

**Minutes
IEEE P1394.1 Working Group
September 8-9 1998
Chicago**

Agenda:

1. Administrative
 - a. Minutes of August 20-21, 1998 meeting
 - b. Future meeting schedule

10/15-16	Thu-Fri	Maui, HI.	1.5 days
12/10-11	Thu-Fri	Monterey, CA	2 full days
 - c. Review of SCAT
2. Technical topics
 - a. Virtual Node Ids
 - b. <any other topics, if there is time>

Record of the meeting

1. Administrative

Dick Scheel thanked Max Bassler & Beth Harrison of Molex for their hospitality in hosting the meeting.

a. Minutes of August 20-21, 1998 meeting

Dick noted one correction: After the meeting was formally adjourned, Dick held an informal "task group" discussion about virtual node IDs. The notes of that session are working group document BR033R00.pdf.

Steve Bard moved that the minutes be accepted as amended, David Wooten seconded. The motion was accepted without opposition.

b. Future meeting schedule

10/15-16	Thu-Fri	Maui, HI.	1.5 days
12/10-11	Thu-Fri	Monterey, CA	2 full days
Feb 1-5		Newport beach	??

Since the working group web site is now fully functional, Dick suggested that the ftp site be discontinued. There was no objection. Dick will announce this intent on the reflector, and if no objection is expressed in a reasonable time the ftp site will be removed.

c. Review of SCAT

Dick presented a summary of the current SCAT list. This will be published on the web site as document BR035R00.pdf.

2. Technical topics

a. Virtual Node Ids

- 1) Consider how many protocols & apps will have problems with virtual node ids (SBP-2, IP/1394 ARP, etc.)
- 2) Note: the ARP response will contain the virtual node id in the async response packet header, but the physical node id in the ARP packet – packet header always has the correct address
- 3) Consider more generally what legacy nodes will be able to do in a bridged network, vs. nodes that are bridge-aware – such as handling of split transaction timeouts.
- 4) Note: We may need to have the assumption that virtually all legacy devices & apps will NOT work, and a lucky few may.
- 5) Peter Johansson said that he is convinced that physical node ids (with reset notification, etc.) is too cumbersome to work in a real system
- 6) There were some questions about how we got to the point where we felt we wanted virtual node ids (what problems are we trying to solve?). We as a group need a document on file that lists this, as well as any number of other topics that exist mostly in the minds of the working group members.
- 7) Neil asked why we can't just propagate bus resets across bridges. Some of the answers: too much reset traffic, too hard to route the resets.
- 8) Consider: Should we try to make a design that supports some set of legacy devices, or just move to a design goal that assumes all .1 aware devices?
- 9) Peter has a document in progress for virtual node id design. It will be file BR034R00.pdf on the web site when it is finished.
- 10) Each portal on the bus knows which portal is the alpha portal (highest phy_id)
- 11) Each portal on the bus uses the same algorithm for assigning virtual node ids with respect to the alpha portal, yielding the same map in each portal
- 12) How should we handle the case where the alpha portal is new, and therefore has no memory of the virtual node ids that were in use for the bus before the change? All virtual

