# IEEE P1394.1 Bridge Standards Meeting

**March 25-26, 1999**
**Embassy Suites Hotel, Tempe, Arizona, USA**

## AGENDA

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

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

1. **Administrative:**
   1.1. Minutes of previous meetings (vote to approve Friday morning)
      1.1.1. December 10-11, 1998
      1.1.2. February 11-12, 1999

   1.2. Future Meeting schedules
      1.2.1. 4/26-27/99 Monday-Tuesday San Jose, California 2 full days
      1.2.2. June 1999 <we need a host>
      Possibly in Seattle (Microsoft), Philips-Briarcliff also made an invitation.
      Six weeks between consecutive meetings seems to work well.
      Next tentative meeting date: 2nd week of June 1999.

   **1.3.** Call for patents

2. **Liaison reports**
   2.1. Status of P1394a – Johansson
   2.2. Status of 1212r - James

3. **Technical topics (see handouts)**
   3.1. Simple Routing Method application to P1394.1 (document BR050R00) – Frouin
   3.2. Draft Text Submission (document BR047R03) – James
   3.3. Bus ID Enumeration in a Net of Serial Buses (document BR051R00) – Johansson
   3.4. Review of SCAT (document BR035R04)
   3.5. <any other topics, if there is time>

## Minutes of the March 25-26, 1999 Meeting

# Administrative Topics

**1.1** Minutes of the December 1998 and February 1999 meetings were approved unanimously on 3/26/99.

**1.2** P1394.1 meeting in April will be held in Sheraton Hotel, San Jose on April 26-27. Please let the host know if you will be attending (via email at Richard.Scheel@am.sony.com).

**1.3.** Call for patents. – None were identified. Some individuals said that only their company lawyers can accurately answer this, and only when a near-final draft is available.

**1.4.** Discussion on making faster progress on our standardization work

- We need to distribute the workload amongst us to make quick progress. For this we need a common platform so that people can take pieces of the problem and submit solutions in a standard format.
- For a common format C code or state machines were suggested. However, C code may not be helpful for a majority of the people.
- We may also need to collect the "corner ideas" at one place for convenient access. They can be send to Dick Scheel. There are corner cases that don't crash the network but may require extensive reset instead of local reset. If these corner cases occur infrequently then we can just ignore them for standardization. They, however, may be addressed in specific implementations. *E.g.,* two people trying to plug-in *networks* in a home at the *same* time. We don't need to go to undue complexity for handling such "corner" cases.

# Technical Topics

## March 25, 1999, Thursday: Morning Session

**A. Presentation Topic: Simple Routing Method application to P1394.1**
**Presenter: *Philippe Boucachard* and *Laurent Frouin,* Canon**
Web document no. : BR050R00

- This presentation described a source-routing based routing approach for inter-bus routing in bridged 1394 network. Primary objectives of this approach were to reduce memory requirements over bit-map based routing tables and to reduce complexity over previously proposed path dependent addressing.
- The presentation provided a summary of the remaining issues in this area – including bus number allocation, address assignment, management of routing table, removing a bus, and joining a bus.
- Dave Wooten's presentation in December meeting on path-dependent addressing was briefly reviewed.
- This proposal is in-line with path-dependent addressing proposal, and it attempts to address the issues of: memory consumption, routing complexity, ease of joining buses, congestion avoidance, and relation to virtual Ids.
- This approach needs support of an ARP like mechanism. Note that, ARP like service will be required with any routing scheme.
- This proposal is based on a source-routing approach that employs the BusID field for source routing header. Size of this field limits the number of buses/bridges/bridge-hops that this approach can support.
- Brief outline of this approach: (For a detailed description of this approach see Slide 5 in BR050R00.)
  - Uses BusID field as the source routing header.
  - The source routing header indexes to an offset that points to a path descriptor in bridge memory. Note that, this offset is independent of the 1394 offsets

