To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 12 July 2008
Subject: 08-283r0 SAS-2 SNW-3 SATA port selector confusion

Revision history

Revision 0 (12 July 2008) First revision

Related documents

sas2r14 - Serial Attached SCSI - 2 (SAS-2) revision 14
sas2r14c - Serial Attached SCSI - 2 (SAS-2) revision 14c
08-212r3 - SAS-2 revision 14 letter ballot comment resolution as of sas2r14c (Rob Elliott, HP)

Overview

Mark Seidel (Intel) identified a problem with the COMWAKEs used in SNW-3 by SAS-2 phys confusing phys that only participate in SNW-1 into thinking that a SATA SATA port selector is present. This was carried in an editor's note for several revisions prior to sas2r14, and submitted as a letter ballot comment on sas2r14. sas2r14c includes a description of the problem, including a figure, but does not include any solutions.

Also, if two phys each support SNW-1 and SNW-3 only, or if the interconnect cannot negotiate SNW-2 successfully, they will end up running at G1 rather than G3.

SAS-2 revision 14c problem description

Figure 151 shows speed negotiation between a phy A and phy B where the phys participate in:

1) SNW-1, supported by phy B but not by phy A; and
2) SNW-2, supported by neither phy.

Phy A proceeds to SNW-3 while phy B returns to the OOB sequence.

Phy A and phy B detect:

a) SNW-1 invalid; and
b) SNW-2 invalid.

Phy A detects SNW-3 invalid.

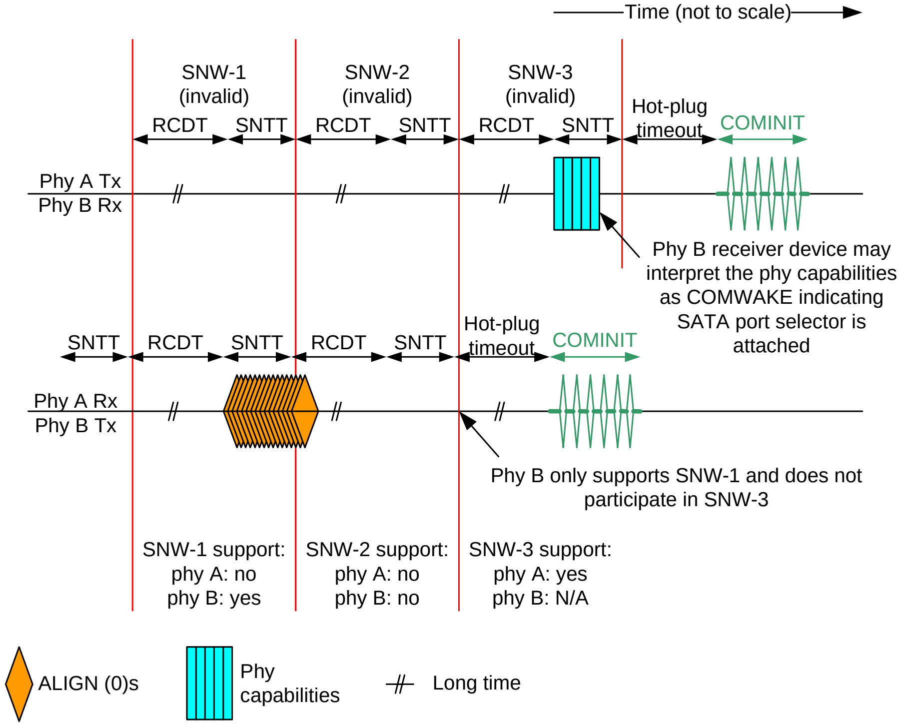

Figure 151 — SAS speed negotiation sequence (phy A: SNW-3 only, phy B: SNW-1 only)

Since a phy capabilities bit set to one is defined as COMWAKE (see table 97 in 6.7.4.2.3.3), phy B interprets the first phy capabilities bit set to one from phy A as being a COMWAKE in response to its COMINIT during the OOB sequence. This falsely identifies a SATA port selector and causes phy B to incorrectly set its ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response (see 10.4.3.10).

