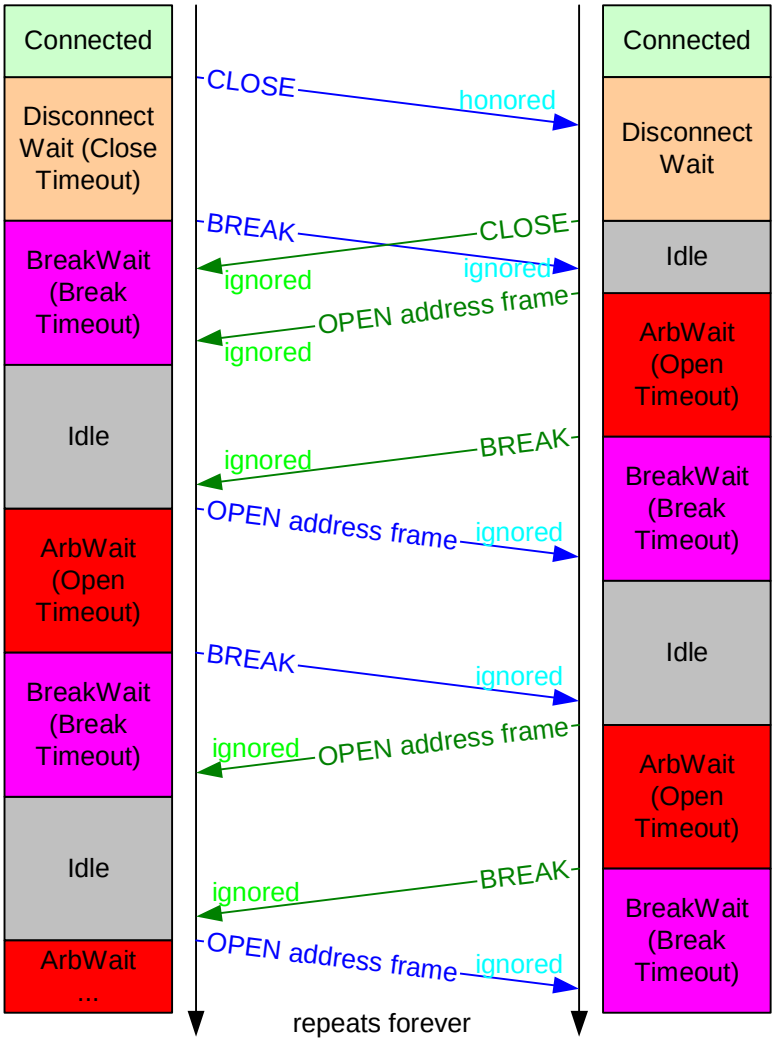

Figure 2: BREAK crossing CLOSE

Proposal 05-040r0 suggested that the left-hand phy transition from BreakWait to Idle upon reception of OPEN_REJECT to prevent this loop as demonstrated in Figure 3.