- Limitations due to size of source routing header field:
  - 3 bridges per bus 9 bridge hops max 62 buses
  - 7 bridges per bus 5 bridge hops max 86 buses
  - Other variations also possible
- Details: Slides 7-13
  - Configuration:
    - Self numbering of routing labels based on alpha-portal mechanism
    - 2-5-bit routing label width. This limits the number if portals per bus (not per bridge!)
    - Routing cache table can be initialized through ARP like mechanism
  - Packet routing:
    - Source bridge: Looks at 3 lsbs of destination bus ID and then it fills up source routing information
    - Intermediate bridge: Fills up the reverse-path routing information
  - Maintenance
    - Joining buses: Local processing only!
      Few items to be standardized, most part will be implementation dependent.
  - Congestion:
    - Proposes limiting number of bridges per bus as way of congestion avoidance
    - Bus Ids are also virtualized
    - Assumes that more than 50 buses are not useful in most practical situations
    - Number of portals could be increased beyond 2 without changing the routing scheme

**Discussion:**

- The committee expressed reservation on the limitations imposed on number of hops, number of buses and number of bridges that this approach can support. It appears that due to the number of hops limitation situations with disconnected networks may arise.
- If this approach is compared with bit-mapped routing supporting a limited number of 64 buses then the proposed approach does not save much in routing table size.
- It was suggested that bit mapped routing table may not be all that difficult after all. Also ARP implementation with this routing scheme may get complicated.
- Consider Buses *A, B*, and *C*. Can a node on Bus *A* tell a node on Bus *B* how to talk to a node on Bus *C*? No, not if source routing is used. We have to pass EUI-64s instead. Should such "matchmaker" operation be a requirement for bridges? We don't know if this can be done at all. So we can't say *now* if this can be a requirement. It, however, has significant value for supporting legacy devices.

**B. Presentation Topic: Draft Text Submission to p1394.1 Committee**
**Presenter: *David James, Sony***
Web Document No.: BR047R03

- **Net Initialization (Section 5 of BR047R03)**
  - Net Resets
    - This part of the presentation describes propagation of resets in a 1394 network and the associated bus enumeration operations. Two types of resets are proposed: Hard and Soft.
    - Hard resets: Robust but disruptive
    - Soft Reset: less robust but non disruptive
    - We may not need separate hard and soft resets
    - Net reset packets are distributed in the network along a previously constructed spanning tree
    - Portal ID is defined with global scope. There can be as many as 1023*63 portal IDs. Note that, Portal ID is not related to PHYID or VID
    - Hard reset follows the following steps:
      - Local bus conquest: reset Conquest message is circulated amongst the local bridge portals. The node with highest resetID conquers the bus. Note that, a higher priority portal can take over the enumeration process from a lower priority one.

- When the *resetConquest* packet comes back to the prime portal, it delegates responsibility of propagating the reset to the one of its neighboring bus. This process continues until all the buses in the network are sent the resetConquest message. Note that during this phase bus numbers are also assigned.
- For hard resets, bridge portals discard all received packets and clear their AYSNC_ENABLE registers while observing resetConquest indications.
- Then *resetEnding* indication is propagated along the previously established path. Upon receiving this message hard reset recovery phase is entered. Normal remote transactions are allowed once the resetConquest indications are removed.

**Discussions**

- Further details can be provided in the next meeting.
- Working on a non-disruptive approach was suggested, and will be done. That is, resets should not disrupt configurations that they don't need to.
- It was suggested to use a two-step process before committing the bus numbers. That is, delay committing bus numbers until a later phase so that bus numbers are not changed needlessly. The presenter agreed.
- If bus addresses are not assigned during the first phase then portal enumeration can be carried on in parallel instead of sequentially as originally proposed. Sequential approach may still be safer because it is deterministic. Parallel approach may have race conditions.
- In order to control reset priority of the portals, couple of software settable bits was proposed, to prepend the EUI-64. For example, Node designed to a more recent specification revision might have higher precedence.

