IEEE P1394.1 Bridge Standards Meeting

July 22-23, 1999
Philips Research, Briarcliff Manor, New York, 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: (see handout)**
   1.1. Minutes of previous meetings

   1.2. Future meeting schedule
      1.2.1. August 26-27, 1999, Portland, Oregon, USA, Host: HP
      1.2.2. October 14-15, 1999, Rennes, France, Host: Canon
      1.2.3. November 4-5, 1999, San Jose, Host: Sony (tentative)
      1.2.4. December 2-3 or 9-10, 1999, Redmond, Washington, Host: Microsoft (tentative)

   1.3. Call for patents

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

3. **Technical topics (see handouts)**
   3.1. Efficient Bandwidth Reservation for Remote Isochronous Connections via 1394.1 Bridge(s) (document BR056R00) – Banerjee
   3.2. Draft Text Submission, Rev. 6 (document BR047R06) – James
   3.3. Review of SCAT (document BR035R05) –Scheel
   3.4. <any other topics, if there is time>

Minutes of the July 22-23, 1999 Meeting

Administrative Topics

1. General

1.1. June '99 meeting minutes were unanimously approved. Peter Johansson made the motion for approval and John Fuller seconded it.

1.2. There is a plan for a 2-day editorial session in the West Coast. From a show of hand it appears that 6-7 members will show up at the editorial session. Details will be available on the web/reflector.

1.3. Future Meeting schedule:

1.4. Call for patents. – None. Dick Scheel explained that it is not up to us to see if there is any relevant patent. We are not responsible in evaluating is a patent claim is relevant. Dick has a copy on this topic from IEEE Standards.

2. Liaison Reports

2.1. IEEE P1394a (Peter Johansson)

Draft 3.0 has been handed off to IEEE for re-circulation. After that there will be a 30-day ballot period. Hopefully the draft will be submitted to IEEE standards review committee this year and it may also get approved this year.

2.2. IEEE 1212r (David James)

Moving along. Draft document will go out for review before the next meeting. Hopefully the document will be approved for balloting in the next meeting. This will be the last chance to review and submit technical contribution.

2.3. IEEE1394b

Making rapid progress. Hopeing to close by October meeting. Few open issues are left. Editorial effort may need time beyond October. Resolves some of the ambiguity of "ack missing". A PHY forwards a NAK only when it sees NAK on all of its downstream nodes.

Technical Topics

July 22, 1999, Thursday: Morning Session

A. Presentation Topic: **Efficient Resource Reservation for Remote Isochronous Connections via IEEE1394.1 Bridges**

Presenter: ***Subrata Banerjee, Philips Research***

Web document no. : BR056R00

- Bandwidth reservation in 1394 networks may be inefficient for some applications. For example, is an application needs to send K-byte isochronous packets in 4 out of 5 isochronous cycle and a 2K-byte packet in every fifth cycle then an isochronous bandwidth of 2K byte/cycle must be reserved. This situation may arise in the case of MPEG video formatted according to IEC61883. (See the presentation for details.)
- An efficient method of isochronous resource reservation method was presented which requires additional parameters such as peak rate, burst duration etc. in order to reduce bandwidth wastage in 1394.1 bridge fabrics. Note that bandwidth reservation within a 1394 bus will still follow the 1394 specification.
- The group's response to this proposal was positive. Following suggestions was made
  - → Benchmark with respect to practical applications on how inefficient the current approach is.
  - → Compute the worst case memory requirement for the proposed approach
  - → If a common fixed delay for all isochronous channel is acceptable then detailed benchmarking is not required
- Additional details on this proposal are to be presented at a later meeting

B. Presentation Topic: Draft Text Submission
Presenter: ***David James, Sony***
Web Document No.: **BR047R06**

Isochronous Connection Management

- **Non-overlaid Connection (Page 79, Section 5.2.1.1)**
  - → For one-talker multiple-listener connections, one isochronous connection is used in the shared portions of the point-to-multipoint routing tree. A use count is maintained indicating the number of listeners currently sharing the connection. Use count > 1 indicated overlaid connection, otherwise the connection is non-overlaid
  - → Setup is proposed to be from talker to listener. However, connection set-up from listener to talker side appears to be similar.
  - → Discussion on who and how existing connections are re-established after a bus reset. Proxy controller on the bus may be reallocating the resources. The connection counter also needs to be updated.
  - → Connection counts are reset after one second. If there is a reset on any bus on the path – a device on that bus (*e.g.,* a proxy) reallocates the resources within 1 second. Listener – listens on the same channel for one second after reset, if nothing then stop listening – resources didn't get allocated
  - → This approach is for peer-to-peer connections. Multicast and broadcast connections need different treatment.
  - → Maximum number if listeners in a bus should be allowed to be as large as 63, instead of imposing an artificial limit on that number.