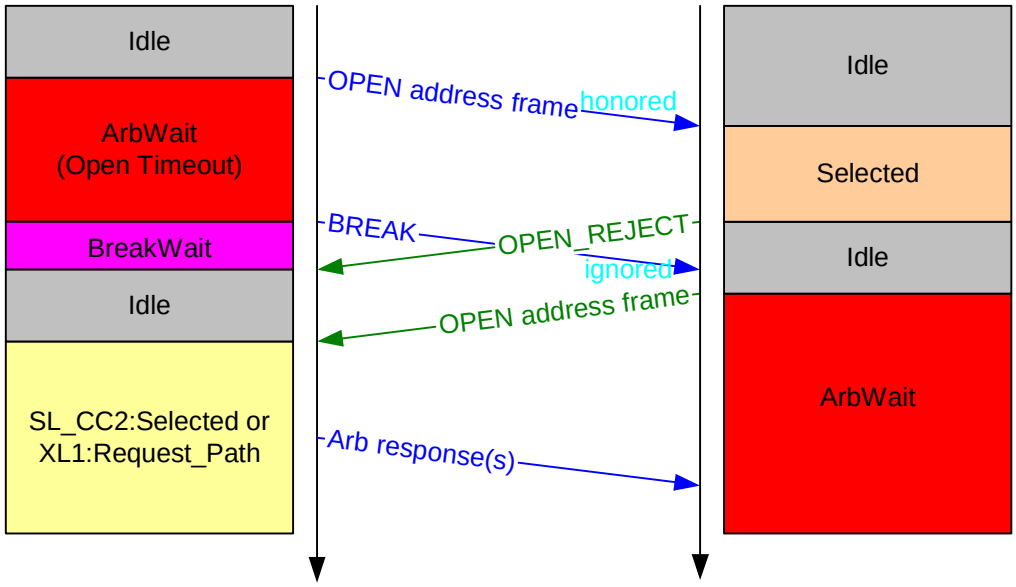

Figure 3: Transition to Idle when OPEN_REJECT received in BreakWait

Note that while 05-040r0 did not address it, this method could be extended to cover the BREAK crossing CLOSE case as shown in Figure 4.

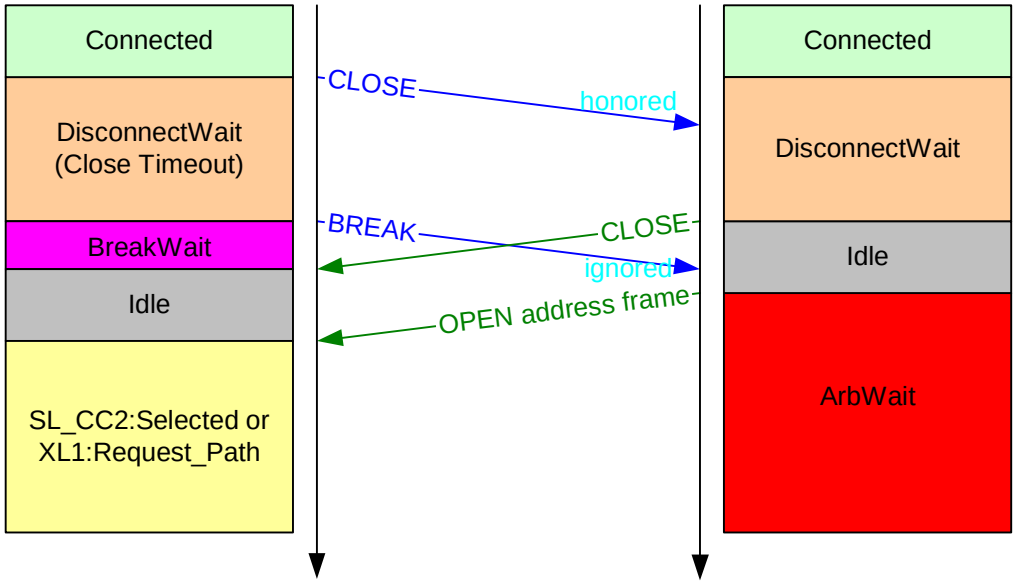

Figure 4: Transition to Idle when CLOSE received in BreakWait

An identified shortcoming of the method shown above is that it can degenerate into the ping-pong cases if the OPEN_REJECT primitive or CLOSE primitive sequence are subjected to bit errors and therefore not properly received.

Proposal 05-086r0 called for a combination of time limits and additional state transitions to avoid link layer race conditions.

Time limits (05-086r0 Table 3) were established to set an upper bound on how long a phy could wait before responding to a received OPEN address frame, CLOSE, or BREAK (also how long an expander phy can wait to send OPEN_REJECT after transmitting AIP). These time limits were set to 0.5 ms vs the 1ms Open, Break, and Close Timeout values to insure that in the absence of application induced breaks (Stop Arb request) no race conditions will occur.

Additional state transitions were included to ensure that BREAKs caused by the Stop Arb request (which are completely asynchronous to phy state) also properly resolve. However, these transitions will not always properly occur in the presence of bit errors on the wire – this is one reason why proposal 05-093r1 was favored over this approach.

