To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 22 May 2006
Subject: 05-322r4 SAS-2 Wide port simultaneous connection recommendations

Revision history

Revision 0 (2 November 2005) First revision

Revision 1 (11 November 2005) Incorporated comments from November 2005 SAS protocol WG - assume that a wide SSP target port also has initiator capability and can perform discover process on its own, avoiding the need to define a mode page to configure destination groups and worry about how multiple initiators will coordinate configuring the target.

Revision 2 (4 March 2006) Incorporated comments from January 2006 SAS protocol WG - added some example figures.

Revision 3 (3 May 2006) Incorporated comments from March 2006 SAS protocol WG.

Revision 4 (18 May 2006) Incorporated comments from May 2006 SAS protocol WG.

Related documents

sas2r00 - Serial Attached SCSI - 2 revision 00

Overview

SAS-1.1 includes no guidance about how a wide SSP port should decide how many connections to attempt to open at one time. After a wide SSP port opens the maximum number of connections that can be supported on a particular physical link, additional requests will end up waiting at some expander for the earlier connections from the same SSP port to complete, tying up pathways to that expander that could be used by other SAS ports.

A wide SSP port should decide how many connections to request at one time based on:

a) its own port width
b) the minimum port width on the pathway between it and the destination port
c) the status of other connections from itself using the same potential pathways to the same or different destination ports (to avoid congesting the fabric with requests that wait on other requests from itself)
d) how busy it is (if one phy suffices for all the traffic, then it might not bother with more than one connection at a time even if more are possible)

A wide STP port can also make this decision, if it is capable of opening connections to more than one STP port at a time (although it cannot open more than one connection to the same other STP port).

A set of STP target ports in an expander device can also make this decision, since the expander device knows they are sharing a wide link and can feed back information to them to collectively make this decision.

Suggested implementation

A recommended implementation (outside the scope of the standard) is that the port classifies all the destination ports into *destination groups* based on the shared wide links needed to access them. It knows the maximum number of connections possible to a destination group based on the width of the wide links on the pathway to that point. A destination port is commonly in more than one destination group.

For example, in figure 1, the initiator port considers all the expander devices and target ports to be part of a destination group sharing the 4-wide link. A, B, C, D, and X are in a separate destination group sharing a 3-wide link. E, F, G, H, and Y are in another destination group sharing a 3-wide link. I and J are just in the main destination group (each could be considered part of its own 2-wide destination group as well, if that is a convenient way to keep track of destination port widths).