## **March 25, 1999,** Thursday**: Afternoon Session**

**B. Presentation Topic: Continued…**
**Presenter: *Dave James, Sony***

- Discussion on how to find out that a remote request is being sent to a non-existing bus and how to detect and respond to such requests?
  - If a node sends a remote request to a bus that no longer exists then the sender will have to wait for remote transaction timeout before learning that the destination bus may not be available. If an intermediate bridge portal knows that the remote bus does not exists then it can send a "NACK" type to message to the sender so that it does not have to wait unnecessarily for remote timeout period. A directory service that tells if a bus is not available in the network is another possibility but it may get lot of hits.
  - Outbound map of both portals will tell us which buses exist in the network. Bridge hardware will need a map so that it can decide which packets are to be picked up. This map may be derived from the two outbound portal maps. A portal may look at the map that tells whether a packet should be accepted or not. Then, at a slower rate, the map on the other side of the bridge can be looked up to see if the packet's destination node exists. To reduce delay, at the cost of extra hardware, each portal may cache the outbound map of its peer portal.
  - Different possible ways of sending a "bus doesn't exist" message need to be examined.
  - Bridge standard will specify 'a' way to set up the bit-mapped routing table. Other more intelligent routing table assignment techniques may also be allowed. Example: Fig. 4: Deadlock free routing with loops. Here not every bridge knows every bus in the network, however that information is collectively available from *all* the bridges on a bus.
  - During the configuration process we can leave the information on which buses exist in the network at all the bridge portals.
  - The committee agreed that "*Information on which buses exist in the network should be available locally in each bus."* Every time someone needs this information should not have to go to the prime portal.

- **EVENTS Messages (Page 51)**
  - In a 1394 bus, bus reset *event* was generated in response to a number of different events. In a bigger network of bridged 1394 buses, we may need different types of events to be communicated rather than just a single bus reset event.
  - Proposed format of such an event message is shown in Fig. 35. Event messages may be delivered via broadcast mechanism (with null target EUI) or sent to a specific node.
  - It was noted that certain event notifications need acknowledgements and hence they cannot be reliably sent via broadcast mechanism.
  - Note that, network initializations phases do not use this event mechanism

- **Redundant path topologies (Page 14)**
  - Redundant path topologies allow cyclical, but non-associative and deadlock free topologies of bridged 1394 buses.
  - Intelligent and optimized routing algorithms may be out of our scope. However, we should keep hooks in place so that the standard can support other routing implementations. Additionally, we may propose such routing schemes in an informative annex. Note that identifying optimum network topology may be part of the standard.
  - A concern was raised that if there is an existing network topology and an optimized network topology is known, then how does one transition from the existing topology to the optimum one? This issue appears to be somewhat involved and may be a separate group can look into this.
  - Cost to implement various alternate approaches should be considered as well.

- **Asynchronous Capable Node (Page 31)**
  - What is an asynchronous capable node? Dick Scheel gave a presentation on issues that end nodes need to be aware of in the 1998 Developers Conference.
  - It may be difficult to design bridge-aware end nodes based on ones anticipation of how the 1394.1 standard will evolve. We need to make sure that current operation on a local bus will not get in the way of bridge operation.
  - Maybe we need a new SCAT item on a list of ongoing activities in this group that may impact other relevant groups. We may also post a separate document on the web site so that other groups can look at them and sort of 'predetermine' how the bridge specifications may affect them.

- **Clock Reference Routing (Page 19)**
  - Clock Master: sends go slow/go fast to cycle master on local bus. There is also a Net Cycle Master and a Net Clock Master.
  - A proposal was made to have time (absolute value) locked clocks in the network instead of just frequency locked.
  - Note that, adjustment of time stamps in certain isochronous packets crossing a bridge has to be performed anyway.
  - Sometime ago, Mike Teener commented on why we do/don't need to have same time on all the buses. Dick will send email to Mike with a copy to reflector to get his opinion.
  - Do we need an alternate asynchronous routing for clock distribution? However the benefits of having a different routing tree for clock distribution may not be worth the effort. Different arbitration protocol for clock distribution may help.

- **Asynchronous Stream Routing**
  - One or more channels may be set as default broadcast channels. But who will set it up?
  - For now, let us assume that there is ***a*** way to do broadcast routing.
  - The clock synchronization information will follow the same mechanism used for broadcast messages.
  - Note that, bridges need to remember if they are part of acyclic tree via which the clock reference values are broadcasted.
  - Note that, if the clock master is also the cycle master then there will not be any go fast/go slow messages in that bus. However, If the clock master is different from the cycle master then the clock master can be on any bus.

