

## IEEE P1394.1 Bridge Standards Meeting

June 10-11, 1999  
Houston, Texas, USA

### AGENDA

**Chair:** Dick Scheel, [richard.scheel@am.sony.com](mailto:richard.scheel@am.sony.com)  
**Editor:** Peter Johansson, [pjohansson@aol.com](mailto:pjohansson@aol.com)  
**Secretary:** Subrata Banerjee, [sub@philabs.research.philips.com](mailto:sub@philabs.research.philips.com)

**Web:** <http://grouper.ieee.org/groups/1394/1>

1. Administrative
  - a. Minutes of previous meeting - April 29-30, 1998
  - b. Future meeting schedule
    - 7/22-23/99 Briarcliff, NY
    - 8/26-27/99 Portland, OR
  - c. Call for patents
2. Liaison reports
  - a. P1394a - Johansson
  - b. P1394b - Wooten
  - c. P1212r - James
3. Technical topics
  - a. Further discussion of Simple Routing Method (document BR50R01) - Frouin
  - b. Draft Text Submission, with sections broken out for separate acceptance (document BR047R05) - James
  - c. Review of SCAT (Document BR035R05)
  - d. <any other topics, if there is time>



After the discussion, a straw poll was taken to see if the working group wanted to continue pursuing the source routing method, or work only on the bit vector routing method. The result of the straw poll was:

- Pursue source routing: 2
- Work only on bit vector routing: 8

The P1394.1 working group will no longer consider the source routing method.

**B. Presentation Topic: Draft Text Submission, with sections broken out for separate acceptance**

**Presenter: Dave James / Sony**

Web document no. : [BR47R05](#)

Dave James presented several sections of the latest revision of his document. He asked for acceptance of each section, or review comments. See document number BR047R05

Section 1.8 (net refresh): Consensus of the group is that after merging subnets, all buses with duplicate bus\_ids will be renumbered (change 1.8.2.5 to renumber both buses). Retired numbers will be aged and reused. The same concept will be applied to virtual\_ids in the case of merging of buses. An idea was considered to allow two or more bus\_ids to refer to the same bus. We will think about this more later.

**June 11, 1999, Friday**

Before resuming the technical topics, Dave James moved and John Fuller seconded that the minutes of the April 1999 meeting be accepted as written. The motion was unanimously accepted. We then returned to Dave James' presentation.

Virtual node\_id reuse & quarantine: Use quarantine rather than reuse timeout.

Remote subaction payload: Bridges will be required to be able to forward 512 byte payloads, and will be permitted to handle larger payloads. If a bridge receives a packet that is larger than it is capable of forwarding, it will synthesize a response packet (response code TBD).

Remote subaction speed: The initiator will be responsible to choose the initial speed. Each bridge will be responsible to choose the speed to forward the packet at.

Isochronous: Dave's text stating the bridges need a 4 cycle buffer should be an implementer's note rather than being normative.

Request and response delays: "max hold time" is a more correct term. The working group needs to discuss (later) whether the time should be the same for request and response subactions.

Treatment of redundant paths: For any two buses A and B, the path for a request or response from A to B must be exactly the reverse of the path for a request or response from B to A. For any buses A, B, and C, the path from A to C is **not** required to be the concatenation of the paths from A to B and B to C. The standard will define the initial setup of bridges such that the routing will be a spanning tree. Hooks will be included such that a device can modify the resulting routing tables, without specifying or suggesting how the routing tables might be more optimally configured.

Section 1.4.1 (Async routing): Dave's text uses two bit entries in the routing table, with four states for each bus\_id. David Wooten asked for more text describing how the quarantine states are used, what the transitions are, etc.

Section 1.4.3 (clock reference routing): The working group agrees on the routing and the need for frequency lock. We have not received any response from the 1394 community about the need for the gross time to be the same on all buses in a network, or the need for cycles to be at the same phase.