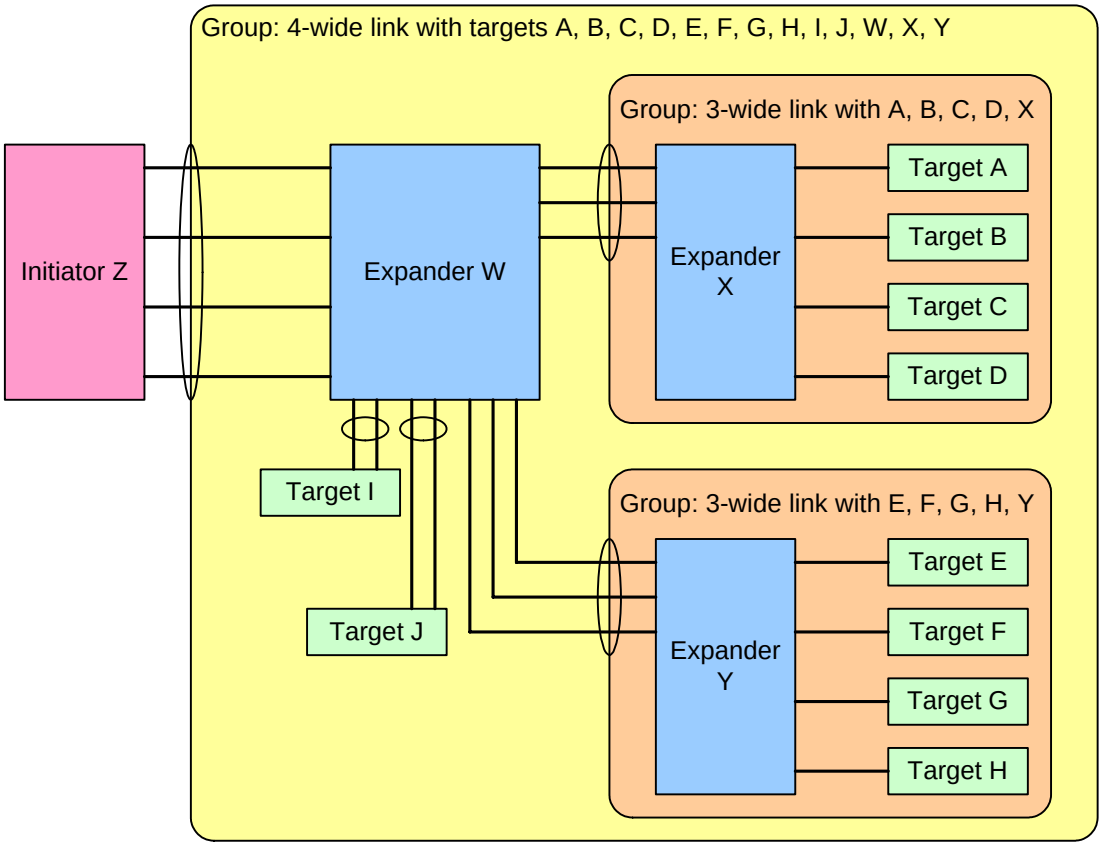

Figure 1 — Destination group example (perspective from initiator port)

The initiator port follows these rules:

a) Do not attempt to establish more connections to a destination group than the width of any of the wide links used to talk to those destination ports, so the connection request does not sit waiting on other connections involving the port to complete. Although waiting does not result in a deadlock (since the connections are expected to be regularly closed), it wastes bandwidth that could be used more productively establishing connections to other destinations.
b) Keep track of the number of connections open to each destination group. When the port wants to establish a connection to a destination, it first checks to ensure that at least one logical link is available to that destination. If so, it makes the connection request; if not, it tries a destination that is not a member of a currently fully utilized destination group.

For example, if Z is connected to A, B, and C, it should not attempt a connection to D (because the 3-wide group to which it is a member is exhausted); instead, it should favor a connection to another destination that does not belong to a group that is exhausted.

The scheme is imperfect, since an incoming connection request could still arrive that consumes another logical link in the wide link after the port already sent its outgoing request on one of the logical links. In this case, the port is not advised to try to cancel its outgoing connection request with a BREAK; it just lets it sit and congest. This impairs performance but does not lead to starvation, livelock, or deadlock.

Also, the scheme does not account for other ports making connection requests in the fabric - the basic SAS rules requiring connections to be regularly closed must suffice. In a closed environment like dual RAID controllers sharing a set of disk drives, an initiator may choose to limit its requests based on knowledge of what the other initiators are doing. Targets are unlikely to have that coordination ability.

The proposed rules to allow for/suggest this implementation are minimal.

Suggested changes

7.12.2 Opening a connection

7.12.2.1 Connection request

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

A wide port should not attempt to establish more connections to a destination port than the destination port width or the width of the narrowest physical link on the pathway to the destination port. A wide port should not attempt to establish more connections than the width of the narrowest common physical link on the pathways to the destination ports of those connections. Additional requirements for STP connection requests are defined in 7.17.6. Additional requirements for SMP connection requests are defined in 7.18.x.

Figure 2 shows an example of the simultaneous connection recommendations for wide ports.