- **Partially Overlaid Connections (Section 5.2.3.1)**
  - → There is an advantage of setting up the connection from listener to talker for overlaid connection as then the use counts are easier to manage
  - → Each portal maintains stream-id registers.
  - → Note that multiple controllers may be trying to set-up connections simultaneously. Count reflects the number of listeners throughout the network. A connection is taken down when its connections count reaches zero. Connections are torn down from listeners' side by going along the upstream path and releasing resources whenever (count-1) reaches zero. This process stops when (count-1) > 0 at some point in the path.

- **Talker Disconnection - Section 5.2.4.**

- → Proposed procedure is not optimal
- → Error recovery
  - ✓ How does the talker know what resources are allocated? What if the controller got unplugged or talker got unplugged. Tear down is to be performed by someone when the talker disappears or any resource in the middle is gone
  - ✓ Confirmation when the whole path is torn down or separate confirmations as soon as discrete parts of the connections are torn down is/are required. Upon Bus reset confirmation message may get lost?
  - ✓ Only confirmation we may need is that the first portal got the teardown message and that it is taking care of the rest of the teardown procedure. Controller doesn't need a confirmation when the whole relevant path is torn down. Note that, Confirmation comes back from the first point where another listener's connection branches out. Confirmation tells the controller that the isochronous resources are successfully deallocated
  - ✓ Questions raised:
    - Q: What do the controller do if the confirmation is not received?
    - Q: Is the confirmation from the first portal that the teardown message is received sufficient?
      DVJ probably not.
    - Q: For set up, when is the confirmation to the controller sent that the connection is set-up?
    - If there is bus reset on the a remote bus who will reallocate the channel?
  - ✓ Step by step confirmation suggested by DW instead of waiting for the final confirmation. However, if we do it only once at the end then less messages to be sent
  - ✓ This issue needs further investigation.

❖ **Talker Bus Reset Recovery (Section 5.3.1.)**

- → Controller can change channel number after bus reset by rewriting PCR
  - ✓ Channel can be changed on the fly for Sony camcorders without stopping the stream.
  - ✓ Panasonic camcorders first needs to be stopped then channel can be changed and the device can be started again.
- → If the bridge is to be 61883 aware, then it might create more problems than what it solves.

❖ **Talker-bus net-refresh recovery (Section 5.4.1)**

- → Half-baked, not thoroughly thought out.
- → If a net is inconsistent then how do we make it consistent?
- → We are trying to make the legacy devices work. Note that, bus reset on a bus along the path may not be seen by the controller.
- → So one suggestion was that whenever there is a reset on any bus – send an asynchronous broadcast message to the entire network indicating the event (bus reset) and location (bus number)
- → Responsibility of a bridge talker will be to restore what the use count was before bus reset
- → Note that, use count in 61883 also reflect internal connections

❖ **Outgoing virtual address translation (Section 2(b))**

- → Nothing new here. As noted earlier, reading portions of the virtual id map is not efficient. The entire address map should be read.
- → After reset, portals need to handle inbound traffic which comes to the bridge portal. So, immediately after the reset *all* the bridge portals will read the mapping from the alpha portal.

❖ **Quarantine**

- → Needed:
  - ✓ To prevent legacy devices from going off bus
  - ✓ To allow time for the network to become stable after a reset before resuming remote transactions.
- → If a quarantined node initiates a remote transaction then its bridge portal returns a response saying, "your transaction is denied because you are quarantined."
- → Timeouts were devised to solve the problem of killing stale connections. Can this approach eliminate these timeouts? When virtual ids runs out, new bus number is to be used. When one is out of bus numbers then all the pending transactions should be terminated? Does this eliminates need for timeouts on validity of virtual ids.? This needs further investigation.
- → Let's assume that a bus number has been changed. Now transactions to the old bus number should cause an error code to be returned as response code. But who will return this error code?
- → After an old bus id. is renumbered, fellow portals are to be instructed to remove their routing entry for the old bus id. As the notification of change in bus number is propagated, all transactions to the old bus number are automatically discarded.
- → Alpha portal will pick up the transaction request to old bus number while the bus number is in transition mode. However, the Alpha portal may not be on the path of the remote incoming transaction.
- → Need to look at more detail if the proposed net reset approach would solve some of the outstanding issues.