When phy B transmits COMINIT during SNW-1 of the next OOB sequence, phy A detects that a SAS phy is attached, not a SATA phy, and sets the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response. However, the phys may keep repeating this process after each hot-plug timeout.

An expander device originates Broadcast (Change) whenever the ATTACHED SATA PORT SELECTOR bit changes from zero to one (see 7.12), so this results in Broadcast (Changes) at hot-plug timeout intervals.

List of SNW combinations

Table 1 shows the different combinations of SNW support between two phys.

Table 1 — Combinations of SNWs (part 1 of 2)

Phy A	Phy B	Description
SNW-1	SNW-1	SNW-1 valid, SNW-2 invalid, Final-SNW (G1)
	SNW-2	SNW-1 invalid, SNW-2 invalid If phy is SAS-1: end If phy is SAS-2: SNW-3 invalid (no COMWAKE problem), end
	SNW-3	SNW-1 invalid, SNW-2 invalid If phy A is SAS-1: end If phy A is SAS-2: SNW-3 invalid, end Phy B: SNW-3 invalid (COMWAKE problem), end Suggested change: Phy B does not enter SNW-3 because it detects activity during SNW-1 but not during SNW-2
	SNW-1, SNW-2	SNW-1 valid, SNW-2 invalid, Final-SNW (G1)
	SNW-2, SNW-3	SNW-1 invalid, SNW-2 invalid If phy A is SAS-1: end If phy A is SAS-2: SNW-3 invalid, end Phy B: SNW-3 invalid (COMWAKE problem), end Suggested change: Phy B does not enter SNW-3 because it detects activity during SNW-1 but not during SNW-2
	SNW-1, SNW-3	SNW-1 valid, SNW-2 invalid, Final-SNW (G1)
	SNW-1, SNW-2, SNW-3	SNW-1 valid, SNW-2 invalid, Final-SNW (G1)
SNW-2	SNW-2	SNW-1 invalid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
	SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 invalid, end
	SNW-1, SNW-2	SNW-1 invalid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
	SNW-2, SNW-3	SNW-1 invalid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
	SNW-1, SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 invalid, end
	SNW-1, SNW-2, SNW-3	SNW-1 invalid, SNW-2 valid, SNW-3 valid, Final-SNW (G2)

Table 1 — Combinations of SNWs (part 2 of 2)

Phy A	Phy B	Description
SNW-3	SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 valid, Train-SNW (G3)
	SNW-1, SNW-2	SNW-1 invalid, SNW-2 invalid, SNW-3 invalid, end
	SNW-2, SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 valid, Train-SNW (G3)
	SNW-1, SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 valid, Train-SNW (G3)
	SNW-1, SNW-2, SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 valid, Train-SNW (G3)
SNW-1, SNW-2	SNW-1, SNW-2	SNW-1 valid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
	SNW-1, SNW-3	SNW-1 valid, SNW-2 invalid, Final-SNW (G1))
	SNW-2, SNW-3	SNW-1 invalid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
	SNW-1, SNW-2, SNW-3	SNW-1 valid, SNW-2 valid, SNW-3 invalid, Final-SNW (G2)
SNW-1, SNW-3	SNW-1, SNW-3	SNW-1 valid, SNW-2 invalid, Final-SNW (G1) (but G3 is desired) Suggested change: Prohibit this combination.
	SNW-2, SNW-3	SNW-1 invalid, SNW-2 invalid, SNW-3 valid, Train-SNW (G3)
	SNW-1, SNW-2, SNW-3	SNW-1 valid, SNW-2 invalid, Final-SNW (G1) (but G3 is desired) Suggested change: Prohibit this combination.
SNW-2, SNW-3	SNW-2, SNW-3	SNW-1 invalid, SNW-2 valid, SNW-3 valid, Train-SNW (G3)
	SNW-1, SNW-2, SNW-3	SNW-1 invalid, SNW-2 valid, SNW-3 valid, Train-SNW (G3)

Proposed solution

The main concern is expanders sending out Broadcast (Change)s forever as they think a SATA port selector is attached, then not attached, then attached, etc. This will only happen if the expander phy is SAS-1 and configured to support SNW-1 only.