- Another approach is to send go-fast/go-slow anyway independent of whether the clock master and cycle master is the same node or not.
- Davis James commented that synchronization of absolute clock values might be possible without significant additional expenses may be possible. It was also proposed to have software settable bits based on which a preferred clock master source can be selected.
- David Wooten gave arguments why we need not have a preferred cycle master and why we need not bother with synchronization of clock value.
- The following items were tentatively agreed to.:
  - There will be no go-fast/go-slow messages on the prime bus, *i.e.,* the bus with the prime portal.
  - Cycle master of the prime portal's bus is the cycle master for the network.
  - Alpha portals will be capable of becoming frequency master.
  - A node attempting to become the root node should have a go fast / go slow adjustable clock.
- A proposal was made to have same *cycle* number throughout the network
  - It was noted enforcing same cycle number throughout the network might require abrupt change of cycle number when two networks are joined. This might have bad unintended consequences.
  - However, in such cases the new root can watch the bus for sometime before exercising cycle master role to have a smoother transition.
  - This issue was tabled until next meeting.

## **March 26, 1999,** Friday**: Morning Session**

### C. **Presentation Topic:** **BusID Enumeration in a Net of Bridged Serial Buses**

**Presenter:** ***Peter Johansson***

Web Document no.: BR051R00

- This presentation describes how the bus numbers may be re-enumerated and re-assigned when two independent 1394 networks are connected together. Note that when two networks are connected together, duplicate bus numbers need to be resolved and a single prime portal has to be chosen to replace the two existing ones. These reconfiguration operations should be done with minimal disruption of the networks involved.
- Note that the *Alpha Portal* of a bus is the portal on the tree-based path to the Prime Portal. There is one Alpha Portal per bus in stable situations. However, during a reconfiguration there may be more than one Alpha Portal per bus.
- A field in each bridge portal on a bus will have PHYID of the *Alpha* Portal of that bus. Each bridge portal will also have a field containing EUI64 of the *Prime* Portal.
- There are two ways of joining two nets: (a) via a 1394 cable (merging two buses), or (b) via a bridge portal (adding a bridge to a bus)
- A bridge portal upon seeing a bus reset suspends sending packets *to* that just reset-ed bus. It stores those remote subaction packets in a (frozen) queue. However, packets *from* that bus are allowed as before.
- Dominant portals (alpha or prime portals) will be in touch with each other to find out which one is the survivor network. Note that bus ID assignments in the survivor network will remain unchanged while those in the victim network that conflict with those in the survivor network will be reassigned. Survivor network is chosen based on bus population. Tiebreaker is based on EUI64s. We may also use other parameters to choose the surviving net, such as number of nodes involved in remote transactions. Suggestion to have a mechanism to choose the surviving network based on software settable bit fields was made. Details of this step are being worked out.
- One of the goals in this reconfiguration process is to get things going for the surviving network as quickly as possible.
- A subordinate portal just needs to wait for a dominant portal to get in touch with it.
- Relevant information after reconfiguration has to be communicated to remote transaction capable nodes as well.
- Portals of the victim clan will get notification that they are in the victim net. These portals then suspend all cross bus transactions. When the victim clan portals get their updated VNID mappings then cross bus transactions are resumed.

- When do bridge portals of the reset bus know that it is OK to resume transmission to the reset bus? Each portal observes SelfID packets and determines if there are any new portals on bus. This is done by counting portals and also by looking at the bus topology from SelfIDs. If no new portal is found then the bridge portal can resume forwarding *to* the reset bus. Another approach is to have Alpha portal on each bus to look around to see if anything has changed after every bus reset. If nothing has changed then it will send survivor notification to all the bridge portals.
- Note that there may be multiple dominant portals when a 1394 cable joins two or more clans. How to pick one dominant portal? Details of this case are to be worked out.
- Alpha portal on each bus after every reset looks around to see if anything has changed. If nothing has changed then it sends survivor notification.
- Reasonable effort will be made for maintaining the same bus ids in victim net as much as possible. However, if there are too many duplicate bus ids in the victim clan then we may as well re-enumerate the victim clan.
- Duplicate addresses in the victim clan are quarantined for local split-transaction timeout.
- For details on actions taken by a portal in victim network see Page 7 of BR051R00.
- After the duplicate bus numbers are removed, we now have, in the former victim clan, pockets of buses without any bus number assignment. Now, the dominant portal in each of such buses sends a bus id request to the prime portal and gets a bus id assignment.
- For further details on this proposal see BR051R00 available on the 1394.1 web site.