July 22, 1999, Thursday**: Afternoon Session**

❖ **Packet Sizes (Page 21, Section 1.5.1)**

- → Bridges should be able to handle at least 512 byte packets. Bridges operating at higher speed might be able to handle larger packets
- → If a bridge cannot handle a large packet then can the bridges truncate the packet (*e.g.,* write response)? In that case the write response is to be converted to an error response.
- → Big transactions are discarded. Big subactions that are too big are also discarded but an error response is returned instead.

❖ **Packet Transfer Speed (Section 1.5.2)**

- → Bridges have to look at the whole path to find out the most suitable speed
- → Require bridges to have link speed capabilities that are at least as fast as their PHYs. Then speed capabilities of a node can be inferred from Self-ID packets.
- → What error codes are to be used for too big payload or too fast transmission? Proposal was to send a general "type" error with subtypes providing additional details.

❖ **Routing decision logic** (See Page 33 + Table 2)

- → Top part of the table checks destination and bottom part checks the source
- → Outbound = leaving a bus, inbound = entering a bus(not a bridge)
- → Routing rules are described in the draft. Group questioned if there was anything wrong with that or if this write-up is a different way of expressing the same thing. Presenter (DVJ) was not sure. May be the one in the draft is not as complete.
- → Recommends error count register in bridges.
- → Eight bits are used per physical id. Two bits for quarantine purposes and remaining six for translation purposes.

- ✓ Do bridge aware nodes need to get permission from the bridge portals to be involved in remote transactions? Note that quarantine makes sure that a wrong device is never accessed.
- ✓ Mappings at a quarantined node are thrown away. All portals should synchronize their quarantine information with the Alpha portal. This is inefficient since every portal needs to check twice with the Alpha portal. This approach works but may be a better way exists.
- ✓ Overhead to check quarantine information with Alpha portal will have some effect on remote timeouts. But this effect may be acceptable. This approach appears to have more overheads than distributing the new virtual id information to the bridge portals.
- ✓ A bridge aware device may need to be quarantined to prohibit it from addressing a non-existing bus.
- ✓ How to broadcast a message to a bus? Device discovery? ARP? Is there any other protocol using broadcast writes other than A/VC? Most likely not. We do have broadcast write via asynchronous streams. A new discovery mechanism that uses asynchronous streams is required.
- ✓ How to support broadcast write to a remote bus? PHYID 63 indicates broadcast on the destination bus. Broadcast messages to a remote bus may or may not require an acknowledgement It was informally agreed that broadcast to a remote bus will not be supported unless someone provides a good reason for it. VNID 63 will be converted to local id. of the alpha portal.
- ✓ Current host controllers will not accept 63 as a valid address.
- ✓ In Table 2 we need to add a column for destination Id (if destination id=63 then don't forward)

❖ Following two topics were briefly touched upon:

→ When two networks are disconnected and reconnected what properties/parameters might be maintained. (Fig. 35, Page 46)

→ Non-Disruptive Reversed Secondary Reconnection (See Section 4.4.4., Page 48)

❖ Discussion on how IEC61883 PCRs can be handled after various bus and net changes e.g., resets etc.

→ Discussed possible proxy controller – architecturally separate from bridges but may be packaged together

- ✓ Would have to handle all work after a local bus reset that would normally be done by a controller in a single bus environment. This may get standardized in 1394.1 or elsewhere. We may need to work on three parts of its functions: (a) Talker, (b) listener, and (c) routing – intermediate buses.
- ✓ We need to understand what controllers on single buses have to do. We need very unambiguous description of what 61883 says – Peter will give that to David James once he intends to work on this topic.

→ DJ showed some proposed message layouts

- ✓ A separate topic will be how to deliver these messages?
  - DJ's proposal with ext. Tcode
  - DJ proposes that portal on a bus to look at the messages to see if they need to deal with it
  - DW pointed out that with the ext.Tcode method, we may redefine the 48-bit offset field to whatever we want it to be, if that is useful.

→ If there is a topological change such that asynchronous routing changes (but isochronous routing could stay the same), should the isochronous connections be torn down and reconnected according to the new asynchronous routing? Several members leaned towards teardown and rebuilt, however they need more time to think about it.