Proposal 05-093r1 sought to remedy the above described link layer race conditions by causing a SAS device in the BreakWait state respond to a received OPEN address frame with OPEN_REJECT(RETRY) if an end device or with OPEN_REJECT(WAITING FOR BREAK) if an expander device as illustrated by Figure 5.

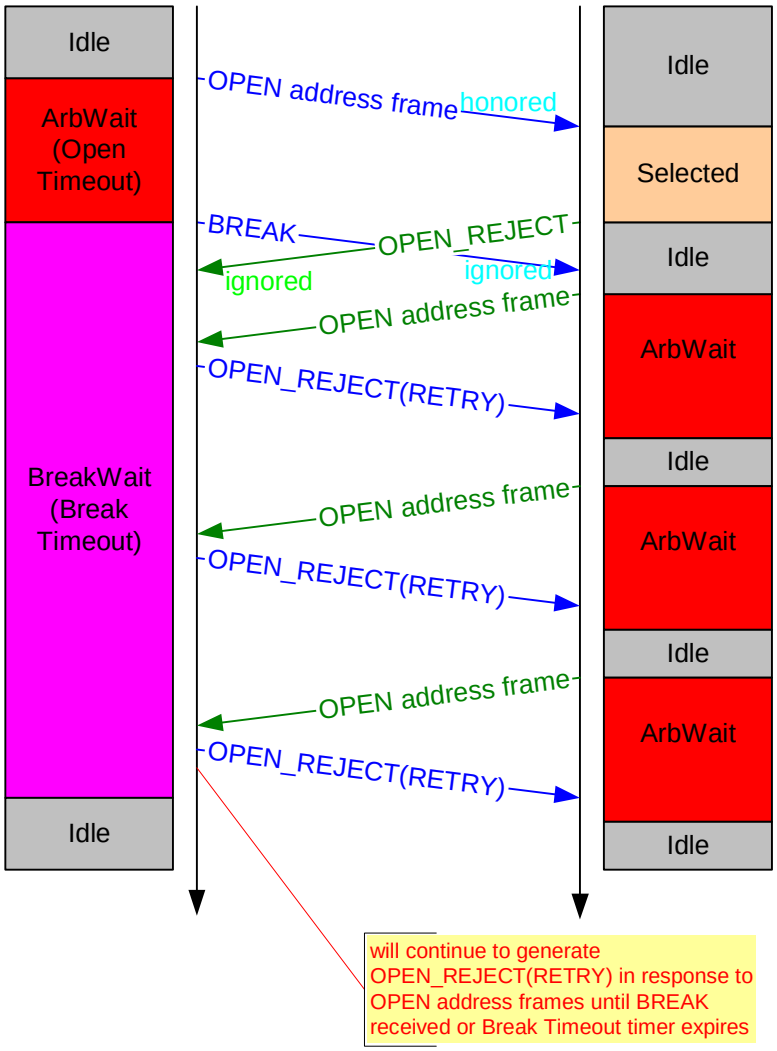

Figure 5: OPEN_REJECT(RETRY) response to OPEN address frame while in BreakWait

While this method prevents any link layer race conditions for BREAK crossing either OPEN_REJECT or CLOSE, it unfortunately does more harm than good when BREAK crosses an OPEN address frame as illustrated by Figure 6 and subsequent description.