An end device configured for SNW-1 only could also be confused, but it will only confuse itself - it may internally generate the equivalent of Broadcast (Change), but this is not visible on the SAS physical link and could be ignored by the standard.

Several rules can be added to improve this situation.

1. **Detect D.C. idle (or not) during SNW-1 and SNW-2**.

This keeps a SAS-1.1 expander phy configured as SNW-1 only from being confused by a SAS-2 phy.

a) The SAS-2 phy should attempt to detect whether or not D.C. idle is occurring during SNW-1 and SNW-2, even if it is not participating (transmitting). (It could attempt to fully lock on ALIGN (0), but this might not work in SNW-2.)

If it detects lack of D.C. idle in SNW-1, but then detects D.C. idle in SNW-2, it should assume that the attached phy is is not proceeding to SNW-3, and avoid going to SNW-3 itself. This way, it won't send SNW-3 COMWAKEs that the attached phy could confuse with a SATA port selector (as the attached phy proceeds back to SP0).

That helps for SAS-1.1 phys, which would never support SNW-3 and always go back to SP0 in this situation. SAS-2 phys sit through SNW-3 in this case

Below, SAS-2 phys will be prohibited from attempting a combination of SNW-1 and SNW-3.

If lack of D.C. idle is detected in SNW-2, then it always proceeds to SNW-3.

2. **Prohibit attempts to support SNW-1, not support SNW-2, and support SNW-3**.

Speed negotiation cannot reach SNW-3 if SNW-1 passes and SNW-2 fails. SNW-3 can be reached if both fail, both pass, or SNW-2 alone passes.

To avoid this, a SAS-2 phy should only be allowed to use these combinations:

a) SNW-1 only
b) SNW-2 only
c) SNW-3 only
d) SNW-1 and SNW-2
e) SNW-2 and SNW-3
f) SNW-1, SNW-2, and SNW-3

and not allowed to attempt this combination:

g) SNW-1 and SNW-3

Combination d) (SNW-1, SNW-2, and SNW-3) can also encounter problems on an interconnected supported by SAS-2 but not SAS-1.1 phys (e.g., a 10 m cable). It is possible that SNW-1 succeeds and SNW-2 fails on the interconnect, which leads to running at 1.5 Gbps. However, SAS-2 phys may be able to run 3 Gbps or 6 Gbps over that interconnect using on SNW-3 and Train-SNW. The phy cannot attempt combination g) without risk of not getting to SNW-3.

The SAS physical WG is discussing adding a reverse key to the Mini SAS 4x connectors to block a long passive cable from being attached to a current receptacle. That would avoid the need for a SAS-2 phy to ever use combination g); it can just use combination d), assured that the other phy will also be SAS-2 if they happen to have a long cable between them. If the phy supports SNW-2 in this situation, then it should also support SNW-3 to negotiate 3 Gbps without SSC.

3. **Sit through SNW-3 before restarting the OOB sequence if only SNW-1 is attempted**

Assume the phy has been configured to support only SNW-1. If SNW-1 is invalid, then the normal algorithm is to sit through SNW-2, then start the hot-plug timeout and restart the OOB sequence. Instead, the phy should sit through both SNW-2 and SNW-3 before starting the hot-plug timeout and OOB sequence. This way, if the attached phy supports SNW-3, the phy will ignore the COMWAKEs. The phy should respond to an incoming COMINIT during that time.

Suggested changes

6.7.4.2 SAS speed negotiation sequence

6.7.4.2.1 SAS speed negotiation sequence overview

The SAS speed negotiation sequence establishes communications between the two phys of a physical link at the highest possible transmission rate.

The SAS speed negotiation sequence is a peer-to-peer negotiation technique that does not assume initiator and target (i.e., host and device) roles. The rules for speed negotiation are the same for both participating phys.

The SAS speed negotiation sequence consists of a set of speed negotiation windows (SNWs). Each SNW is identified by a name (e.g., Speed Negotiation Window-1 or SNW-1).