Section 1.5.1 (packet sizes): This needs to be rephrased to indicate that 512 byte packet support is a minimum for bridges. When a too-big read response packet cannot be forwarded, then the bridge should forward a type\_error response packet with some additional information inserted in some field that is currently reserved. We will add some method to provide a node with information about a path to another node. This will be sort of a

"probe" that asks about max packet size, phy\_id of portal on initiator's bus that will route the packets, maybe number of hops, maybe speed that the initiator should send at, maybe optimized remote timeout. The method is TBD, but a bridge command using the extended tcode method discussed at the April meeting seems likely.

It was noted that even on an isolated single bus, a read request for a too-large payload should return a type\_error response. The responder should not ignore the request (causing a transaction timeout), and should not attempt to transmit the too-large packet. We should probably send a reminder to the 1394 community about this.

There was consensus that any request or response packet whose length field is too big (only exists in block read/write) will be turned into a response type\_error back to the requestor. One effect of this is that a block read request will be trapped on the way to the responder, rather than going all the way to the responder and then having the too-large response packet trapped along the path back to the requestor.

Section 1.5.2 (speeds): This section is mostly OK, but needs to adjust the first sentence from "speed of the slowest local-bus portal" to "speed of the slowest path to any local-bus portal". We should clarify that a bridge aware node may see a remote response come back at a different speed than the associated remote request was sent.

Section 1.5.3 (timeouts): Dave James proposes that some links (such as wireless) may have different forwarding times. Maybe we can use the "probe" idea above, and such links may enter values into the probe that reflect the "effective hop count" (or forwarding time). Note that nodes will still need to know what remote transaction timeout to use for the probe. Also, during network setup a network wide maximum remote timeout must be calculated and distributed.

Section 7.4.1 (GASP packet formats): The group decided that go fast / go slow packets will have the following header field values:

- t\_code = A (isoch / GASP)
- data length = 0 (unless we later include data to allow gross time setting)
- tag = 3 (GASP)
- channel = 31
- sy = 2/3 (go fast / go slow)

Note that P1394a requires non-bridge-aware nodes to put 0 in the sy field of GASP packets.

In section 7.4.1, table 9, change the sy values to:

- 0 - local
- 1 - global
- 2 - go fast
- 3 - go slow
- 4 .. F - reserved

The group discussed how to do gross time setting without conflict with Ipv4 use of GASP. One method is to follow the GASP specification and have two quadlets of RAC\_ID / SPEC\_ID before the time field. Another method would be to add language in P1394a "if tag = 3 and data length < 8 then the packet format is reserved for future standardization". The group did not decide whether to include a gross time setting capability, but this would at least leave a "hook" in P1394a in case we need it.

Section 1.6.2 (bridgebound virtual address translations): A question was raised. Should a node only clear the quarantine on the alpha portal, and use Dave's method of temporarily holding the transaction, check the alpha portal, and then forward the packet? Or should the alpha portal update all portals on the bus? Or should individual nodes clear quarantine bits in all portals of the bus?

Section 1.6.4 (local address translations): This needs to clarify that it addresses local bus transactions addressed by virtual node\_id. Perhaps change the title to "intercept request". If the requestor is not bridge-aware, then do as this section describes. If the requestor is bridge-aware, then return an error response with the phy\_id included in the response. This would require bridge-aware devices to use the more efficient physical addressing.

Section 1.6.5 (virtual address exceptions): This deals with the case of a request that uses a virtual node\_id addressed to the local bus\_id but using a nonexistent virtual phy\_id. It was noted that an ack\_missing on a terminal bus of a remote transaction may generate a response\_ack\_missing (the working group has not decided

this yet). Another possible behavior if a terminal bridge sees a nonexistent virtual node\_id is to generate a response\_address\_error with elaboration details included.

Section 1.7.1 (quarantine states): Dave needs to provide full documentation of the states and transitions of his proposal. This needs to include how a bridge-aware controller assists non-bridge-aware devices. It was suggested that while this section is being revised, that there be separate sections for bridge-aware nodes and non-bridge-aware nodes. Currently the group intends that non-bridge-aware devices be allowed to send one posted write request, and then they have to wait for the response (which is discarded) before they can send another request. An idea was offered for consideration: Since the legacy node would not see the real response (with the t\_label of the original request), maybe bridges don't need to block new requests.