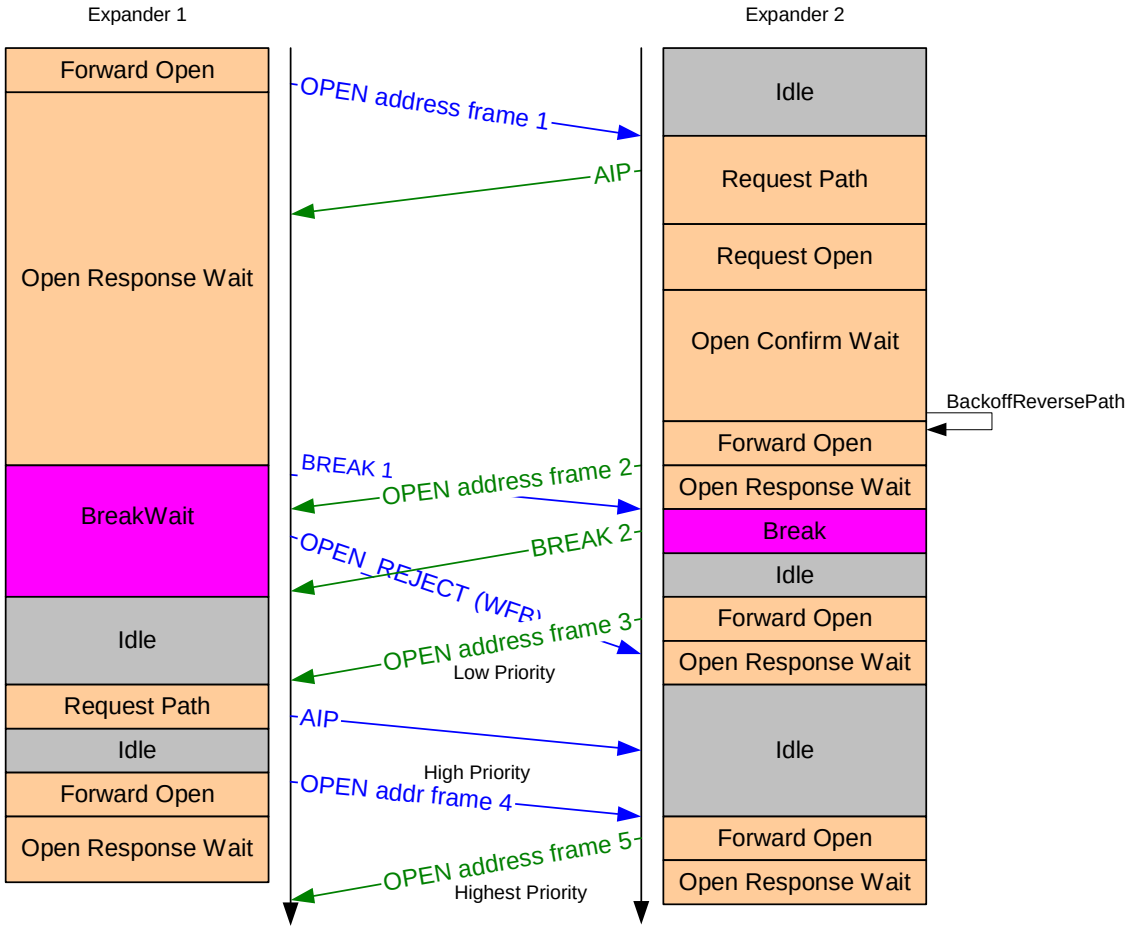

Figure 6: Issues introduced by 05-093r1 OPEN_REJECT while in BreakWait

Note that while a BackoffReversePath case is shown, other scenarios could also lead to similar results such as lost arbitration.

Figure 6 description

Expander 1 sends Break #1 coincident with Expander 2 sending OPEN address frame #2.

Thereafter in accordance with 05-093r1, Expander 1 sends OPEN_REJECT(WAITING FOR BREAK) in response to received OPEN address frame #2 but coincidently with Expander 2 sending BREAK #2.

- Expander 2 considers BREAK #1 to apply to OPEN address frame #2
- Expander 1 considers BREAK #2 to terminate the connection request associated with OPEN address frame #1
- Endpoints at either end properly consider OPEN address frame #1 and #2 as terminated

However, Expander 2 has already begun to forward OPEN address frame #3 at the time it receives/recognizes OPEN_REJECT(WAITING FOR BREAK).

- Expander 2 will terminate the connection request associated with OPEN address frame #3 due to the OPEN_REJECT(WAITING FOR BREAK).
- Expander 1 will consider OPEN address frame #3 as a valid connection request and arbitrate for path resources.