## D. Discussion on SCAT List:

Updated SCAT list is available on the P1394.1 web site as document no. BR035R04. A brief summary of these updates is given below.

(a) Item 3 changed status from Hold to Closed
(b) Item 5: in Draft 04. Closed
(c) Item 6: written in Dave James' document. Needs more review
(d) Item 8: not connected to item 5 any more. Changed status from Wait to Open
(e) Item 10: bridges will be 1394**a** compliant. Closed
(f) Item 32: decision made by group. Closed

Dick Scheel will update the list, and will also improve the readability by using a small number of consistent terms for status, and sort the list by status. The updated document will be posted on the web site as document BR035R05. It was also suggested that a link to the current edition of the SCAT list be be added to the home page of the web site.

Meeting was adjourned at around 2:00pm (?).

### Handouts distributed at the meeting:

| | | |
|---|---|---|
| (1) | Agenda | |
| (2) | Minutes of Dec 1998 meeting | See 1394.1 web site |
| (3) | Minutes of Feb. 1999 meeting | See 1394.1 web site |
| (4) | Simple Routing Method: Application to P1394.1 | Document BR050R00 (hardcopy and floppy disk) |
| (5) | Draft text submission | Document BR047R03 (via floppy disc) |
| (6) | BusID enumeration in a Net of Bridged Serial Buses | Document BR051R00 (via floppy disc) |
| (7) | IEEE 1394.1 SCAT | Document BR035R04 |
| (8) | Information on April 1999 meeting in San Jose | See 1394.1 or 1394 TA web site |

## Attendees of March 26-26, 1999 IEEE P1394.1 Bridge Working Group Meeting

| First Name | Last Name | Company | Phone | Email |
|---|---|---|---|---|
| 1. Subrata | Banerjee | Philips Research | +1-914-945-6129 | sub@philabs.research.philips.co |
| 2. Philippe | Boucachard | Canon CRF | +33-2-99-87-68-73 | boucachard@crf.canon.fr |

| First Name | Last Name | Company | Phone | Email |
|---|---|---|---|---|
| 3. Tapas | Datta | Wipro | +91-80-557-8227 | tapas@wipro.com |
| 4. Firooz | Farhoomand | Matsushita | +1-408-653-4059 | firoozf@ix.netcom.com |
| 5. Laurent | Frouin | Canon CRF | +33-29-987-6870 | frouin@crf.canon.fr |
| 6. John | Fuller | Microsoft | +1-425-703-3863 | jfuller@microsoft.com |
| 7. Jerry | Hauck | Zayante | +1-510-668-1006 | jhauck@zayante.com |
| 8. David | James | Sony | +1-408-955-6295 | dvj@alum.mit.edu |
| 9. Peter | Johansson | Congruent Software | +1-510-527-3926 | pjohansson@aol.com |
| 10. Ats | Nakamura | Canon | +81-44-739-6634 | atsnaka@bsd.canon.co.jp |
| 11. Yoshikatsu | Niwa | Sony | +81-3-5448-4603 | niwa@sm.sony.co.jp |
| 12. Tomoki | Saito | NEC | +81-44-856-2082 | saito@ccm.cl.nec.co.jp |
| 13. Dick | Scheel | Sony | +1-408-982-5834 | Richard.Scheel@am.sony.com |
| 14. David | Smith | Texas Instruments | +1-972-480-6345 | desmith@ti.com |
| 15. Carlton | Sparrell | Interval Research | +1-650-842-6115 | carltonj@interval.com |
| 16. Colin | Whitby-Strevens | Zayante | +1-831-461-4948 | colin@zayante.com |
| 17. David | Wooten | Compaq | +1-281-518-7231 | David.Wooten@compaq.com |
| 18. Frank | Zhao | Panasonic | +1-408-653-4077 | zhaof@panasonic.com |