→ DW: Are we staying with the idea of bus ids and virtual ids being replaced/retired on chabges, garbage collection after "safe" time or when we run out of ids we do a net reset.

- ✓ Do we allow > 1 bus id for a bus at a time? – not decided yet.
- ✓ When a bus runs out of virtual ids, recycle all virtual ids to get a new bus id
- ✓ Need to deal with case where:
  - Request comes in to a bus with vid to phyid mapping

- Destination node receives the request
- Virtual node id map changes
- Node sends response
- Now how do we map phy id to vid as the response goes off bus?

✓ Note that when any net refresh occurs, all bridges/buses can/should recycle their vids (free the dirty ones), since addresses are flushed in all devices in the net.

→ We found that we had different ideas about what we thought "net refresh" means. We discussed:

✓ Case when recycling bus ids (dirty -> free), so we have to flush all queues, all ARP cashes. This is net reset.

✓ Case when some bus ids have changed, but only free to in-use and in-use to dirty conversions too place. This will cause routing tables to change as well. This is net refresh.

✓ DJ also mentioned a third possibility when only the net topology – and hence routing tables – changes. This may be called as net restart.

→ Net reset would use quarantine and net refresh

✓ One possibility is for Net Reset to cause bus resets on all buses, instead of using quarantine

✓ This would include a generation number for each bus. Bridge aware nodes will check the generation number after every bus reset to determine if there was a net reset as well.

→ Back to the question on what to do when virtual id map changes between request reception and response transmission?

✓ Not sure. We may require bridges to remember old virtual id mapping and send responses with old vid. Here we are trying to avoid the dreaded remote-timeout.

✓ Another idea: send out notification of old bus id -> dirty. Then for a local split timeout treat the bus as having *no* bus id (not reachable, except allow responses to go out with old virtual address)

- Need to think more on this approach
- Note that if responder is legacy then the local bus reset will cause it to cancel the request.

❖ Section 1.11.3.1 – 1.11.3.4 in Draft Text Submission:

→ Maybe we should assign a new bus_id to the merged bus and then make the previous bus ids as dirty

→ Also change generation number so that all bridge aware nodes will know to cancel all outstanding responses and transmissions.

July 23, 1999, Friday: Morning Session

C. Discussion Topic: How to handle pending remote transactions when bus-id changes
Discussion Leader: *David Wooten, Compaq*

❖ Consider a request is sent and destination bus number changes before the response is sent by the destination node. Hence Response comes back as bus number doesn't exist

❖ How to take care of this situation? What type of notification is required? Change the bus number register at the local nodes?

❖ Is there any case when a bus number will change without a bus-reset close by. Yes, there can be reset on another bus that will change the bus number on this bus. When there a bus number has changed at a bus there can be a notification to all the bridge-aware nodes that cancels pending remote transactions at local nodes whether or not there is a bus reset.

❖ Bus Reset is such a notification that is seen by everyone. But bus number change notification is not required to be seen by non bridge-aware devices.

- How quickly the portal will know that there is a need for notification. Polling from bridge aware nodes?
- Note that, until addresses are not sorted out nodes are not allowed to generate anything that goes outside the remote bus. The nodes have to be notified to proceed.
- Should Alpha portal do individual confirmed writes to bridge-aware nodes? What happens if a node does not return response?
- Broadcast write that *guarantees* that everyone has seen it is difficult.
- Note, dot-1 compatible nodes must handle GASP. Broadcast async. write is preferred to GASP for lower overhead. Also, all nodes may not handle GASP. For device discovery, devices that do not support GASP may have a GASP-aware proxy. Indirect device discovery for such nodes is then possible.
- Peter proposed **three** types of notifications for events
  - → *Queues to be flushed* – needs ack confirmation from the recipient
  - → *Courtesy notification* – no error would occur if this is not heard by everyone, no ack confirmation
  - → Queues *not* to be flushed but need ack confirmation
- Is the additional overhead of reset as a means of notification large enough so that we need another class of notification?
- We don't have the new bus number information instantly after a bus reset. It takes time to figure out the new bus number that will be assigned. Don't allow anything to cross the bus during this period of uncertainty. Two portals may try to pick up a remote packet, which is wrong.
- If the portals are set up for blocking remote transactions then there will be resynchronization when attempting to re-enable remote transactions. Blocking should be done at the source nodes not at the portals.
- If someone puts something on the bus and gets ack-missing then something is wrong. There is no known mechanism to handle it. Ack pending is OK.
- Hardware behavior – write to a specific address will force specific events to happen at the destination node. But we are trying not to change link behavior as mush as possible.
- Notification: You are not allowed to send anything off-bus until it gets notification that it is OK. We can send Ack pending for something that got dropped (OHCI). After bus reset if you are bridge aware you must wait for a (broadcast) notification before you resume off-bus transactions.
- If a bridge-aware node doesn't hear anything after bus reset then it can query the bridge portals on the status.
- If there is no associated bus reset (bus number changed due to some event off bus) then a bus reset can be forced and same protocols when there is a bus reset can be used.
- Can't broadcast survivor notification because everyone on the bus is not a survivor.
- Side effect of using bus reset for such notifications. Others can use this for many different notifications. However, the scheme that uses bus reset works. However, other viable alternative should not be ruled out.