Expander 1 arbitrates which leads to one of the following, depending on the arbitration priority of OPEN address frame #3:

1. Expander 1 ArbWon cases

    a) OPEN_ACCEPT transmitted to idle Expander 2
    → Expander 2 will ignore the OPEN_ACCEPT – endpoint should timeout

    b) OPEN_ACCEPT transmitted to Expander 2 forwarding an OPEN address frame
    → OPEN_ACCEPT will be misrouted

2. Expander 1 ArbLost cases

    a) OPEN address frame #4 > OPEN address frame #5
    → Expander 2 will correctly backoff

    b) OPEN address frame #4 < OPEN address frame #5
    → neither expander will backoff – OPEN timeout at endpoint should resolve

3. Expander 1 ArbReject cases

    a) OPEN_REJECT(any) sent to idle Expander 2
    → Expander 2 ignores

    b) OPEN_REJECT crosses OPEN address frame
    → Expander 2 associates the OPEN_REJECT with the wrong OPEN address frame and the potential to repeat this cycle indefinitely exists.

Because of the potential for the OPEN_REJECT(RETRY) or OPEN_REJECT(WAITING FOR BREAK) to be associated with the wrong OPEN address frame, proposal 05-093r1 is deemed too problematic to remain within SAS-1.1.

Note that without the OPEN_REJECT(RETRY) or OPEN_REJECT(WAITING FOR BREAK) response to an OPEN address frame while in the BreakWait state, no problem exists when OPEN address frames cross with BREAK as illustrated in Figure 7.

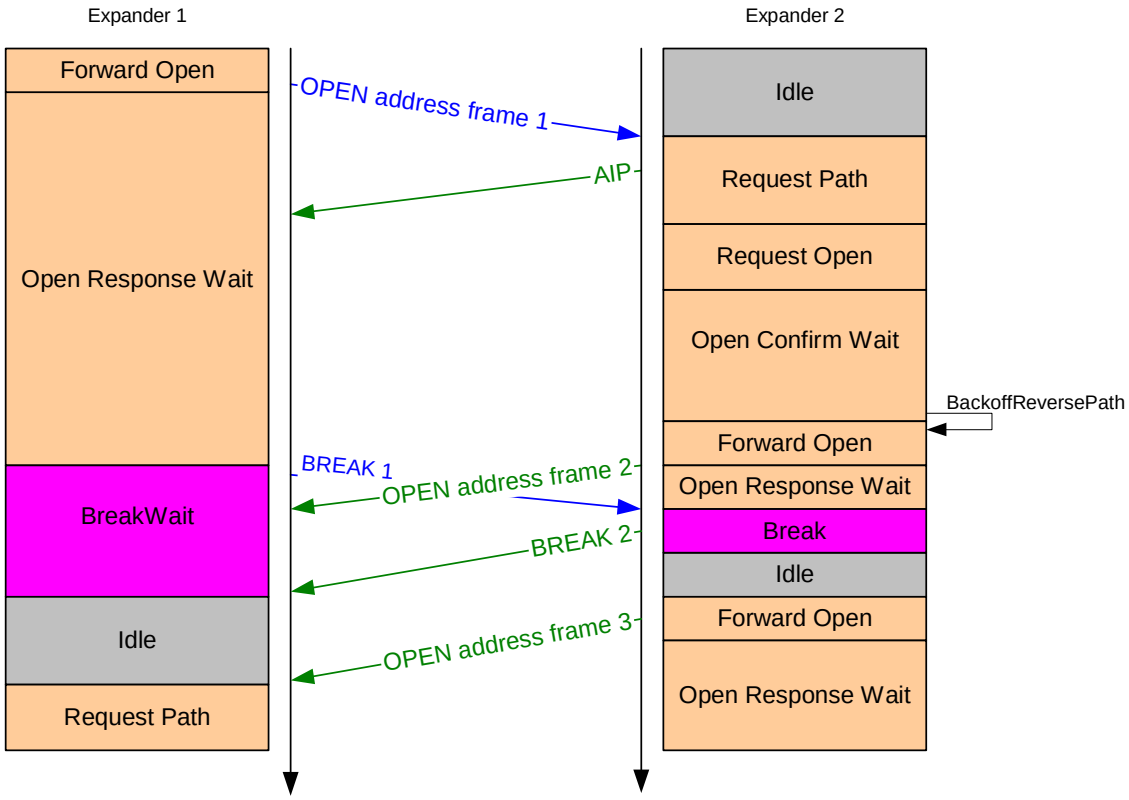

Figure 7: BREAK crossing OPEN address frame without OPEN_REJECT response

Note that in this case there is no disagreement about the state of any of the connection requests (OPEN address frames) by either expander – the disagreement is introduced by the OPEN_REJECT in 05-093r1.