SNWs conform to one of three defined types:

a) speed negotiation without training: SNW-1, SNW-2 and Final-SNW (see 6.7.4.2.3.2);
b) phy capabilities exchange: SNW-3 (see 6.7.4.2.3.3); and
c) speed negotiation with training: Train-SNW (see 6.7.4.2.3.4).

Many of the timing parameters used for defining the SNWs are common to multiple SNW types. All of the timing specifications for all SNW types are defined in 6.7.4.2.2.

A SAS speed negotiation sequence may or may not include all three types of SNWs. Phys may implement a subset of SNWs provided that the subset implements a valid speed negotiation sequence. SAS speed negotiation sequences are defined in 6.7.4.2.4.

The transmitter device shall use SAS signal output levels during the SAS speed negotiation sequence as described in 5.3.6.6.

6.7.4.2.4 SAS speed negotiation sequence

The SAS speed negotiation sequence consists of a set of SNWs (see 6.7.4.2.3) in the order shown in figure 152.

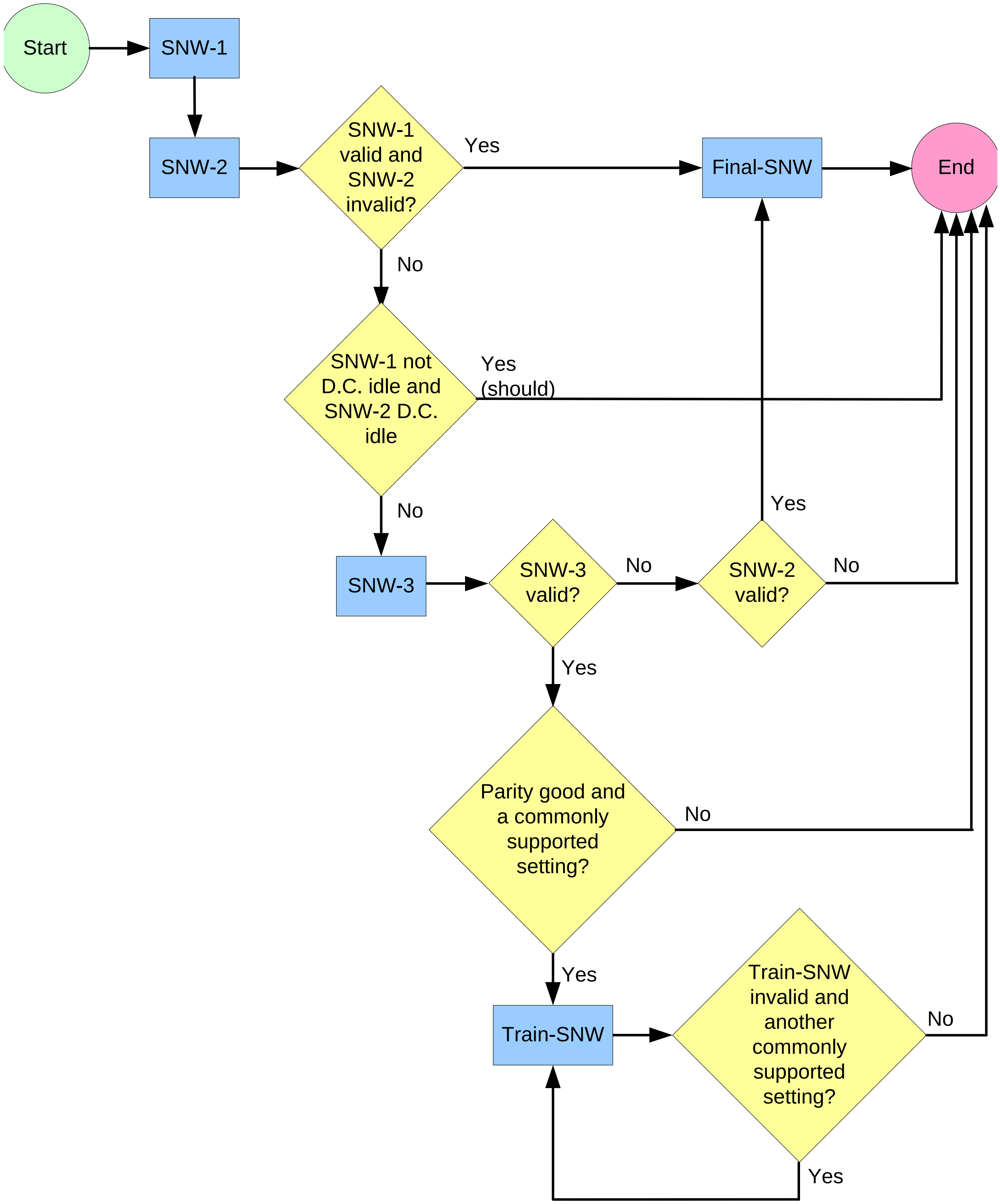

Figure 152 — SAS speed negotiation sequence SNW flowchart [changed]

Editor's Note 1: Note that if SNW-1 and SNW-2 are both invalid, a SAS-2 phy still sits through SNW-3. It only skips SNW-3 if SNW-1 is valid and SNW-2 is invalid.

To avoid causing an attached phy compliant with a previous version of this standard for misdetecting a SATA port selector, a phy should detect whether the physical link is D.C. idle during SNW-1 and SNW-2, even if it does not support that SNW. If the phy detects:

a) SNW-1 is not D.C. idle; and
b) SNW-2 is D.C. idle,

then it should end the speed negotiation sequence without progressing to SNW-3.

Train-SNW is based on the highest untried commonly supported settings based on the outgoing and incoming SNW-3 supported settings bits (see 6.7.4.2.3).

