Editor's note: This email thread discusses the issues of whether a response should be routed back through the reverse of the path taken by the corresponding request, and whether multiple paths should be allowed between two buses. The emails were posted to the IEEE P1304.1 reflector between 6/25/98 and 7/12/98.

I compiled the thread as best I could from my email archives. I attempted to sort them in chronological order, and I edited out a lot of the repeated messages as people replied to earlier messages. In general, I tried to leave enough of the earlier message to identify what the reply is addressed to.

Dick Scheel
Chair, IEEE P1394.1
8/4/98

June 26

From: DickScheel@aol.com
Date: Thu, 25 Jun 1998 23:20:30 EDT
To: stds-1394-1@majordomo.ieee.org
Subject: [P1394.1]: Minutes + vote announcement

The attached file contains the minutes of the June 9-10 meeting. Many thanks to Subrata Banerjee for producing the minutes quickly. The file will be uploaded to the ftp site (along with soft copy of presentations) when I return to my office next week.

**** Important notice of vote: In the minutes you will find three items that will be decided by email vote. The three items are:

1. May multiple bridges actively connect a pair of buses?
2. Should a request and its response follow the same path?
3. Will we use a subnet architecture?

Please read the minutes to see a more complete description of these items.
Voting will be by email to the chair at dicks@lsi.sel.sony.com. Votes must be received no later than Friday July 10 1998. Between now and the deadline, please discuss these items thoroughly on the working group reflector. The result of the vote will be announced the following week at our meeting in Bath, as well as on the reflector.

In order to be eligible to vote, you must have attended at least two of the last three meetings. Those meetings were: March 19-20 in Tempe AZ
April 27-28 in Newport Beach CA
June 9-10 in St. Petersburg Beach FL

If you are uncertain about your voting status, check the attendance lists in the minutes on the ftp site. I will check all votes against these lists to verify eligibility. We also hope that you will only vote if you feel that you understand the issues. It is acceptable to vote on only one or two of the three issues if you only understand one or two issues. Finally, remember that in IEEE you vote as an individual engineer, not as a company representative.

Note that even if you are not eligible to vote, you may engage in the reflector discussions if you understand the issues enough to offer comments that you think will be helpful to the voters.

If there are any questions about the voting process, please contact me as soon as possible.

Regards,
Dick Scheel
Chair, IEEE P1394.1

Date: Fri, 26 Jun 1998 10:15:23 -0500
From: Neil Morrow <nmorrow@ti.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Minutes + vote announcement

Dick and all,

I think the voting items need some clarifications. For example, if #1 is voted as "yes" we can have loops. Then it follows that we may use a routing table instead of base-bounds. If it is voted as "no" we cannot have loops, then we may use base-bounds routing instead .

When voting on #3 I would like to know whether there can be loops or not.
The subnet architecture seems to address issues with base-bounds routing, but I'm not convinced of the benefit when using routing tables.

So how about rephrasing #3 as:

"Will we use a subnet architecture if loops are *not* allowed?"

or break it into two questions

"Will we use a subnet architecture if a) loops allowed ?
                                       b) loops not allowed ?

Best Regards,

Neil Morrow

DickScheel@aol.com wrote:
> The attached file contains the minutes of the June 9-10 meeting. Many thanks
> to Subrata Banerjee for producing the minutes quickly. The file will be
> uploaded to the ftp site (along with soft copy of presentations) when I return
> to my office next week.
>
> **** Important notice of vote: In the minutes you will find three items that
> will be decided by email vote. The three items are:
>
> 1. May multiple bridges actively connect a pair of buses?
> 2. Should a request and its response follow the same path?
> 3. Will we use a subnet architecture?
[snip]

--
Texas Instruments
Product Development Engineering
MSDS - PCIbus Solutions
Phone: 972-480-6211
Fax: 972-480-6400
email: nmorrow@ti.com

July 2

From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@ieee.org
Subject: Re: [P1394.1]: Minutes + vote announcement
Date: Thu, 02 Jul 1998 21:16:57 +0900

Hi all,

There are no discussion on the reflector about vote as shown below.
I understand the voting items of 1) and 2) are essentially whether the loop bridge are allowed or not, as Neil wrote. I believe that the loop bridge has many advantages and many disadvantages. However these are not cleared. I think more discussion is needed, and I feel these voting items are still open issues.

Furthermore I would like to confirm about the voting policy.
Does the email's voting permit in the P1394.1 officially ?

Best regards.

>**** Important notice of vote: In the minutes you will find three items that
>will be decided by email vote. The three items are:
>
>1. May multiple bridges actively connect a pair of buses?
>2. Should a request and its response follow the same path?
>3. Will we use a subnet architecture? (removed)

Tomoki SAITO
1394 Home Networking Engineer
NEC C&C Media Research Laboratories
4-1-1 Miyazaki Miyamae-ku Kawasaki 216 JAPAN
Telephone: +81-44-856-2082
Facsimile: +81-44-856-2222
Internet : saito@optsys.cl.nec.co.jp

July 3

Date: Fri, 3 Jul 1998 11:53:14 +0900
To: stds-1394-1@ieee.org
Subject: Re: [P1394.1]: Minutes + vote announcement
From: togu@av.crl.sony.co.jp (Kazunobu Toguchi)

All,

I wonder if a 1394node is waiting for a response when once it sends a request and gets its ack_pending.

Let's see the item #2. I like you to think that a request is going through some bridges. Since the request must be split transaction, when one of the portals gets the request, it should send an ack_pending to its sender on the same bus immediately.
Dose the portal have to send its response to the sender like what the 1394-1995 draft shows on the page 29 ?
If yes, I think we can't say 'No' to the item #2.

Toguchi

Kazunobu Toguchi (togu@av.crl.sony.co.jp)
Shirota digital systems lab. Media Processing Labs, Sony Japan
Voice: +81-3-5448-5615, Fax: +81-3-5448-5617

In <199807021217.VAA09081@optsys1.optsys.cl.nec.co.jp>, saito wrote:
| There are no discussion on the reflector about vote as shown below.
| I understand the voting items of 1) and 2) are essentially whether
| the loop bridge are allowed or not, as Neil wrote. I believe that
| the loop bridge has many advantages and many disadvantages. However
| these are not cleared. I think more discussion is needed, and I feel
| these voting items are still open issues.
|
| Furthermore I would like to confirm about the voting policy.
| Does the email's voting permit in the P1394.1 officially ?

Kazunobu Toguchi (togu@av.crl.sony.co.jp)
Shirota digital systems lab. Media Processing Labs, Sony Japan
Voice: +81-3-5448-5615, Fax: +81-3-5448-5617

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@ieee.org
Subject: [P1394.1]: Logical loop architecture
Date: Fri, 03 Jul 1998 12:33:01 +0900

Hello all,

Firstly I would like to confirm the meaning of the following voting issue:

1. May multiple bridges actively connect a pair of buses?

I understand that if voting result of the issue is "yes", "multiple bridges between a pair of buses" architecture will be accepted as an "optional" architecture. On the other hand, if the result is "no", we will not accept logical loop between a pair of buses, but whether we accept "physical" looping connection or not still remains as an open issue.

Is my understanding correct?

Anyway, the logical loop architecture may bring us some advantages of high capacity and reliability, but it will also bring us following problems to be solved. I hope that people for logical loop would show us some ideas to solve these problems on this reflector.