Proposal Summary

1) remove changes introduced by 05-040r0 and 05-093r1
    a. remove OPEN_REJECT (RETRY) response to OPEN ADDRESS frame while in SL_CC5:BreakWait state
    b. remove OPEN_REJECT(WAITING FOR BREAK) response to OPEN ADDRESS frame while in XL10:Break_Wait
    c. change OPEN_REJECT(WAITING FOR BREAK) to OPEN_REJECT(RESERVED STOP 0)
    d. state machine diagram changes related to the above
2) add new redundant primitive sequence BREAK_RESPONSE
    a. select from Annex J table of primitives with Hamming distance of at least 8, suggest K28.5 D02.0 D29.7 D16.7
3) update SL and XL state descriptions and transitions
    a. SL and XL use BREAK primitive sequence to originate (abandon a connection request or break a connection).
    b. SL and XL use BREAK_RESPONSE primitive sequence to respond to a BREAK primitive sequence (“should” with NOTE suggesting future versions may require), i.e. SL_CC6:Break and XL9:Break transmit BREAK_RESPONSE primitive sequence.
    c. SL and XL respond to BREAK while in SL_CC0:Idle and XL0:Idle with BREAK_RESPONSE(“should” with NOTE suggesting future versions may require).
    d. SL and XL transition from BreakWait to Idle upon receiving BREAK or BREAK_RESPONSE.
4) add a method to negotiate for the use of BREAK_RESPONSE on a point-to-point basis via an Identify Address frame bit field.

Figure 8 shows how the BREAK_RESPONSE solves the BREAK crossing with OPEN_REJECT.

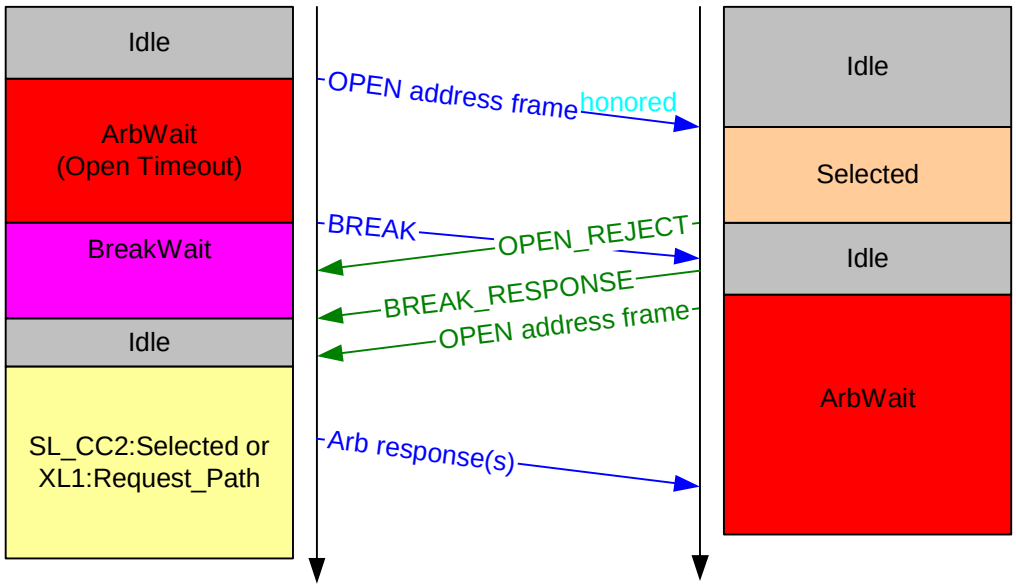

Figure 8: BREAK_RESPONSE following BREAK crossing OPEN_REJECT

Figure 9 shows how the BREAK_RESPONSE solves the BREAK crossing with CLOSE.