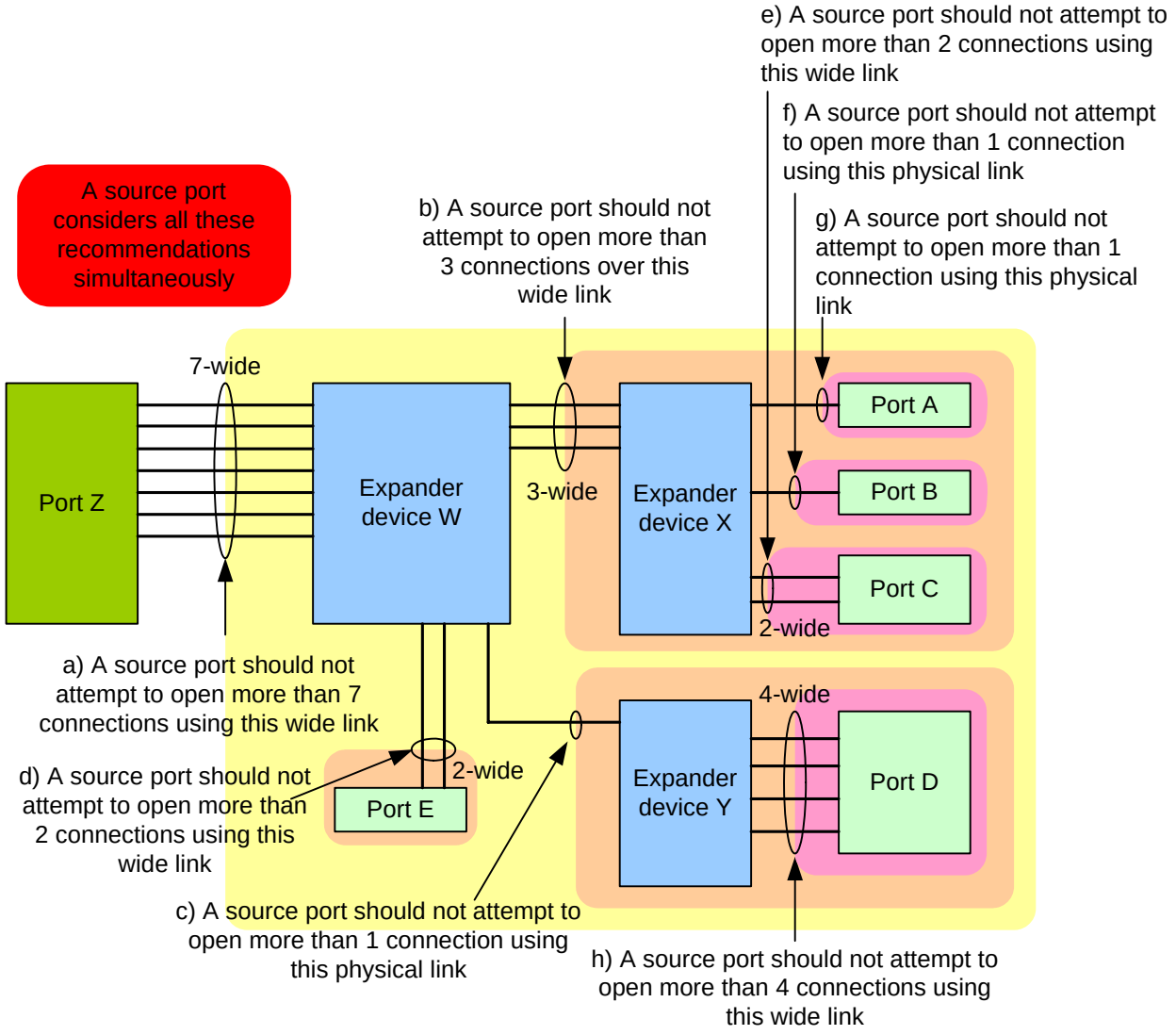

Figure 2 — Example simultaneous connection recommendations for wide ports

In figure 2, some of the recommendations are combined as follows:

a) Recommendations a), b), and e) together mean port Z should not attempt to open more than 2 connections to port C;

b) Recommendations a), b), e), f), and g) together mean that if port Z has 2 connections open to ports A, B, and X, it should not attempt to open more than 1 connection to port C. If it has 6 connections open to ports A, B, D, E, W, X, and Y, it should not attempt to open more than 1 connection to port C and

c) Recommendations a), c), and h) together mean port Z should not attempt to open more than 1 connection to port D. If it has a connection open to port Y, it should not attempt to open another connection to port D until the first connection is closed.

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

After transmitting an OPEN address frame, the source phy shall initialize and start a 1 ms Open Timeout timer. Whenever an AIP is received, the source phy shall reinitialize and restart the Open Timeout timer. Source phys are not required to enforce a limit on the number of AIPs received before aborting the connection request. When any connection response is received, the source phy shall reinitialize the Open Timeout timer. If the Open Timeout timer expires before a connection response is received, the source phy shall transmit BREAK to abort the connection request (see 7.12.5).