My position is against logical loop architecture because it seems for me to very complex. However, if it is clarified that there are simple ways to support this architecture, I will change my position.

a)clock synchronization
If we transfer isochronous stream packets through a bridge, clock frequency of both buses must be synchronized. "go fast" and "go slow" commands will be used for this purpose. I think that the bridge will initiate these commands for synchronization, because only the bridge can monitor clock frequencies of both buses. If we accept multiple-bridge architecture and multiple bridges initiate these commands independently, I feel clock synchronization will be impossible. So we will have to select only one bridge which can initiate the clock synchronization commands to one local cycle master before starting routing isochronous stream. How do we select that bridge?

b)routing path control
When there are many routes between two buses, we will need to select route(s) before transmitting packets.
The following ways may be possible to select the route(s):

b-1) Requester adds routing information into packets to be transmitted
b-2) Requester dynamically changes routing map (or bounds) in each bridge

Which way is better? or are there more excellent ways?

c)stream control register
There may be some applications that needs bidirectional packet transmission on a single stream channel. I suppose video conference is categorized into this. For supporting such an application on Serial Bus net, we will need to allocate two pairs of STREAM_CONTROL registers (one pair for bus A to bus B, and the other for bus B to bus A). In multiple-bridge architecture, I think these pairs must be set in a single bridge. If not (one pair in bridge A and the other in bridge B, or two pairs in every bridge), stream packets will go around forever in looping path. So we will have to prohibit such a usage of STREAM_CONTROL registers and may need some monitoring facilities to avoid packet looping.

If we need such facilities, are bridges suitable for implementing these? Or is there something else more sutable? Or... , is this problem out of the scope of P1394.1?

Best regards,
Wataru Domon

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Minutes + vote announcement
Date: Fri, 03 Jul 1998 14:49:11 +0900

Hello Toguchi-san and all,

I think there is no relation between split transaction and voting item #2(Should a request and its response follow the same path?).

Let's consider a case that node #1 on bus #1 sends a request pacekt to node #2 on bus #2 and there are bridge #1 and bridge #2 between bus #1 and bus #2.

When the request packet is received by a portal of bridge #1, the portal sends an acknowledge packet with its ack_code of ack_pending. This ack packet does not contain source ID, so node #1 can not specify who sent it, unless node #1 selects bridge #1 for routing the request packet.

On the other hand, a response packet sent by node #2 contains destination ID of node #1 and source ID of node #2, but it does not contain routing informaiton. So node #1 will surely able to receive this response packet even if it has been routed via bridge #2.

Best regards,
Wataru Domon

>I wonder if a 1394node is waiting for a response
>when once it sends a request and gets its ack_pending.
>
>Let's see the item #2. I like you to think that a request is
>going through some bridges.
>Since the request must be split transaction,
>when one of the portals gets the request, it should send
>an ack_pending to its sender on the same bus immediately.
>Dose the portal have to send its response to the sender
>like what the 1394-1995 draft shows on the page 29 ?
>If yes, I think we can't say 'No' to the item #2.

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 03 Jul 1998 15:48:31 +0900

I changed the subject.

I think that domon-san's comment is correct when node #2 response using splitt transaction and concatenated transaxtion except unified transaction.

When node #2 send a ack packet by unified transaction (this is the only response of write request, as you know), the same path may be needed, since this ack has no node ID information.

>I think there is no relation between split transaction and
>voting item #2(Should a request and its response follow the
>same path?).

>
>Let's consider a case that node #1 on bus #1 sends a request
>pacekt to node #2 on bus #2 and there are bridge #1 and
>bridge #2 between bus #1 and bus #2.
>
>When the request packet is received by a portal of bridge #1,
>the portal sends an acknowledge packet with its ack_code of
>ack_pending. This ack packet does not contain source ID, so
>node #1 can not specify who sent it, unless node #1 selects
>bridge #1 for routing the request packet.
>
>On the other hand, a response packet sent by node #2 contains
>destination ID of node #1 and source ID of node #2, but it
>does not contain routing informaiton. So node #1 will surely
>able to receive this response packet even if it has been routed
>via bridge #2.

Tomoki SAITO
   1394 Home Networking Engineer
   NEC C&C Media Research Laboratories
     4-1-1 Miyazaki Miyamae-ku Kawasaki 216 JAPAN
        Telephone: +81-44-856-2082
        Facsimile: +81-44-856-2222
        Internet : saito@optsys.cl.nec.co.jp

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 03 Jul 1998 17:35:15 +0900

Saito-san,

I think there is a way for supporting voting item #2 as follows, even if node #2 uses unified transaction.

The way is that bridge #2 monitors all request packets sent by bridge #1 to a node on bus #2. When bridge #2 knows that bridge #1 sent a request packet to node #2, bridge #2 waits a response packet or an acknowledge packet from node #2. If a response packet is received, bridge #2 only transfers the response packet on bus #1 side. On the other hand, if an ack packet is received, bridge #2 will need to generate a response packet for split transaction on bus #1 and send it to node #1.

Note that generating a response packet for split transaction is independent of voting item #2. That will be necessary even if there is only one bridge connected to bus #1 and bus #2 when transaction on bus #2 is unified transaction.

Regards,
Wataru Domon

saito@optsys1.optsys.cl.nec.co.jp wrote:
>I changed the subject.
>
>I think that domon-san's comment is correct when node #2
>response using splitt transaction and concatenated transaxtion
>except unified transaction.
>
>When node #2 send a ack packet by unified transaction (this is
>the only response of write request, as you know), the same path
>may be needed, since this ack has no node ID information.

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

Date: Fri, 3 Jul 1998 20:26:08 +0900
To: stds-1394-1@majordomo.ieee.org
Subject: [P1394.1]: [Item 2] Should a request and its response follow the same path?
From: togu@av.crl.sony.co.jp (Kazunobu Toguchi)

Domon-san and all,

Maybe we need more bridges and buses.
Suppose there are some bridges between a source and a destination bus.
I like you to focus on one of the bridge portals on path for a request.

My point is whether or not the bridge portal which sent a request and got its ack_pending from someone (the destination or a bridge portal on the path. I don't care which) should be waiting for its response.

If a portal, or node, sends request and gets its ack_pending, I think, it really loves to see its response. :) If the portal doesn't care, I don't care.

Regards,
Toguchi

In <199807030550.OAA19318@optsys1.optsys.cl.nec.co.jp>, domon wrote:
| I think there is no relation between split transaction and
| voting item #2(Should a request and its response follow the
| same path?).
|
| Let's consider a case that node #1 on bus #1 sends a request
| pacekt to node #2 on bus #2 and there are bridge #1 and
| bridge #2 between bus #1 and bus #2.
|
| When the request packet is received by a portal of bridge #1,
| the portal sends an acknowledge packet with its ack_code of
| ack_pending. This ack packet does not contain source ID, so
| node #1 can not specify who sent it, unless node #1 selects
| bridge #1 for routing the request packet.
|
| On the other hand, a response packet sent by node #2 contains
| destination ID of node #1 and source ID of node #2, but it
| does not contain routing informaiton. So node #1 will surely
| able to receive this response packet even if it has been routed
| via bridge #2.