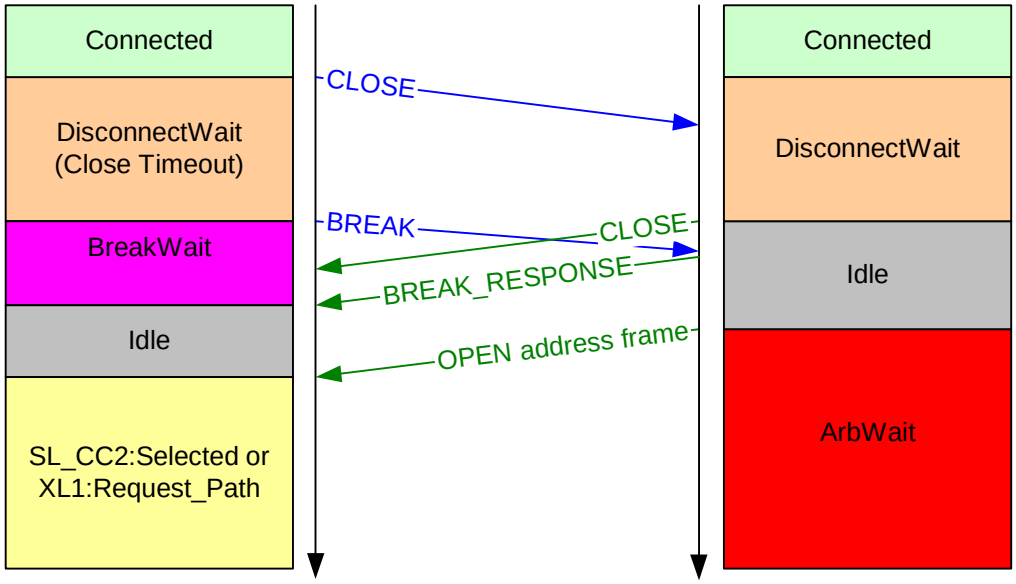

Figure 9: BREAK_RESPONSE following BREAK crossing CLOSE

Proposal Detail

4.5 I_T nexus loss

When a SAS port receives OPEN_REJECT (NO DESTINATION), OPEN_REJECT (PATHWAY BLOCKED), ~~OPEN_REJECT (WAITING FOR BREAK),~~ or an open connection timeout occurs in response to a connection request, it shall retry the connection request until:

7.2.2 Primitive Summary, Table 62 — Primitives not specific to type of connection (part 2 of 2)
change OPEN_REJECT(WAITING FOR BREAK) to OPEN_REJECT(RESERVED STOP 0)
remove E in from column

7.2.3 Primitive Encodings, Table 65 — Primitive encoding for primitives not specific to type of connection (part 2 of 2)
change OPEN_REJECT(WAITING FOR BREAK) to OPEN_REJECT(RESERVED STOP 0)

7.2.5.11 OPEN_REJECT, Table 75 — OPEN_REJECT retry primitives
change OPEN_REJECT(WAITING FOR BREAK) to OPEN_REJECT(RESERVED STOP 0)
remove OPEN_REJECT(RETRY) clause b)

b) a SAS phy received an OPEN address frame while the SL_CC state machine is in the SL_CC5:BreakWait state (see 7.14.4.7) for OPEN_REJECT(RETRY)

Appendix J, Table J.1 — Primitives with Hamming distance of 8 (part 1 of 3)
add BREAK_RESPONSE (K28.5 D02.0 D29.7 D16.7)

Appendix J, Table J.1 — Primitives with Hamming distance of 8 (part 3 of 3)
change OPEN_REJECT(WAITING FOR BREAK) to OPEN_REJECT(RESERVED STOP 0)

7.14.1 SL State Machines Overview

Figure 141 — SL (link layer for SAS phys) state machines (part 1)
remove SL_RA OPEN Address Frame Received message from the SL_RA state machine to the SL__CC5:BreakWait state.