If a Train-SNW is invalid and there are additional, untried, commonly supported settings exchanged during SNW-3, then a new Train-SNW shall be performed based on the next highest, untried, commonly supported settings.

A phy reset problem occurs:

c) after Final-SNW, if Final-SNW is invalid;
d) after SNW-3, if SNW-3 is valid and the parity is bad; or
e) after a Train-SNW, if the Train-SNW is invalid and there are no additional, untried, commonly supported settings.

Phy reset problems terminate the SAS speed negotiation sequence and are counted and reported in the PHY RESET PROBLEM COUNT field in the SMP REPORT PHY ERROR LOG page (see 10.4.3.11) and the Protocol-Specific Port log page (see 10.2.8.1).

6.7.4.2.x SAS speed negotiation sequence supported SNWs

A phy shall support one of the following combinations of supported SNWs:

a) SNW-1;
b) SNW-2;
c) SNW-3;
d) SNW-1 and SNW-2;
e) SNW-1, SNW-2, and SNW-3; or
f) SNW-2 and SNW-3.

A phy shall not support the following combinations of supported SNWs:

a) SNW-1 and SNW-3; and

NOTE 1 - If SNW-1 is successful, the phy is not able to reach SNW-3 using this combination.

b) no SNWs.

If a phy supports SNW-1 only and SNW-1 is invalid, then it shall sit through both SNW-2 and SNW-3 rather than sitting through only SNW-2.

6.7.4.2.5 SAS speed negotiation sequence examples

...

Figure 151 shows speed negotiation between a phy A and phy B where phys participate in:

1) SNW-1, supported by phy B but not by phy A; and
2) SNW-2, supported by neither phy.

If phy A does not follow the recommendation to detect D.C. idle described in 6.7.4.2.4, then phy ~~Phy~~ A proceeds to SNW-3 while phy B returns to the OOB sequence.

Phy A and phy B detect:

a) SNW-1 invalid; and
b) SNW-2 invalid.

Phy A detects SNW-3 invalid.