Kazunobu Toguchi (togu@av.crl.sony.co.jp)
Shirota digital systems lab. Media Processing Labs, Sony Japan
Voice: +81-3-5448-5615, Fax: +81-3-5448-5617

From: Burklin Helmut <burklinh@thmulti.com>
To: stds-1394-1@majordomo.ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 3 Jul 1998 09:42:00 +0200

Me too, I support domo-san's comment.

On the other side, I have a doubt about the unified transactions. If a transaction is split or not is not decided by the source but by the destination. Due to the delay inherent in a bridge, this can not be the final destination, but only the first intermediate destination, i.e. the bridge portal itself. And since the bridge portal will not be able to decide which response code to send, before it has itself got the response from the final destination (or the next bridge on the route), it has no choice: it will always have to use the split procedure.
The only way to guarantee a unified transaction would be rather tricky, and I have a doubt in the reliability: the bridge answers 'busy' to the first reception of the request, but forwards it nevertheless to the destination.
It continues to answer 'busy', until having received the response, and then will answer in a unified transaction with the correct response code. A problem arises if the source withdraws its request, not knowing that it has already been forwarded.
Saito-san, in your answer it could be understood that a split write transaction is forbidden by the standard. This is not my understanding - I think in 3.6.2.1 is written that it depends on the speed of the receiver, if a write is split or unified. Am I wrong?

So, you understand that I don't see an exception to the remarks of Domo.

In my mind the multiple port bridges still are a more important item, than the multiple bridges connecting a same pair of busses - but anyway I'm not entitled to vote. Since from Europe it's more difficult to attend West Coast meetings than those in Bath...

Best regards

Helmut Burklin

==========
Helmut BURKLIN
Technical Manager Home Communications
Home Communications and Servers Laboratory Corporate Research Rennes

THOMSON multimedia R&D France SNC
1, av. de Belle Fontaine
B.P. 19
35511 Cesson-Sevigne cedex
France

tel +33 2 99 27 36 27 (direct)
tel +33 2 99 27 30 00 (std)
fax +33 2 99 27 30 01
email burklinh@thmulti.com
==========

From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: [P1394.1]: Voting-Item[2] the same path
Date: Friday 1998 July 03 7:48

I changed the subject.

I think that domon-san's comment is correct when node #2 response using splitt transaction and concatenated transaxtion except unified transaction.

When node #2 send a ack packet by unified transaction (this is the only response of write request, as you know), the same path may be needed, since this ack has no node ID information.

From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 03 Jul 1998 20:15:52 +0900

Hello Helmut,

I think that you misunderstand my comment.
What I am saying is the case that a responder node tries the unified transaction while all bridges on the way to the path try splitt transaction as shown below.

```
  requester ----- bridge ----- bridge ----- responder
    request -------->
       <--------ack pedding (splitt trans.)
                  request------->
                    <---------ack pedding (splitt trans.)
                               request--------->
                                  <---------ack(complete) (unified trans.)
```

Best regards.

>Me too, I support domo-san's comment.
>On the other side, I have a doubt about the unified transactions. If a
>transaction is split or not is not decided by the source but by the
>destination. Due to the delay inherent in a bridge, this can not be the
>final destination, but only the first intermediate destination, i.e. the
>bridge portal itself. And since the bridge portal will not be able to decide
>which response code to send, before it has itself got the response from the
>final destination (or the next bridge on the route), it has no choice: it
>will always have to use the split procedure.
>The only way to guarantee a unified transaction would be rather tricky, and
>I have a doubt in the reliability: the bridge answers 'busy' to the first
>reception of the request, but forwards it nevertheless to the destination.
>It continues to answer 'busy', until having received the response, and then
>will answer in a unified transaction with the correct response code. A
>problem arises if the source withdraws its request, not knowing that it has
>already been forwarded.
>Saito-san, in your answer it could be understood that a split write
>transaction is forbidden by the standard. This is not my understanding - I
>think in 3.6.2.1 is written that it depends on the speed of the receiver, if
>a write is split or unified. Am I wrong?
>
>So, you understand that I don't see an exception to the remarks of Domo.
>
>In my mind the multiple port bridges still are a more important item, than
>the multiple bridges connecting a same pair of busses - but anyway I'm not
>entitled to vote. Since from Europe it's more difficult to attend West Coast
>meetings than those in Bath...
>
>Helmut Burklin

Tomoki SAITO
   1394 Home Networking Engineer
   NEC C&C Media Research Laboratories
     4-1-1 Miyazaki Miyamae-ku Kawasaki 216 JAPAN
        Telephone: +81-44-856-2082
        Facsimile: +81-44-856-2222
        Internet : saito@optsys.cl.nec.co.jp

July 4

Date: Sat, 04 Jul 1998 11:53:52 -0500
From: Neil Morrow <nmorrow@ti.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Logical loop architecture

Domon-san and all,

Although I'm not a voting member in this vote, I thought I'd give my two cents worth anyway.

[1] Loops. I understand the issue same as you, and have similar reservations as what you express below.

[2] RQ/RS Path. I understand the issue is that a response packet is generated at the speed of the request in split transactions, and two different paths for requests and responses would make this method unreliable. So I think RQ and RS packets should follow the same path, and one way for that is what you describe in a separate email given below (this would be rather complicated).

A much easier method is to vote "NO" on item #1 (Loops).

"The way is that bridge #2 monitors all request packets sent by bridge #1 to a node on bus #2. When bridge #2 knows that bridge #1 sent a request packet to node #2, bridge #2 waits a response packet or an acknowledge packet from node #2. If a response packet is received, bridge #2 only transfers the response packet on bus #1 side. On the other hand, if an ack packet is received, bridge #2 will need to generate a response packet for split transaction on bus #1 and send it to node #1."

Best Regards,
Neil Morrow

domon@ccm.cl.nec.co.jp wrote:
> Firstly I would like to confirm the meaning of the following
> voting issue:
>
> 1. May multiple bridges actively connect a pair of buses?
>
> I understand that if voting result of the issue is "yes", "multiple
> bridges between a pair of buses" architecture will be accepted as
> an "optional" architecture. On the other hand, if the result is "no",
> we will not accept logical loop between a pair of buses, but whether
> we accept "physical" looping connection or not still remains as an
> open issue.
>
> Is my understanding correct?
>
> Anyway, the logical loop architecture may bring us some advantages
> of high capacity and reliability, but it will also bring us following
> problems to be solved. I hope that people for logical loop would
> show us some ideas to solve these problems on this reflector.
>
> My position is against logical loop architecture because it seems
> for me to very complex. However, if it is clarified that there are

> simple ways to support this architecture, I will change my position.
>
> a)clock synchronization
> If we transfer isochronous stream packets through a bridge, clock
> frequency of both buses must be synchronized. "go fast" and "go
> slow" commands will be used for this purpose. I think that the
> bridge will initiate these commands for synchronization, because
> only the bridge can monitor clock frequencies of both buses. If we
> accept multiple-bridge architecture and multiple bridges initiate
> these commands independently, I feel clock synchronization will
> be impossible. So we will have to select only one bridge which
> can initiate the clock synchronization commands to one local cycle
> master before starting routing isochronous stream. How do we
> select that bridge?
>
> b)routing path control
> When there are many routes between two buses, we will need to select
> route(s) before transmitting packets.
> The following ways may be possible to select the route(s):
>
> b-1) Requester adds routing information into packets to be
> transmitted
> b-2) Requester dynamically changes routing map (or bounds) in
> each bridge
>
> Which way is better? or are there more excellent ways?
>
> c)stream control register
> There may be some applications that needs bidirectional packet
> transmission on a single stream channel. I suppose video
> conference is categorized into this. For supporting such an
> application on Serial Bus net, we will need to allocate two
> pairs of STREAM_CONTROL registers (one pair for bus A to
> bus B, and the other for bus B to bus A).
> In multiple-bridge architecture, I think these pairs must be
> set in a single bridge. If not (one pair in bridge A and the
> other in bridge B, or two pairs in every bridge), stream
> packets will go around forever in looping path. So we will
> have to prohibit such a usage of STREAM_CONTROL registers
> and may need some monitoring facilities to avoid packet looping.
>
> If we need such facilities, are bridges suitable for implementing
> these? Or is there something else more sutable? Or... ,
> is this problem out of the scope of P1394.1?
--