The OPEN address frame flows through expander devices onto intermediate physical links. If an expander device on the pathway is unable to forward the connection request, it returns OPEN_REJECT (see 7.12.4). If the OPEN address frame reaches the destination, it returns either OPEN_ACCEPT or OPEN_REJECT unless the OPEN address frame passed an OPEN address frame from the destination with higher arbitration priority (see 7.12.3). Rate matching shall be used on any physical links in the pathway with negotiated physical link rates that are faster than the requested connection rate (see 7.13).

7.12.2.2 Results of a connection request

After a phy transmits an OPEN address frame, it shall expect one or more of the results listed in table 1.

Table 1 — Connection Results of a connection request

Result	Description
Receive AIP	Arbitration in progress. When an expander device is trying to open a connection to the selected destination port, it returns an AIP to the source phy. The source phy shall reinitialize and restart its Open Timeout timer each time it receives an AIP. AIP is sent by an expander device while it is internally arbitrating for access to an expander port.
Receive OPEN_ACCEPT	Connection request accepted. OPEN_ACCEPT is transmitted by the destination phy.
Receive OPEN_REJECT	Connection request rejected. OPEN_REJECT is transmitted by the destination phy or by an expander device in the partial pathway. The different versions are described in 7.2.5.11. See 4.5 for I_T nexus loss handling.
Receive OPEN address frame	If AIP has been previously detected, this indicates an overriding connection request. If AIP has not yet been detected, this indicates two connection requests crossing on the physical link. Arbitration fairness determines which one wins (see 7.12.3).
Receive BREAK	The destination phy or an expander device in the partial pathway may reply with BREAK indicating the connection is not being established.
Open Timeout timer expires	The source phy shall abort the connection request by transmitting BREAK (see 7.12.5). See 4.5 for I_T nexus loss handling.

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

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

7.16 SSP link layer

7.16.1 Opening an SSP connection

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

7.17 STP link layer

7.17.6 Opening an STP connection

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

~~Wide~~A wide STP initiator port~~s~~ shall not request more than one connection at a time to a~~n~~ specific STP target port.

While a wide STP initiator port is waiting for a response to a connection request to an STP target port, it shall not reject an incoming connection request from that STP target port because of its outgoing connection request. It may reject incoming connection requests for other reasons (see 7.2.5.11).

If a wide STP initiator port receives an incoming connection request from an STP target port while it has a connection established with that STP target port, it shall reject the request with OPEN_REJECT (RETRY).

~~Wide~~A wide STP target port~~s~~ shall not request more than one connection at a time to a~~n~~ specific STP initiator port.

An expander device should not allow its STP ports (e.g., the STP target ports in STP/SATA bridges and any STP initiator ports in the expander device) to attempt to establish more connections to a specific destination port than the destination port width or the width of the narrowest physical link on the pathway to the destination port. This does not apply to connection requests being forwarded by the expander device.

An expander device should not allow its STP ports (e.g., the STP target ports in STP/SATA bridges and any STP initiator ports in the expander device) to attempt to establish more connections than the width of the narrowest common physical link on the pathways to the destination ports of those connections. This does not apply to connection requests being forwarded by the expander device.

Figure 3 shows an example of the simultaneous connection recommendations for an expander device containing STP ports.