The group discussed how portals know if nodes are bridge-aware. Some choices are:

- read config rom
- node notifies portals

This will definitely **not** be done in self\_id, since we don't want to require new phy chips.

We need to find out from the AV/C people whether there will be delayed AV/C responses. We think we remember that Hisato Shima said that there will not be, but some attendees may have heard that there will be. For example, the controller sends a "play" request to a VCR, followed by a "status" request. two hours later a status of "end of tape" is returned. Dave James was asked to confirm whether this may happen.

Some ideas were proposed in the discussion of Dave's section 1.7.1:

- Consider the "post" state as being a one-shot. A returning write response transitions back to "some" state.
- Maybe any write request to a legacy device causes "some" to transition to "post".
- When a legacy device transitions from "post" to "some", maybe the portal should force a bus reset, if the transition was due to a net event.
- Any local bus reset should transition all "post" to "some".

Dave James was requested to ask the AV/C people if AV/C can work with these ideas.

The group discussed the state flow for bridge-aware devices, and came up with the following:

- A new device on the bus starts in "some" (and the portals need to determine if the device is bridge-aware).
- Minor network changes (such as recycling addresses) cause "norm" → "some".
- Bridge-aware nodes will set "some" to "norm". We have several ideas about how this is communicated from the device to the portals on the bus.

One way to synchronize the quarantine states on all portals of a bus would be for the bridge-aware node to send a command to the alpha portal to change from "some" to "norm". The alpha portal would update all the other portals on the bus before returning the command response to the node.

Another possibility is to not synchronize the quarantine states among the portals on a bus. if a bridge-aware node sends a request and the portal handling the packet has the node marked "some" then return a response that has the phy\_id of that portal included. The node then sends a command to that one portal to change "some" to "norm". The other portals stay in "some" state. If future transactions go through other portals, then the node will handle the quarantine in each of them separately. We need to decide which response code should indicate running into a quarantine. Tentatively it will be response\_address\_error, but we should think about this some more.

We should also look into whether there is some clever way to have the quarantine logic not need to know if a node is bridge-aware.

**Handouts distributed at the meeting:**

- (1) Agenda
- (2) Minutes of the April 1999 meeting
- (3) Further discussion of Simple Routing Method Document [BR050R01](#)
- (4) Draft text submission Document [BR047R05](#)
- (5) IEEE 1394.1 SCAT list Document [BR035R05](#)

**Attendees of June 10-11, 1999 IEEE P1394.1 Bridge Working Group Meeting**

First Name	Last Name	Company	Phone	Email
1. Jean-Paul	Accarie	Canon CRF	+33-2-99-87-6864	accarie@crf.canon.fr
2. Chris	Dorsey	ST	+1-972-466-7850	christopher.dorsey@st.com
3. Firooz	Farhoomand	Matsushita	+1-408-653-4059	firoozf@ix.netcom.com
4. Laurent	Frouin	Canon CRF	+33-29-987-6870	frouin@crf.canon.fr
5. John	Fuller	Microsoft	+1-425-703-3863	jfuller@microsoft.com
6. David	James	Sony	+1-408-955-6295	dvj@alum.mit.edu
7. Peter	Johansson	Congruent Software	+1-510-527-3926	pjohansson@aol.com
8. Ats	Nakamura	Canon	+81-44-739-6634	atsnaka@bsd.canon.co.jp
9. Keiji	Nishimaki	HITS Laboratories	+81-462-77-9460	nisimaki@hits.co.jp
10. Yoshikatsu	Niwa	Sony	+81-3-5448-4603	niwa@sm.sony.co.jp
11. Tomoki	Saito	NEC	+81-44-856-2082	saito@ccm.cl.nec.co.jp
12. Dick	Scheel	Sony	+1-408-982-5834	Richard.Scheel@am.sony.com
13. Satoru	Toguchi	HITS Laboratories	+81-462-77-9460	wing@hits.co.jp
14. David	Wooten	Compaq	+1-281-518-7231	David.Wooten@compaq.com