- node ids have to be retired & new ids issued. Also, the new alpha portal doesn't know what old virtual node ids cannot be used (it wasn't there to see that they had been used), so could just quarantine all virtual node ids.
- 13) How do we handle the case where 2 buses are merged? The new alpha portal may not be either of the 2 previous alpha portals. Note: In this case the bigger disruption is the bus_id change.
 - 14) Note: When a non-portal device is plugged or unplugged, the alpha portal may change. However, the portals should only use this point of reference for calculating the virtual node id of the new device, and not change the virtual node ids of the old devices.
 - 15) In the case of merged buses, all portals who see that the new alpha portal is a portal that had been on their old bus can continue to use the same virtual node ids. If the new alpha came from the other old bus, then I need to invalidate all of my old virtual node ids (& learn from new alpha?)
 - 16) Dick polled the sense of the working group: When net topology changes, is it OK for it to take a moderate time to get things reconnected? (moderate = multiple seconds ... a minute or so). Note that a PC with an internal bridge plugging in/out would be a net topology change. The working group did not choose to say what its sense was
 - 17) Peter & David say that item 2 of the July virtual node id notes should be changed to indicate tracking only by topology, not EUI-64; since Peter's algorithm for assigning virtual node ids is topology based
 - 18) Peter says we may want to add some sort of a "refresh" as a way of keeping a virtual node id stored in a requestor active, even when it doesn't have a real transaction to do. Details to be supplied later.
 - 19) In item 4 from July in the "Virtual node ids discussion notes" (BR033r00.pdf): This should be changed to "t4 is the time that a virtual node id is kept unused after the reset that revealed that the device is gone" (approximate wording)
 - 20) !!!!! It was pointed out that we need to consider the effect not only of the time for actual reset/treeid/selfid, but also the time for the slowest possible bridge to digest the selfid info & do all of the follow-on processing
 - 21) Peter suggested that issuing of new virtual node ids only by the alpha portal (& pass to other portals) helps avoid some time problems
 - 22) BR033r00 July item 6 – Peter disagrees
 - 23) BR033r00 July item 7 – Peter's document would cover all of this.
 - 24) BR033r00 Aug notes: SBP2 example (2nd item) – Peter said that he is not sure that legacy SBP2 devices will work correctly
 - 25) BR033r00 Aug notes about 3ff src bus_id: Perhaps consider this an integral part of virtual node id ↔ physical node id mapping
 - a) We discussed whether individual (non-portal) nodes should have 3ff or real bus_id in their bus_id register
 - b) Also raised the issue of whether bridges would accept virtual node id-destined packets destined for same bus, & re-transmit after mapping onto the bus
 - 26) David said in P1394b they have discussed putting virtual node ids in end devices. Then they might need something (such as 3FE or 000) as a bus_id that means "this bus, with virtual phy_id".
 - 27) We discussed the idea of a well known virtual node id (bottom 6 bits). Maybe the alpha portal always has its own virtual node id + the well known virtual node id (zero). For just this one virtual node id, the actual assigned device may change at any instant (bus reset) – not quarantined. This needs more thought. Note that this limits the available virtual node ids to 62 per bus.
 - 28) We discussed the idea of various forms of wrappers:
 - a) carry time stamp / time of death?
 - b) carry src/dest EUI-64? (& possibly eliminate virtual node ids?)
 - (a) long discussion
 - (b) conclusion:
 - (i) lots more overhead (compared to virtual node ids)
 1. table space in bridge

2. packet size
 3. processing
- (ii) doesn't gain anything (compared to virtual node ids)

Side note: we are trying to find a solution that works OK with existing link & phy chips (for end devices), but we assume that firmware will have to change

- 29) Assumptions of what virtual node id info is stored in portals:
 - a) virtual node id
 - b) physical node id
 - c) EUI-64 (used only to respond to queries – from outside bus only???)
- 30) Do we want to add some optimization: When a bus reset shows that a new device has been connected, & its EUI-64 matches one for a virtual node id that is in quarantine period, then re-assign the same virtual node id?
- 31) !!! Note: the reserved bits in self-id packets (such as the one we want to use for identifying bridge portals) are set by the phy, not the link – so, portals will require new phys !!!

Notice was given that if anyone knows of any patents that are needed to implement this standard, they should announce it. Failing to do so may jeopardize their IP rights. (Dick will check with IEEE for correct wording.)

Day 2-----

b. <any other topics, if there is time>

Discussed go fast / go slow. Decided by consensus to use preallocated isoch channel method. Some details:

- The tag or sy header field will indicate go-fast or go-slow.
- All P1394.1-aware and P1394b nodes that are cycle master capable must be IRM capable and handle these adjustment commands.
- Node must indicate that it supports the commands in its config rom.
- P1394.1 specification will predefine a channel number.
- The cycle master/root must preallocate this channel & bandwidth before making the channels / bandwidth available registers available after reset.
- Should the command be "sticky"? (How long should the effect of the command be?)
 - one cycle
 - n cycles
 - until another command is received
- We must deal with the case of one / some / all buses in a network that are not frequency locked (briefly or long term), due to having a cycle master that does not support adjustment commands.

David and Peter started a list of what we need to know about nodes and when we need to know it.

	info	when needed
1.	node is a bridge portal	self id
2.	adjustable CM capable	cfg rom (bus info block preferred)
3.	aware of implications of portals in its environment (.1 aware) <ul style="list-style-type: none"> • split remote transaction timeout • don't need to serialize off-bus transactions • <whatever else – TBD> 	cfg rom (bus info block preferred)
4.	1394b	self id
5.	virtual node id support in node itself	when virtual node ids established ???