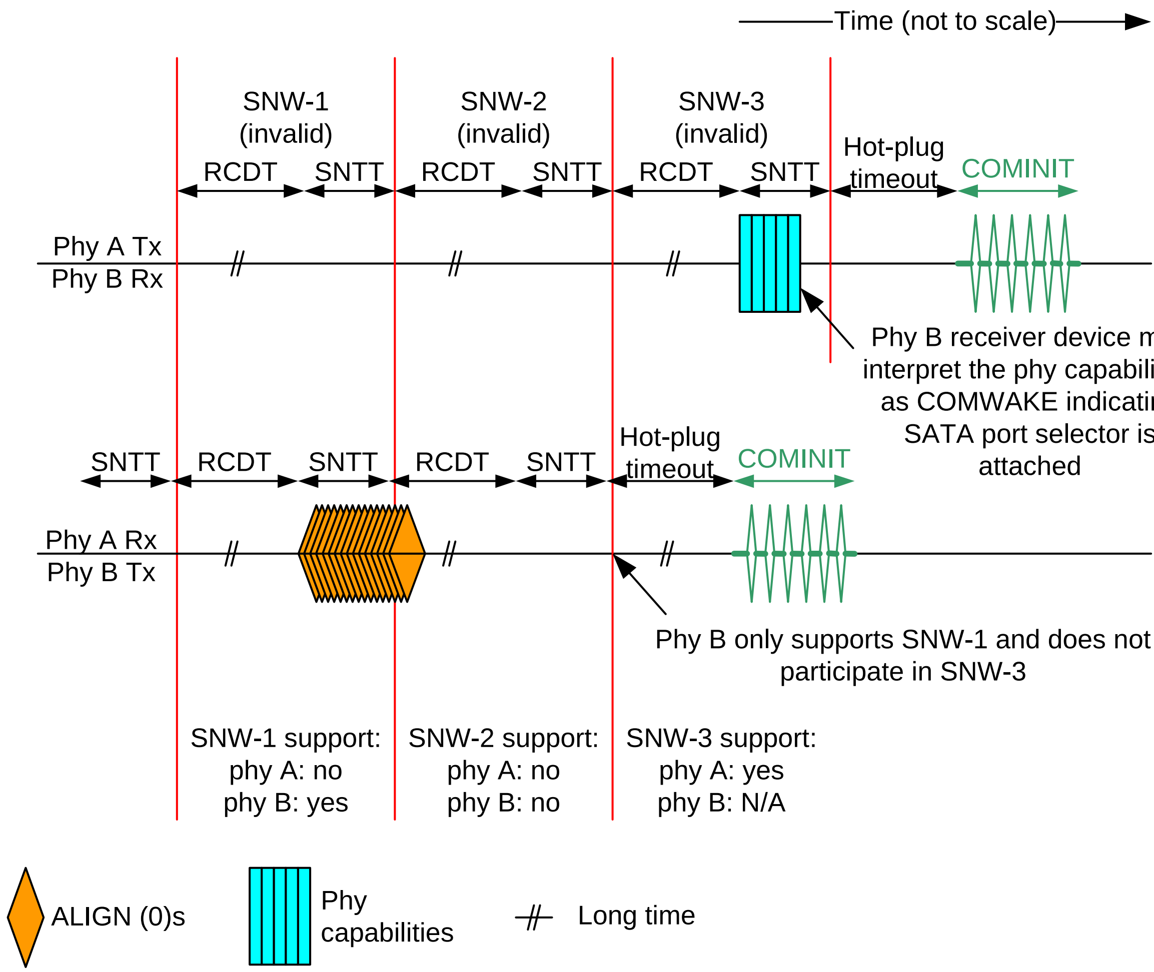

Figure 151 — SAS speed negotiation sequence (phy A: SNW-3 only without D.C. idle detection, phy B: SNW-1 only) [no changes]

Since a phy capabilities bit set to one is defined as COMWAKE (see table 97 in 6.7.4.2.3.3), phy B interprets the first phy capabilities bit set to one from phy A as being a COMWAKE in response to its COMINIT during the OOB sequence. This falsely identifies a SATA port selector and causes phy B to incorrectly set its ATTACHED SATA PORT SELECTOR bit to one in the SMP DISCOVER response (see 10.4.3.10).

When phy B transmits COMINIT during SNW-1 of the next OOB sequence, phy A detects that a SAS phy is attached, not a SATA phy, and sets the ATTACHED SATA PORT SELECTOR bit to zero in the SMP DISCOVER response. However, the phys may keep repeating this process after each hot-plug timeout. An expander device originates Broadcast (Change) whenever the ATTACHED SATA PORT SELECTOR bit changes from zero to one (see 7.12), so this results in Broadcast (Changes) at hot-plug timeout intervals.