D. Presentation Topic: Draft text submission: Disconnection - Section 1.11.4

Presenter: *David V. James*, Sony

- Slides of test cases needed to validate new algorithm – Dick
- Section 1.11.5 – Node connection and disconnection effects
  - → Section 1.11.5.1. – Detached node
    - ✓ Difference between a node that is still there but its number has changes and a node that has been disconnected
    - ✓ What if a node gets unplugged during split timeout
    - ✓ Consider a request being issued from 1.2 to 0.2. Meanwhile 0.2 gets unplugged after request was received by it and before any response was generated. Who should handle this situation?
    - ✓ Big management responsibility for bridge aware node.

- ✓ Purpose of the notification is to accelerate the pinging of 0.2
- ✓ Indication that something has changed somewhere – go figure it out what happened. This might be too much burden.
- ✓ Notification caused by suspend
- ✓ There shouldn't be any stranded responses at suspended nodes
- ✓ Net refresh vs. net reset
  - Recycling bus addresses – net reset – flush user queues and address maps. Then refresh the translation maps. Queues to be dumped – not to be concerned about responses coming from old requests.
- ✓ Possibility of undiagnosable remote timeouts

❖ When every bus is renumbered then it is like a net-reset operation

→ At net-reset one can re-verify his address instead of throwing everything right away.

→ Net-reset on split-timeout will give us information that there is no address overlap and notification for potentially stranded request

→ DW looked at this issue before. Some things have changed since then. DW needs to go back and look at it again.

→ Pulling the nets apart and then back together – potential of having some stranded packets

E. Presentation Topic: SCAT List
Presenter: Dick Scheel

❖ Item 31: Resolved

❖ Item 38 added – forwarding time limit in bridges

❖ Item 30 closed - Settled in 1394a. Bit in second quadlet in Self-ID field will identify a node as bridge portal

❖ Item 7: reopened

❖ Item 4: All (blank)

❖ Item 16: Congestion – needs review of document BR047R00

→ If near lifetime limit then a bridge will send a synthesized response that the "bridge is overloaded" or "stuck at bridge"

→ Start sending error response as soon as it is known that the timeout will be violated. To do this bridges on the path needs to know the split timeout values for bridges/buses ahead in the path. If this value is same for all bridges/buses then it is easier to do.

→ Check going into the FIFO, may not go into the wire

→ If we don't have uniform remote timeout throughout the network then we have to be careful. Start with default value? Send ARP and wait for response. That should be the order of remote transaction timeout.

→ Fairness issue 400Mbps on one side, 100Mbps on other side

- ✓ Busy Retry, round-robinish type of service

→ Closed?

❖ Discussion on split timeout value (which is used in congestion item) – *should it be fixed for all buses or should it be variable ?*

→ May have agreed earlier in the committee that split timeout value cannot be changed once set?

→ Note: Max. split timeout in any bus can be 8sec. Remember that it is the maximum value. In most situations nodes will take much less than this maximum allowed value

- → Hook: If a portal sees a change in their local split-timeout value then it will notify others in the network. DS says that in practical situations this part of the firmware will be invoked rarely if at all. DW: On the other hand this may get invoked too often – we don't know.
- → JF: If the split-timeout value is different than a certain value then the bridge portal will set it back to this value. Is that allowed if the bridge portal is not the bus manager? PJ – Yes.

❖ Item: Busy/Retry – In addition to the table (Table 10 in BR047R06) additional text advising implementers on busy/retry. How long to wait to get on the bus. How long to keep trying busy retry – e.g., if there is congestion? Should I wait until the timeout or give up earlier. How often the transaction should be retried?