Texas Instruments
Product Development Engineering
MSDS - PCIbus Solutions
Phone: 972-480-6211
Fax: 972-480-6400
email: nmorrow@ti.com

July 6

From: Burklin Helmut <burklinh@thmulti.com>
To: stds-1394-1@ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Mon, 6 Jul 1998 10:55:00 +0200

Thank you, Saito-san, indeed,

I misunderstood what you said. To get things completely clear, I completed your schematic a bit, using double lines (===>) for asynchronous packets (with source and destination address) and simple lines for acknowledge packets (8bit).
As you pointed out, the responder may - or may not - answer to a write request in a unified transaction immediately with final acknowledge code (complete, data_error, type_error). The bridge receiving this acknowledge byte will translate it into a fullsized four quadlet write response packet with the same response code. The bridge is able to do this, since it just sent out the request, containing the necessary addresses. All further bridges on the route will now relay this response packet. Note that the acknowledge code for the response packet can never be 'pending'.

```
  requester ----- bridge ----- bridge ----- responder
    request ========>
       <--------ack pedding (splitt trans.)
                 request========>
                   <---------ack pedding (splitt trans.)
                             request========>
                                <---------ack(complete, error) (unified trans.)
                    <======== response (split transaction)
                    --------> ack complete
       <======== response (split transaction)
       --------> ack complete
```

So, if I understood the procedure correctly, there IS a difference between unified and split transactions with regards to the final bridge, the one delivering the request to the responder. But that's all. Finally I don't see any reason why the route for the response needs to be the same as for the request (IF there are different routes possible).

By the way, what happens in the case that one of the bridges answers 'busy' ? I guess that the repetition will be handled by the node that just sent out the packet, and will not be sent back to the origin (requester or responder) - do you confirm this?

Regards
Helmut Burklin

From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Friday 1998 July 03 12:15

I think that you misunderstand my comment.
What I am saying is the case that a responder node tries the unified transaction while all bridges on the way to the path try splitt transaction as shown below.

```
  requester ----- bridge ----- bridge ----- responder
    request -------->
```

```
       <--------ack pedding (splitt trans.)
                 request------->
                   <---------ack pedding (splitt trans.)
                             request--------->
                                <---------ack(complete) (unified trans.)
```

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Mon, 6 Jul 1998 09:06:03 -0500

I'm having a hard time following this thread. I thought question #2 was whether or not we would require that a response take the same path as a request. I don't understand why this has anything to do with whether or not one can do a split write transaction over a bus or not. I would think that we could simply decide if we wanted to have responses follow separate paths or not. I would think that it would be much simpler if we did require requests and responses to follow the same path. I can't think of any particular advantage for having them follow separate paths. So, I would tend to vote 'yes' for question #2.

BTW, I believe that there are some potential scheduling 'optimizations' that we can do if we require that requests and responses follow the same path but that's a separate discussion.

David Wooten

> -----Original Message-----
> From: saito@optsys1.optsys.cl.nec.co.jp [SMTP:saito@optsys1.optsys.cl.nec.co.jp]
> Sent: Friday, July 03, 1998 6:16 AM
> To: stds-1394-1@ieee.org
> Subject: RE: [P1394.1]: Voting-Item[2] the same path
>
> I think that you misunderstand my comment.
> What I am saying is the case that a responder node tries the
> unified transaction while all bridges on the way to the path
> try splitt transaction as shown below.
>
> ```
>   requester ----- bridge ----- bridge ----- responder
>     request -------->
>        <--------ack pedding (splitt trans.)
>                  request------->
>                    <---------ack pedding (splitt trans.)
>                              request--------->
>                                 <---------ack(complete) (unified trans.)
> ```

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Logical loop architecture
Date: Mon, 06 Jul 1998 13:18:48 +0900

Hello Neil and all,

I am also not a voting member and I can see in recent technical discussions only two persons who can vote: Toguchi-san and Saito-san.

As I wrote in my previous email, my current position is against logical loop. So I don't support different RQ/RS paths, either.

I understand the way I mentioned regarding to item #2 is complicated and I don't know whether it works reliably or not.

I also agree the issue you pointed out (the speed of the request in split transactions).

Best Regards,
Wataru Domon.

Neil Morrow <nmorrow@ti.com> wrote:
>Although I'm not a voting member in this vote, I thought I'd give
>my two cents worth anyway.
>
>[1] Loops. I understand the issue same as you, and have similar
>reservations as what you express below.
>
>
>[2] RQ/RS Path. I understand the issue is that a response packet
>is generated at the speed of the request in split transactions, and
>two different paths for requests and responses would make this
>method unreliable. So I think RQ and RS packets should follow
>the same path, and one way for that is what you describe in a
>separate email given below (this would be rather complicated).
>
>A much easier method is to vote "NO" on item #1 (Loops).
>
>"The way is that bridge #2 monitors all request packets sent by
>bridge #1 to a node on bus #2. When bridge #2 knows that bridge
>#1 sent a request packet to node #2, bridge #2 waits a response
>packet or an acknowledge packet from node #2. If a response packet
>is received, bridge #2 only transfers the response packet on
>bus #1 side. On the other hand, if an ack packet is received,
>bridge #2 will need to generate a response packet for split
>transaction on bus #1 and send it to node #1."

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: [Item 2] Should a request and its response follow the same path?
Date: Mon, 06 Jul 1998 13:40:43 +0900

Hello Toguchi-san and all,

I think a bridge portal should be waiting for a response packet after getting ack_pending, because it's a 1394-1995 compliant node.

But I cannot figure out the problem you think. I understand there are some bridges between a source and a destination. If the portal should be waiting for response, why can't we say 'No' to the item #2?

Would you please give us the detail?