If phy A does follow the recommendation to detect D.C. idle described in 6.7.4.2.4, then phy A and phy B both return to the OOB sequence.

Phy A and phy B detect:

a) SNW-1 invalid; and
b) SNW-2 invalid.

Phy A detects SNW-3 invalid.

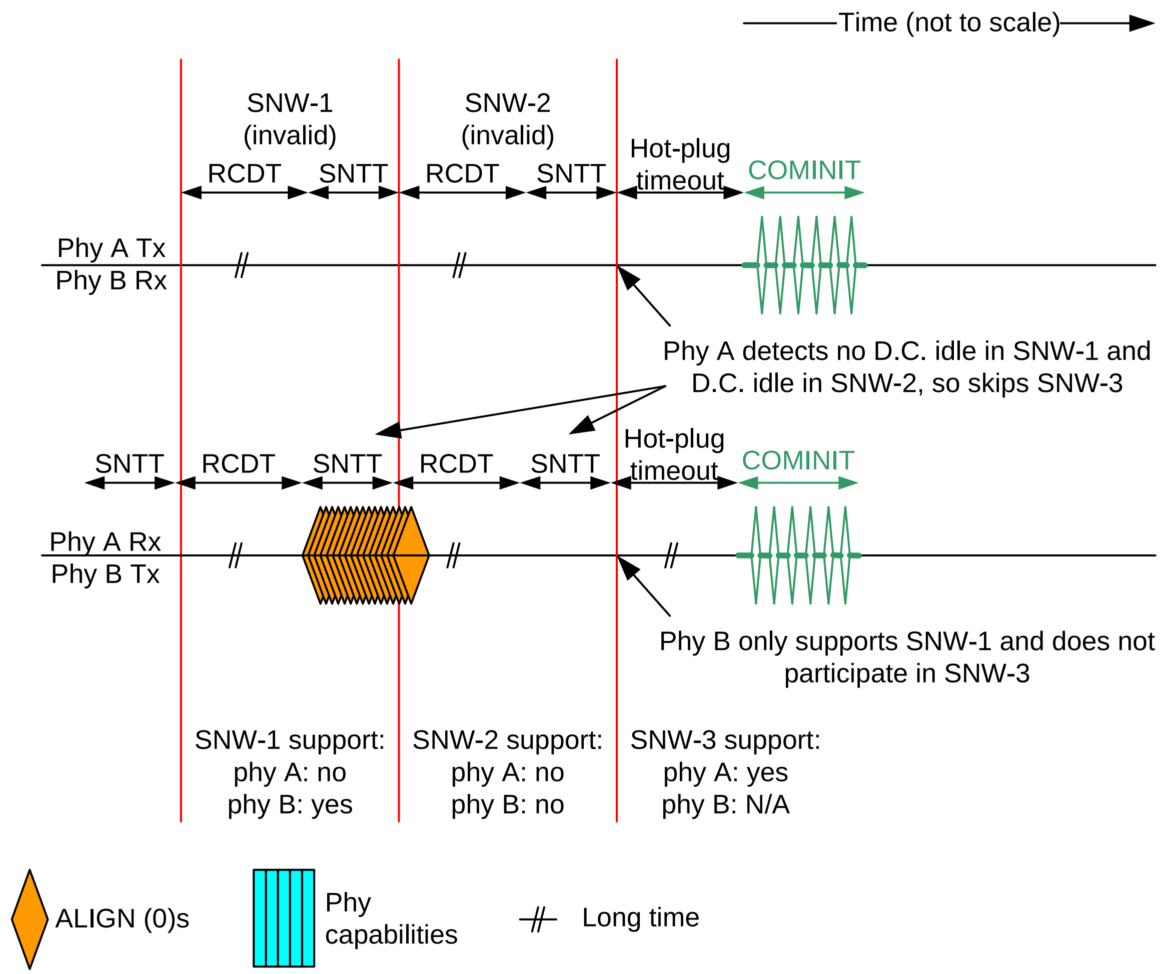

Figure 151 — SAS speed negotiation sequence (phy A: SNW-3 only with D.C. idle detection, phy B: SNW-1 only)