Even though the self-id data can be put into a 2nd self-id packet with no ports listed (on a typical 3 port phy), the possibility of using caboose self-id packet was mentioned. The caboose packet would provide more available bits for other stuff.

Time-of-death handling

The consensus of the group was:

- no wrapper
- each bridge has a limit for how long it may take to forward a packet
 - fixed?
 - configurable?
 - use split timeout value of bus the bridge accepted the packet from
 - use split timeout value of bus the bridge is placing the packet on

How is limit in bridge set?

How do requestors know what remote split timeout to use, based on this?

- Assume worst case 1023 linear buses
- assume almost worst case of total # buses in net are linear
- someone looks at net topology and provides better value
- some sort of ping-like process (doesn't check actual time – instead, accumulates from each bridge in path their forwarding timeout)

Although some cases (such as no-ack) require that the bridge not synthesize a response packet, the case of a bridge discarding a request due to too much time in the bridge does allow safely synthesizing a response packet indicating the discard.

It was noted that (due to split timeout / quarantine stuff) non-.1-aware devices can never do remote requests. For example, SBP-2 initiator & target would need to be .1-aware, but buffer could be non-.1-aware. It was noted that pretty much all real devices that are in the market (mostly AV/C devices) would be limited to local bus conversations, since both ends of an AV/C conversation need to be async requesters. Note: We could just specify that non-.1-aware devices will not have their requests forwarded off-bus by any bridge (result is no-ack, probably). We should widely publicize this as a decision (legacy devices can not request off-bus, and the bridges will block any attempts), & see what the reaction is. (This may result in temporary increase in attendance, to challenge this?) Dick asked if anyone could write a few paragraphs about how we derived this conclusion. There were no volunteers.

David proposed that the cap on bridge forwarding time be a constant, and that there be a response code indicating that a bridge discarded a request due to exceeding this constant. We discussed what that constant would be. If the constant was the worst case local split timeout (which we think is about 8 seconds), then the worst case of 1023 buses yields a remote split timeout of more than a day. People noted that at the human interface level users don't want to wait for more than a few seconds for positive/negative response to their input.

Attendees:

Name	Company	Phone	Email
1. Ats Nakamura	Canon	+81-44-739-6634	atsnaka@bsd.canon.co.jp
2. Brad Saunders	Rockwell	+1-949-221-6513	bradley.saunders@rss.rockwell.com
3. Daisuke Hiraoka	Sony	+81-3-5448-4603	dai@sm.sony.co.jp
4. Dave Eicher	Rockwell International	+1-319-295-8348	deeicher@collins.rockwell.com
5. Dave Zalatimo	Filanet	+1-650-968-9711	davidz@filanet.com
6. David Wooten	Compaq	+1-281-518-7231	David.Wooten@compaq.com
7. Dick Scheel	Sony	+1-408-982-5834	dicks@lsi.sel.sony.com
8. Frank Zhao	Panasonic	+1-408-653-4077	zhaof@panasonic.com
9. Hisaki Hiraiwa	Sony	+81-3-5448-5420	hira@wcs.sony.co.jp
10. Jerry Hauck	Macro Designs	+1-510-668-1006	jerry@macrodesigns.com
11. John Fuller	Microsoft	+1-425-703-3863	jfuller@microsoft.com
12. Laurent Frouin	Canon CRF	+33-29-987-6870	frouin@crf.canon.fr
13. Masa Akahane	Sony	+81-3-5448-5420	akahane@wcs.sony.co.jp
14. Masatoshi Ueno	Sony	+81-3-5448-5615	ueno@av.crl.sony.co.jp
15. Michael Smith	ControlNet	+1-408-341-1428	michael@controlnet.com
16. Neil Morrow	Texas Instruments	+1-972-480-6211	nmorrow@ti.com
17. Ozay Oktay	Canon	+1-949-856-7180	ozay_oktay@cissc.canon.com
18. Peter Johansson	Congruent Software	+1-510-531-5472	pjohansson@aol.com
19. Satoru Toguchi	HITS Laboratories	+81-462-77-9460	wing@hits.co.jp
20. Steve Bard	Intel	+1-503-264-2923	steve.bard@intel.com
21. Takashi Sato	Philips Research	+1-914-945-6099	txs@philabs.research.philips.com