Best Regards,
Wataru Domon.

togu@av.crl.sony.co.jp (Kazunobu Toguchi) wrote:

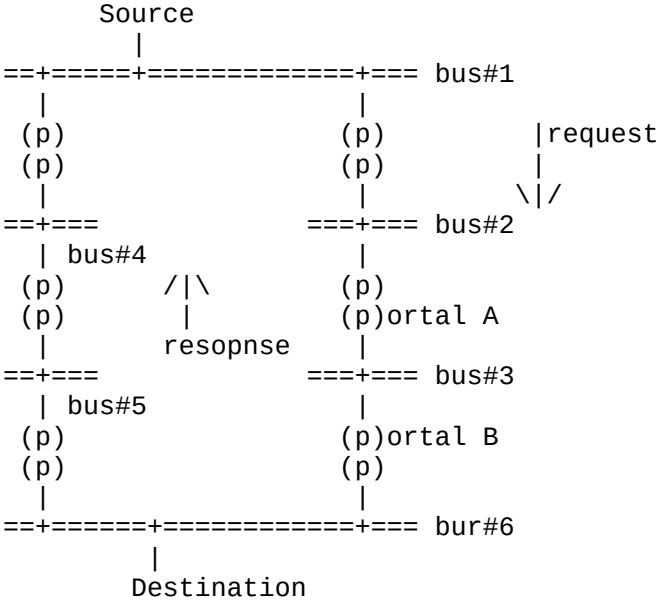
>Maybe we need more bridges and buses.
>Suppose there are some bridges between a source and a destination bus.
>I like you to focus on one of the bridge portals on path for a request.
>
>My point is whether or not the bridge portal which sent a request
>and got its ack_pending from someone (the destination or a bridge portal
>on the path. I don't care which) should be waiting for its response.
>
>If a portal, or node, sends request and gets its ack_pending,
>I think, it really loves to see its response. :)
>If the portal doesn't care, I don't care.

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

Date: Mon, 6 Jul 1998 15:19:31 +0900
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: [Item 2] Should a request and its response follow the same path?
From: togu@av.crl.sony.co.jp (Kazunobu Toguchi)

Dear Domon and all,

Suppose the net has 6 buses with six bridges which is drawn as the figure bellow;



The source is on the bus#1 and the destination node on the bus#6.
A request packet from the source might be coming to the destination through the right side path, but its response keeps the track of the left side path.

In this case, the portal A in the figure should forward the request to the portal B on the bus #3, and I am sure it can get its ack_pending generated by the portal B. And the portal A shall be waiting for its response until waiting time is over, that is exactly what I was asking. Thank you for the answer, Domon-san.
Then the time is going to be over because the response is not coming to the portal A.

If the time-is-up can cause some troubles, I think, we should say the both path must be the same.
I am still wondering if we have any problems caused by the different path.

I think we need more discussions before the voting.

Best Regards,
Toguchi

In <199807060441.NAA09963@optsys1.optsys.cl.nec.co.jp>, domon wrote:
| I think a bridge portal should be waiting for a response packet
| after getting ack_pending, because it's a 1394-1995 compliant node.
|
| But I cannot figure out the problem you think. I understand there
| are some bridges between a source and a destination. If the portal
| should be waiting for response, why can't we say 'No' to the item #2?
|
| Would you please give us the detail?

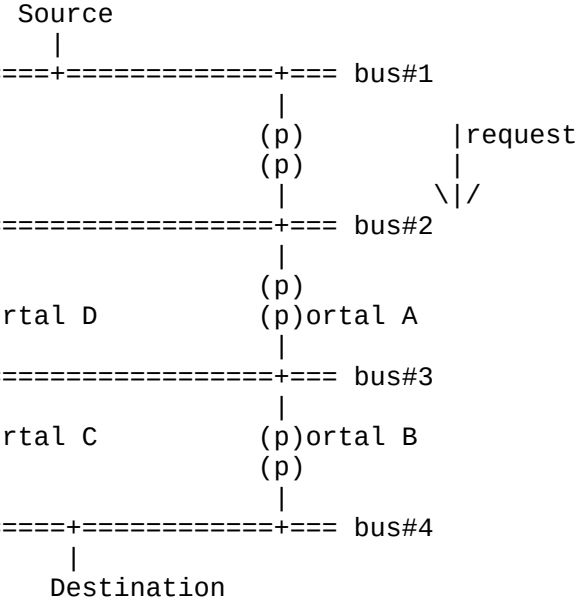
Kazunobu Toguchi (togu@av.crl.sony.co.jp)
Shirota digital systems lab. Media Processing Labs, Sony Japan
Voice: +81-3-5448-5615, Fax: +81-3-5448-5617

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: [Item 2] Should a request and its response follow the same path?
Date: Mon, 06 Jul 1998 17:46:07 +0900

Hello Toguchi-san and all,

Thank you for your detailed explanation. I understand the problem you think, and I agree in the figure you drew that portal A should wait for a response packet which will not come until time-out comes.
I also agree we need to discuss whether the time-out bring us some troubles or not.

I was thinking item #2 using the following configuration (sorry, I prepared this figure by modifying your drawing).



In this figure, there are four bridge portals (A, B, C and D) on bus #3.
Portal A and B are on the request path and portal C and D are on the response path. The most different point from your configuration is that portal A (it's waiting for a response packet) could detect the response packet transmitted by portal C. But another issue arises in this figure that two portals will send an ack packet after receiving the response packet from portal C. Portal D needs to receive the response packet for transferring the packet to bus #2, and portal A also needs to receive for doing split-transaction.

Like the case you showed, we may not be able to say 'No' to the item #2 in this case unless we define the special split transaction for bridges.

Another question occured to me: Does portal B want to send a response packet after sending ack_pending to portal A?
I think the answer is 'yes'.

Best Regards,
Wataru Domon.

togu@av.crl.sony.co.jp (Kazunobu Toguchi) wrote:
>Suppose the net has 6 buses with six bridges which is drawn
>as the figure bellow;
>
>
>   Source
> |
>==+=====+=============+=== bus#1
> | |
> (p)   (p) |request
> (p)   (p) |
> | | \|/
>==+===   ===+=== bus#2
> | bus#4 |
> (p)   /|\   (p)
> (p) | (p)ortal A
> | resopnse |
>==+===   ===+=== bus#3
> | bus#5 |
> (p)   (p)ortal B
> (p)   (p)
> | |
>==+======+============+=== bur#6
> |
>   Destination
>
>
>The source is on the bus#1 and the destination node on the bus#6.
>A request packet from the source might be coming to the destination
>through the right side path, but its response keeps the track of
>the left side path.
>
>In this case, the portal A in the figure should forward the
>request to the portal B on the bus #3, and I am sure it can get
>its ack_pending generated by the portal B. And the portal A shall be
>waiting for its response until waiting time is over,
>that is exactly what I was asking. Thank you for the answer, Domon-san.
>Then the time is going to be over because the response is not
>coming to the portal A.
>
>If the time-is-up can cause some troubles,
>I think, we should say the both path must be the same.

>I am still wondering if we have any problems caused by
>the different path.
>
>I think we need more discussions before the voting.

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

From: "Jerry Hauck" <jerry@macrodesigns.com>
To: <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: [Item 2] Should a request and its response follow the same path?
Date: Mon, 6 Jul 1998 05:57:15 -0700

Hello all,

Upon reading this and similar threads, it is clear that my understanding of bridge operation differs significantly. I can't think of any reason a bridge needs to remember a request subaction which has been successfully forwarded. What am I missing?

In terms of a bridge's queue space:

For requests, queue space isn't guaranteed or reserved until an ack_pending is issued by the bridge. Once the ack_pending is issued, the bridge has complete responsibility (i.e., must maintain queue allocation) for the subaction until the subaction is forwarded, is converted to a response by the bridge, or expires. The subaction is considered "forwarded" when either ack_pending or ack_missing is received by the repeating portal. If an ack_busy code is received by the forwarding portal, the bridge retains the request subaction and retries the packet at a later time (unless expired). If any other ack code is received, the bridge must convert the request into a response subaction and retain it until the response is successfully returned.