Figure 142 — SL (link layer for SAS phys) state machines (part 2)
remove Transmit OPEN_REJECT message from the SL_CC5:BreakWait state to the SL transmitter.

7.12.6 Aborting a connection request
add BREAK_RESPONSE to Table 96

7.12.8 Breaking a connection
add BREAK_RESPONSE to Table 98

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_RESPONSE message to the SL transmitter if PHY_BREAK_RESPONSE is enabled (see 7.8.2).

7.14.4.7 SL_CC5:BreakWait state
7.14.4.7.1 State description

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

Upon entry into this state, this state shall:

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

~~If an OPEN Address Frame Received message is received in this state, then this state should send a Transmit OPEN_REJECT (Retry) message to the SL transmitter.~~

~~NOTE 37 - Future versions of this standard may require that SAS devices send a Transmit OPEN_REJECT (Retry) message in response to each OPEN Address Frame Received message received while in this state.~~

7.14.4.7.2 Transition SL_CC5:BreakWait to SL_CC0:Idle

This transition shall occur after:

a) receiving a BREAK Received message;
b) receiving a BREAK_RESPONSE 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 send:

a) a Transmit BREAK message to the SL transmitter if PHY_BREAK_RESPONSE is not enabled (see 7.8.2); or
b) a Transmit BREAK_RESPONSE message to the SL transmitter if PHY_BREAK_RESPONSE 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_RESPONSE message to the SL transmitter.

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 NotReady) 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.

If a BREAK received message is received, this state shall send a Transmit BREAK_RESPONSE message to the XL transmitter if PHY_BREAK_RESPONSE is enabled (see 7.8.2).

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

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 send:

a) a Transmit BREAK message to the XL transmitter if PHY_BREAK_RESPONSE is not enabled (see 7.8.2); or
b) a Transmit BREAK_RESPONSE message to the XL transmitter if PHY_BREAK_RESPONSE 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_RESPONSE 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 send:

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

~~If an OPEN Address Frame Received message is received in this state, then this state should send a Transmit OPEN_REJECT (Waiting for Break) message to the XL transmitter.~~

~~NOTE 40 - Future versions of this standard may require that expander devices send a Transmit OPEN_REJECT (Waiting for Break) message in response to each OPEN Address Frame Received message received while in this state.~~

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_RESPONSE message is received; or
c) the Break Timeout timer expires.

7.8.2 IDENTIFY address frame

change byte 11 bits 7:0 Reserved to:

bits	7:1	Reserved
bit	0	PHY BREAK_RESPONSE CAPABLE

A PHY BREAK_RESPONSE CAPABLE bit set to one specifies that this phy is capable of transmitting BREAK_RESPONSE primitive sequence in response to BREAK received. A PHY BREAK_RESPONSE CAPABLE bit set to zero specifies that this phy is not capable of transmitting BREAK_RESPONSE primitive sequence in response to BREAK received.

If a phy both transmits an Identify Address frame with the PHY BREAK_RESPONSE CAPABLE bit set to one and receives an Identify Address frame with the PHY BREAK_RESPONSE CAPABLE bit set to one, both phys shall consider PHY_BREAK_RESPONSE enabled and shall use the BREAK_RESPONSE primitive sequence when responding to BREAK. Otherwise, both phys shall use the BREAK primitive sequence when responding to BREAK.

add PHY BREAK_RESPONSE CAPABLE bit to the list of values referred to section 4.1.3

4.1.3 Ports (narrow ports and wide ports)

add PHY BREAK_RESPONSE CAPABLE bit to the set of values transmitted during the identification sequence which are required to be the same on each phy of the wide port.

[end proposal detail]

Alternatives

1. Have SL_CC6:Break and XL9:Break continue to use BREAK, only have SL_CC0:Idle and XL0:Idle states utilize BREAK_RESPONSE when PHY_BREAK_RESPONSE is enabled. This is a less disruptive change but still solves the primary issues. Including SL_CC6:Break and XL9:Break are potentially helpful from a diagnosis perspective and should be considered however (to easily identify which device initiated the BREAK).

2. 05-086r0 method