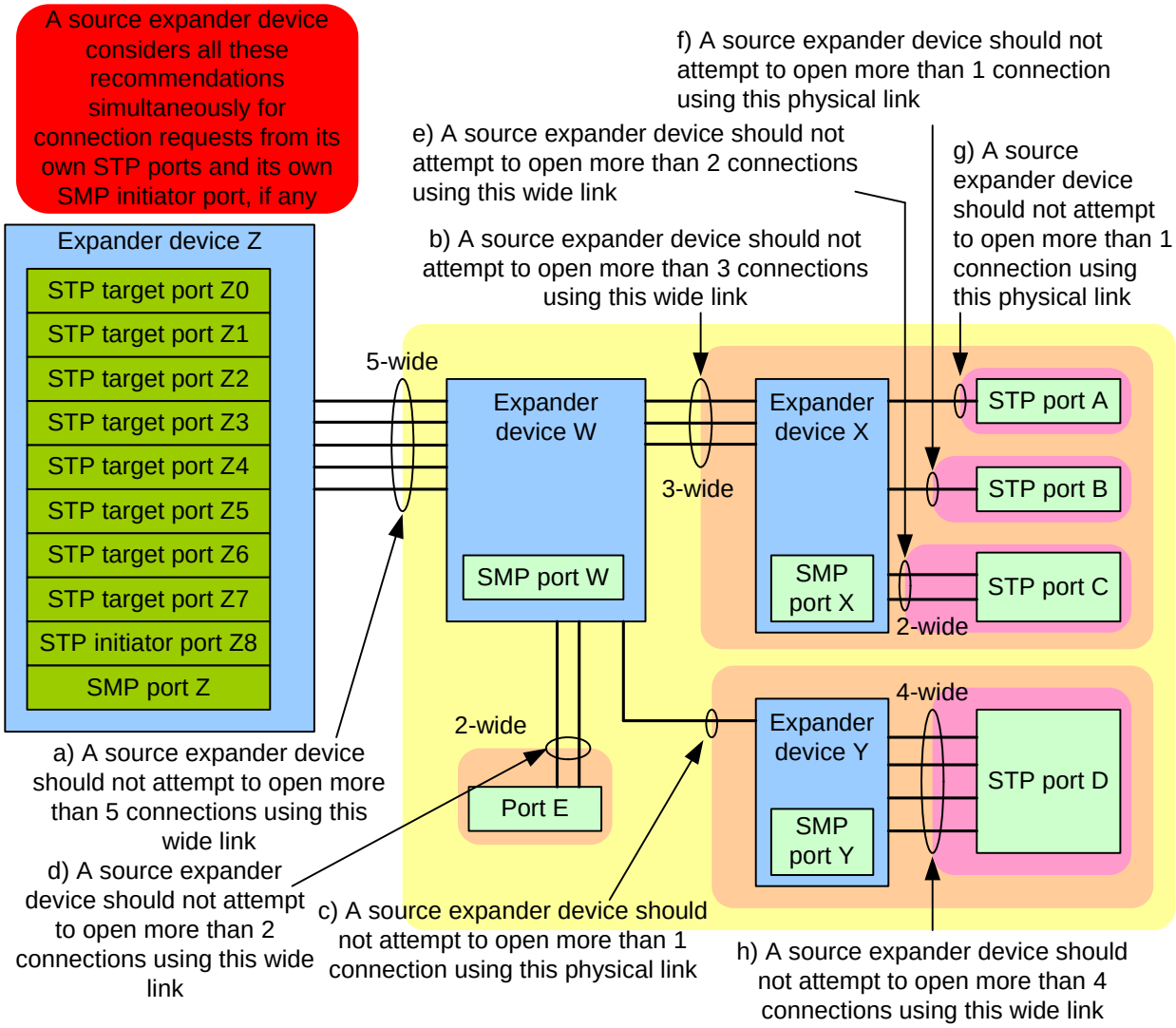

Figure 3 — Example simultaneous connection recommendations for an expander device

In figure 3, some of the recommendations are combined as follows:

a) Recommendations a), b), and e) together mean expander device Z should not attempt to open more than 2 connections to port C;
b) Recommendations a), b), e), f), and g) together mean that if expander device Z has 2 connections open to ports A, B, and X, it should not attempt to open more than 1 connection to port C. If it has 4 connections open to ports A, B, D, E, W, X, and Y, it should not attempt to open more than 1 connection to port C; and
c) Recommendations a), c), and h) together mean expander device Z should not attempt to open more than 1 connection to port D. If it has a connection open to port Y, it should not attempt to open another connection to port D until the first connection is closed.

While a wide STP target port is waiting for a response to a connection request or has established a connection to an STP initiator port, it shall:

a) reject incoming connection requests from that STP initiator port with OPEN_REJECT (RETRY); and
b) if affiliations are supported, reject incoming connection requests from other STP initiator ports with OPEN_REJECT (STP RESOURCES BUSY).

The first dword that an STP phy sends inside an STP connection after OPEN_ACCEPT that is not an ALIGN or NOTIFY shall be an STP primitive (e.g., SATA_SYNC).

7.18 SMP link layer

7.18.x Opening an SMP connection

An SMP target port shall not attempt to establish an SMP connection.