For responses, queue space isn't guaranteed or reserved until an ack_complete is issued (or a previously received request subaction is converted into a response). Once the ack_complete is issued, the bridge has complete responsibility for the response subaction until it is forwarded or it expires. The subaction is considered "forwarded" when either ack_complete or ack_missing is received by the forwarding portal.

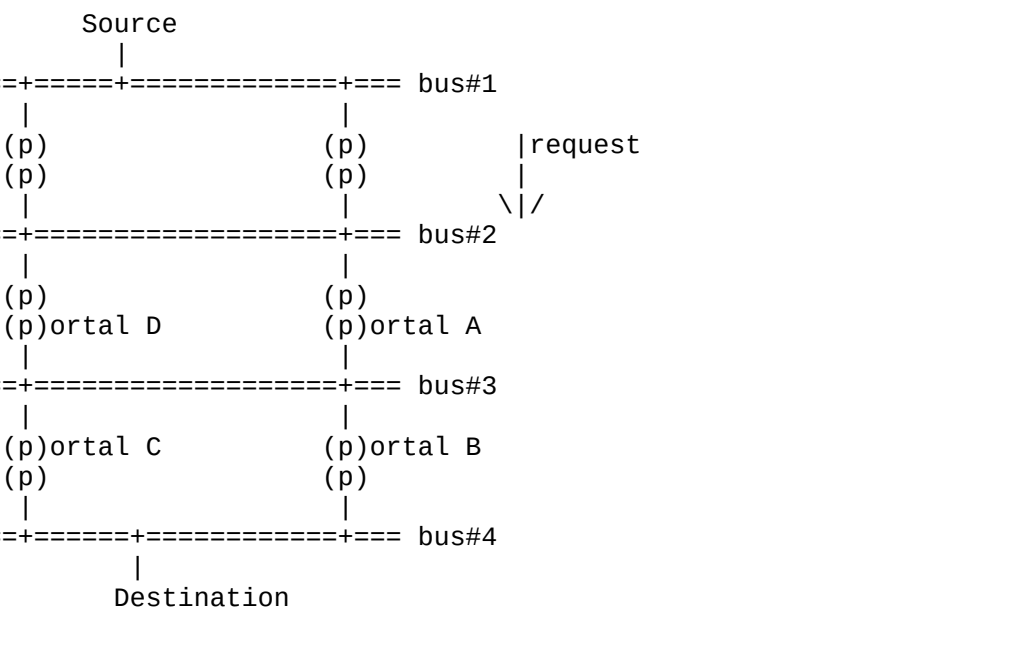
In summary, there is no reason for the bridge to retain any information about a request which has been safely forwarded to the next bridge or to the responder. The split time-out is maintained by the requester ... not the bridges. (The bridges do need to flush expired subactions to prevent the accidental return of a response to a requester after a split-timeout period. However, this expiration period is measured from receipt of a subaction to delivery of the same subaction.)

Perhaps one complication I haven't appreciated fully is the need to make sure that the response is always forwarded at an appropriate speed for the next bridge portal or, ultimately, the requestor.

Regards,
Jerry

==========
Jerry Hauck
Macro Designs, Inc.
1580 Washington Blvd. Tel: 510-668-1006 Fremont, CA 94539
Fax: 510-668-1457 http://www.macrodesigns.com jerry@macrodesigns.com
==========

> -----Original Message-----
> From: domon@ccm.cl.nec.co.jp
> Sent: Monday, July 06, 1998 1:46 AM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: [Item 2] Should a request and its response
> follow the same path?
>
> Thank you for your detailed explanation. I understand the problem
> you think, and I agree in the figure you drew that portal A should
> wait for a response packet which will not come until time-out comes.
> I also agree we need to discuss whether the time-out bring us some
> troubles or not.
>
> I was thinking item #2 using the following configuration (sorry, I
> prepared this figure by modifying your drawing).
>



> In this figure, there are four bridge portals (A, B, C and D) on bus #3.
> Portal A and B are on the request path and portal C and D are on the
> response path. The most different point from your configuration is
> that portal A (it's waiting for a response packet) could detect the
> response packet transmitted by portal C. But another issue arises in
> this figure that two portals will send an ack packet after receiving
> the response packet from portal C. Portal D needs to receive the response
> packet for transferring the packet to bus #2, and portal A also needs
> to receive for doing split-transaction.
>
> Like the case you showed, we may not be able to say 'No' to the item #2
> in this case unless we define the special split transaction for bridges.
>
> Another question occured to me: Does portal B want to send a
> response packet after sending ack_pending to portal A?
> I think the answer is 'yes'.

July 7

From: domon@ccm.cl.nec.co.jp
To: stds-1394-1@majordomo.ieee.org
Subject: RE: [P1394.1]: [Item 2] Should a request and its response follow the same path?
Date: Tue, 07 Jul 1998 19:32:24 +0900

Hello Jerry and all,

I have been misunderstanding in the split transaction that a requester should retry to send a request packet if the requester does not receive a response packet until time-out.

So I withdraw my comment in my previous email.

Best Regards,
Wataru Domon

"Jerry Hauck" <jerry@macrodesigns.com> wrote:
>Upon reading this and similar threads, it is clear that my
>understanding of bridge operation differs significantly. I can't
>think of any reason a bridge needs to remember a request subaction
>which has been successfully forwarded. What am I missing?
>
>In terms of a bridge's queue space:
>
>For requests, queue space isn't guaranteed or reserved until an
>ack_pending is issued by the bridge. Once the ack_pending is issued,
>the bridge has complete responsibility (i.e., must maintain queue
>allocation) for the subaction until the subaction is forwarded, is
>converted to a response by the bridge, or expires. The subaction is
>considered "forwarded" when either ack_pending or ack_missing is
>received by the repeating portal. If an ack_busy code is received by
>the forwarding portal, the bridge retains the request subaction and
>retries the packet at a later time (unless expired). If any other ack
>code is received, the bridge must convert the request into a response
>subaction and retain it until the response is successfully returned.
>
>For responses, queue space isn't guaranteed or reserved until an
>ack_complete is issued (or a previously received request subaction is
>converted into a response). Once the ack_complete is issued, the
>bridge has complete responsibility for the response subaction until it
>is forwarded or it expires. The subaction is considered "forwarded"
>when either ack_complete or ack_missing is received by the forwarding
>portal.
>
>In summary, there is no reason for the bridge to retain any
>information about a request which has been safely forwarded to the
>next bridge or to the responder. The split time-out is maintained by
>the requester ... not the bridges. (The bridges do need to flush
>expired subactions to prevent the accidental return of a response to a
>requester after a split-timeout period. However, this expiration
>period is measured from receipt of a subaction to delivery of the same
>subaction.)
>
>Perhaps one complication I haven't appreciated fully is the need to
>make sure that the response is always forwarded at an appropriate
>speed for the next bridge portal or, ultimately, the requestor.