❖ PJ – may be bridges should not try *dual-phase* busy-retry.

F. Presentation Topic: Draft Text Submission: Request Processing

Presenter: ***David V. James, Sony***

Web. Document no.: BR047R06, Table 9, Page 76

❖ Row 1 – discard -> Normal, no action required
❖ No significant comments
❖ Table 10, subaction reception errors
- → Busy retry if the bridge is busy
- → Can some node make false assumption that busy_ack means that bridge queues are getting filled up – congested
- → Revised dual-phase retry as specified in IEEE1394a now works better than it used to in IEEE1394-1995
- → How busy retries are implemented? Some do out of order – if the transaction at the head of the queue gets busy retry then the next packet in the queue can be retried and so on. However, it is generally easier to process the packet at the head of the queue before going to the next one. Busy retry processing can be left open to the implementation

❖ Define a bridge MIB (Management Information Base) – to help management software – new SCAT list
- → Error count tells that something bad happened but it doesn't tell what happened or where it happened
- → Congestion timeout is not serious enough to increase errorCount

❖ **Asynchronous Routing – Table 6**
- → Appears to be well understood
- → PJ – might be formatted/represented in a different way
- → Type error –invalid values
- → Two set of tables – one for bus bound, another for bridge bound
- → Ack pending error or ack address error
- → If node address is wrong then you get ack_missing
- → Should be unambiguous – response address error is ambiguous unless we add more information to it – don't know which node it came from – no such thing or the device says address error
- → ACK address error
- → BACK – programming error

DISCUSSION ON STANDARD COMPLETION DATE

❖ Our goal is to complete technical work by end of 1999. However, after the technical work is complete it takes 1-2 years to submit a final approved draft to the IEEE standards committee. Hence we need to file for an extension for the standard completion date.
❖ A motion was made to have the chairman Dick Scheel to submit a revised PAR to MSC specifying a completion date of June 2000.

- Dave James proposed an amendment to make 2001 December as the completion date instead. The motion was seconded. Choice of the completion date was influenced by the following factors:
  - → Ability of customers of 1394.1 to wait for 1394.1 specification.
  - → 1394a was approved to go to ballot in 1998 February – didn't get mailed out until July 1999, re-circulation ballot will be mailed out next week. It takes one month to come back. If there are no negative votes then the standard may get approved by March 2000 - which is two years from the date the first balloting was approved. IEEE 1394.1 standard may go faster through the BRC process although it usually takes at least one year.
  - → People should not be mislead that the PAR date of December 31st, 2001 is the date for completion of technical specifications.
  - → Amendment accepted – December 31, 2001
  - → Amended motion was passed as follows: For: 9; Against: 0 .

Meeting was adjourned at 4:25 pm.

Handouts distributed at the meeting:

(1) Agenda	(See Page 1)
(2) Minutes of the June 1999 meeting	(See 1394.1 website)
(3) Efficient Resource Reservation for …	Document BR056R00
(4) Draft text submission	Document BR047R06
(5) IEEE 1394.1 SCAT list	Document BR035R05

Attendees of July 22-23, 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. Subrata	Banerjee	Philips Research	+1-914-945-6129	sub@philabs.research.philips.com
3. John	Fuller	Microsoft	+1-425-703-3863	jfuller@microsoft.com
4. David	James	Sony	+1-408-955-6295	dvj@alum.mit.edu
5. Peter	Johansson	Congruent Software	+1-510-527-3926	pjohansson@aol.com
6. Ats	Nakamura	Canon	+81-44-739-6634	atsnaka@bsd.canon.co.jp
7. Yoshikatsu	Niwa	Sony	+81-3-5448-4603	niwa@sm.sony.co.jp
8. Takayuki	Nyu	NEC	+81-44-856-2082	new@ccm.cl.nec.co.jp
9. Earl	Rydell	Rockwell-Collins	+1-319-295-8405	eerydell@collins.rockwell.com
10. Takashi	Sato	Philips Research	+1-914-945-6099	txs@philabs.research.philips.com
11. Dick	Scheel	Sony	+1-408-955-3058	Richard.Scheel@am.sony.com
12. David	Wooten	Compaq	+1-281-518-7231	David.Wooten@compaq.com
13. Yvon	Legaiiais	Thomson Multimedia	+33-299-273798	legallaisy@thmulti.com
14. Frank	Zhao	Panasonic	+1-408-653-4077	zhaof@panasonic.com