Wataru DOMON (domon@ccm.cl.nec.co.jp)
Access Network Technology Group
C&C Media Research Laboratories
NEC Corporation

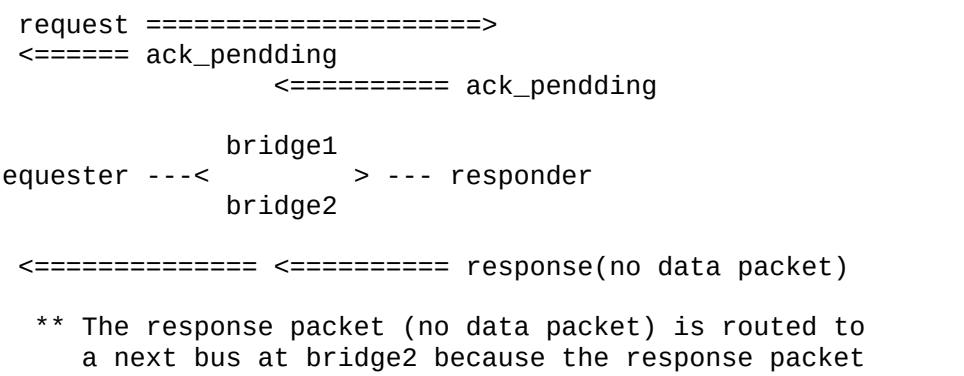
From: saito@optsys1.optsys.cl.nec.co.jp
To: stds-1394-1@ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Tue, 07 Jul 1998 19:15:25 +0900

Hello David,

My thought is following.

[1] In the case that the responder responses by a split
transaction though the different path:

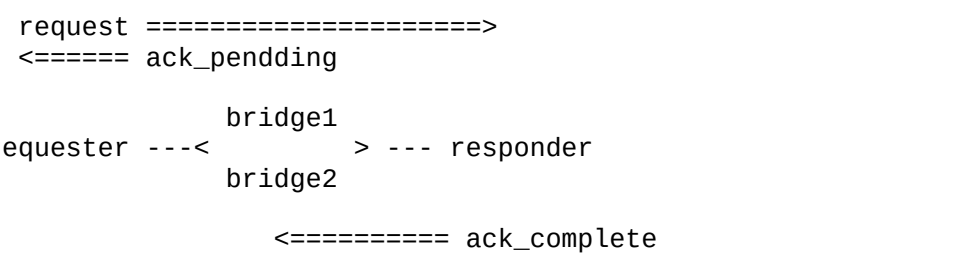
Does the item #2 assume a split transaction at the responder ?
I understand the item #2 indicates the response packet path is
whether the same path or the differrent path as the reqeust
packet path. If the item #2 is limited in this assumption, I
would think the different path is possible as shown in below
figure.



** The response packet (no data packet) is routed to
a next bus at bridge2 because the response packet
has the destination_ID.

[2] In the case that the responder responses by unified
transaction though the different path:

However, regarding the asynchrouns routing, I would point
out the case that the responder responds by a unified
transaction which sends only ack_complete. Since the
ack_complete has no destination_id and bridge2 can not
build the response packet at bridge2 (see below), I would
think that the different path is never used.



** No data packet is not able to built at bridge2.

If item #2 means a split transaction at the responder, I think the routing using either the same path and the different path may be possible. However, taking account of a unified transaction at the responder, we should not use the different path. Then I would vote 'yes' for item #2. Furthermore, since my standing is "NO" for item #1 "loop", the differrent path is never used.

Best reagrds.

>I'm having a hard time following this thread. I thought question #2 was
>whether or not we would require that a response take the same path as a
>request. I don't understand why this has anything to do with whether or not
>one can do a split write transaction over a bus or not. I would think that
>we could simply decide if we wanted to have responses follow separate paths
>or not. I would think that it would be much simpler if we did require
>requests and responses to follow the same path. I can't think of any
>particular advantage for having them follow separate paths. So, I would
>tend to vote 'yes' for question #2.
>
>BTW, I believe that there are some potential scheduling 'optimizations' that
>we can do if we require that requests and responses follow the same path but
>that's a separate discussion.
>
>David Wooten

Tomoki SAITO
1394 Home Networking Engineer
NEC C&C Media Research Laboratories
4-1-1 Miyazaki Miyamae-ku Kawasaki 216 JAPAN
Telephone: +81-44-856-2082
Facsimile: +81-44-856-2222
Internet : saito@optsys.cl.nec.co.jp

Date: Tue, 07 Jul 1998 16:05:04 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Hi all,

Let me participate in this discussion since I'm finally recovered from the holiday weekend.

I am in favor of "a request and the corresponding response shall follow the same path (in the opposite direction) " for the following reasons:

1. No one presented strong arguments against it. Can anybody present merits of not enforcing this?

2. It is more intuitive because there is always a simple requester-responder relationship between requester-bridge, bridge-bridge, or bridge-responder.

3. A solution for unified transactions proposed by Domon-san (i.e., a portal analyzes all the request packets (from other bridges) and the corresponding *ack codes*) is very ugly in my opinion. It may be even impractical to design a link chip that can handle this.

4. As suggested by Dave Wooten, the requests from Bus X to Bus Y still can take another route from the requests from Bus Y to Bus X without breaking this rule.

That's all for now.

Takashi

domon@ccm.cl.nec.co.jp wrote:
> I think there is a way for supporting voting item #2 as follows,
> even if node #2 uses unified transaction.
>
> The way is that bridge #2 monitors all request packets sent by
> bridge #1 to a node on bus #2. When bridge #2 knows that bridge
> #1 sent a request packet to node #2, bridge #2 waits a response
> packet or an acknowledge packet from node #2. If a response packet
> is received, bridge #2 only transfers the response packet on
> bus #1 side. On the other hand, if an ack packet is received,
> bridge #2 will need to generate a response packet for split
> transaction on bus #1 and send it to node #1.
>
> Note that generating a response packet for split transaction is
> independent of voting item #2. That will be necessary even if
> there is only one bridge connected to bus #1 and bus #2 when
> transaction on bus #2 is unified transaction.

==========
Takashi Sato
Philips Research Tel: 914-945-6099 345 Scarborough Road
Fax: 914-945-6580
Briarcliff Manor, NY 10510 E-mail: txs@philabs.research.philips.com
==========

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Tue, 7 Jul 1998 16:13:22 -0500

Apparently there was some misunderstanding about what I was suggesting. The simple thing to do is to have request and responses routed by the destination_ID field of the packet. So, if we want responses from Z to A to follow the same path as requests from A to Z then the easy thing is to also require that requests from Z to A follow the same path at responses from Z to A. If we want requests from Z to A to use a different path than responses from Z to A then we need to decode the tcode of the packet to determine how to route it. We do need to decode tcode in order to know whether we are dealing with an isochronous packet or not so this might not be a bit deal. However, to have requests and responses from a bus take different paths we would need to select among routing tables based on tcode.

What this says is that it is fairly easy to set up unidirectional bridges between buses (path from A to Z not the same as Z to A but everything from A to Z takes the same path and everything from Z to A takes the same path.) So, if we wanted to take advantage of having multiple paths between two buses we could do it but it would mean that request and responses DO NOT follow the same path. Personally, I would prefer that requests and responses follow the same path. If we enforce this, then we can have only one path for data that has a source/destination on one bus and a destination/source on another bus.

David Wooten

-----Original Message-----
From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
Sent: Tuesday, July 07, 1998 3:05 PM
To: stds-1394-1@majordomo.ieee.org

Subject: Re: [P1394.1]: Voting-Item[2] the same path
Let me participate in this discussion since I'm finally recovered from the holiday weekend.

I am in favor of "a request and the corresponding response shall follow the same path (in the opposite direction) " for the following reasons:

1. No one presented strong arguments against it. Can anybody present merits of not enforcing this?

2. It is more intuitive because there is always a simple requester-responder relationship between requester-bridge, bridge-bridge, or bridge-responder.

3. A solution for unified transactions proposed by Domon-san (i.e., a portal analyzes all the request packets (from other bridges) and the corresponding *ack codes*) is very ugly in my opinion. It may be even impractical to design a link chip that can handle this.

4. As suggested by Dave Wooten, the requests from Bus X to Bus Y still can take another route from the requests from Bus Y to Bus X without breaking this rule.

That's all for now.

Date: Tue, 07 Jul 1998 22:52:30 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

David,

Pardon me for my misunderstanding.

If I may paraphrase what you are suggesting: "all the requests and responses from Node X to Node Y follow the same path (in the same direction), but a request and its corresponding response may take different paths".
Am I correct, this time?

I have some questions and comments, provided my interpretation is correct.

1. Do you have a good solution for unified transactions?

2. As you mentioned, the first thing you need to do when you receive a packet is to check its tcode. I believe the overhead of checking whether it is a request or a response is negligible. Some link chips might even store them in separate queues.

3. Suppose when a request and its corresponding response take different paths, and one of the bridges along a path gets corrupted or becomes disabled: How do you find this abnormality? You don't know whether the problem has occurred in the request path or the response path. And quite possibly, you would never find out what has happened if multiple bridges get crippled and request paths and response paths are intertwined in a large network. You might have no way to access or correct the situation in some part of the network in such a case.

On the other hand, if a request and its corresponding response always take the same route (in the opposite direction), then you can inquire each bridge along the way and find out which bridge has gone bad. Your return path is always guaranteed in this case until you reach a bad bridge.

I think this is an important issue, especially for the net manager to inquire or reconfigure the net topology, etc.

Therefore, for better integrity of the network, I believe the following rule serves better : "a request and its corresponding response always take the same path (in the opposite direction)"

Regards,

Takashi

Wooten, David wrote:
> Apparently there was some misunderstanding about what I was suggesting. The
> simple thing to do is to have request and responses routed by the
> destination_ID field of the packet. So, if we want responses from Z to A to
> follow the same path as requests from A to Z then the easy thing is to also
> require that requests from Z to A follow the same path at responses from Z
> to A. If we want requests from Z to A to use a different path than
> responses from Z to A then we need to decode the tcode of the packet to
> determine how to route it. We do need to decode tcode in order to know
> whether we are dealing with an isochronous packet or not so this might not
> be a bit deal. However, to have requests and responses from a bus take
> different paths we would need to select among routing tables based on tcode.
>
> What this says is that it is fairly easy to set up unidirectional bridges
> between buses (path from A to Z not the same as Z to A but everything from A
> to Z takes the same path and everything from Z to A takes the same path.)
> So, if we wanted to take advantage of having multiple paths between two
> buses we could do it but it would mean that request and responses DO NOT
> follow the same path. Personally, I would prefer that requests and
> responses follow the same path. If we enforce this, then we can have only
> one path for data that has a source/destination on one bus and a
> destination/source on another bus.

July 8

Date: Wed, 8 Jul 1998 12:07:26 +0900
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path
From: togu@av.crl.sony.co.jp (Kazunobu Toguchi)

All,

As long as we use a bitmap based routing table for routing asynchronous packets over bridges, an asynchronous packet from A to Z can take the different path that another from Z to A takes.

I think it is better for us to accept that a response can follow the different path as its request did. In other words, we don't need to say that an async packet from A to Z should take the same path as another packet from Z to A takes.
It seems to have not only advantages but also disadvantages so far.

Then we can decide the way to set up unidirectional bridges first.
The bidirectional ones might be left for the future task, that is how to set up routing table for the bidirectional path, etc.
I think the problem related to the item #2 can be reduced to how to set up routing table.

Why don't we make the flexible decition?

Toguchi

David.Wooten wrote:
| Apparently there was some misunderstanding about what I was suggesting. The
| simple thing to do is to have request and responses routed by the
| destination_ID field of the packet. So, if we want responses from Z to A to
| follow the same path as requests from A to Z then the easy thing is to also
| require that requests from Z to A follow the same path at responses from Z
| to A. If we want requests from Z to A to use a different path than
| responses from Z to A then we need to decode the tcode of the packet to
| determine how to route it. We do need to decode tcode in order to know
| whether we are dealing with an isochronous packet or not so this might not
| be a bit deal. However, to have requests and responses from a bus take
| different paths we would need to select among routing tables based on tcode.
|
| What this says is that it is fairly easy to set up unidirectional bridges
| between buses (path from A to Z not the same as Z to A but everything from A
| to Z takes the same path and everything from Z to A takes the same path.)
| So, if we wanted to take advantage of having multiple paths between two
| buses we could do it but it would mean that request and responses DO NOT
| follow the same path. Personally, I would prefer that requests and
| responses follow the same path. If we enforce this, then we can have only
| one path for data that has a source/destination on one bus and a
| destination/source on another bus.

Kazunobu Toguchi (togu@av.crl.sony.co.jp)
Shirota digital systems lab. Media Processing Labs, Sony Japan
Voice: +81-3-5448-5615, Fax: +81-3-5448-5617

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>

Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 09:08:38 -0500

Takashi,

Except for some potential issues with speed traps, I think it would not be terribly difficult (conceptually) for the response back to A to follow a different path than the request from A even in the case of a unified transaction. I don't think the solution is worth trying. That's why I would like to have requests and responses from buses A to B follow the same path as requests and responses from B to A. To rephrase, although I know that other possibilities exist, am in favor of having ALL traffic between a given pair of buses going over the same path. I don't want the path of an asynchronous packet to be dependent on its type.

David Wooten

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@phileabs.research.philips.com]
> Sent: Tuesday, July 07, 1998 9:53 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> Pardon me for my misunderstanding.
>
> If I may paraphrase what you are suggesting: "all the requests and responses from Node X to Node Y follow the same path (in the same direction), but a request and its corresponding response may take different paths".
> Am I correct, this time?
>
> I have some questions and comments, provided my interpretation is correct.
>
> 1. Do you have a good solution for unified transactions?
>
> 2. As you mentioned, the first thing you need to do when you receive a packet is to check its tcode. I believe the overhead of checking whether it is a request or a response is negligible. Some link chips might even store them in separate queues.
>
> 3. Suppose when a request and its corresponding response take different paths, and one of the bridges along a path gets corrupted or becomes disabled: How do you find this abnormality? You don't know whether the problem has occurred in the request path or the response path. And quite possibly, you would never find out what has happened if multiple bridges get crippled and request paths and response paths are intertwined in a large network. You might have no way to access or correct the situation in some part of the network in such a case.
>
> On the other hand, if a request and its corresponding response always take the same route (in the opposite direction), then you can inquire each bridge along the way and find out which bridge has gone bad. Your return path is always guaranteed in this case until you reach a bad bridge.
>
> I think this is an important issue, especially for the net manager to inquire or reconfigure the net topology, etc.
>
> Therefore, for better integrity of the network, I believe the following rule serves better : "a request and its corresponding response always take the same path (in the opposite direction)"

Date: Wed, 08 Jul 1998 11:34:45 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

David,

So, are you suggesting

#2: "a request and its corresponding response shall take the same path (in the opposite direction)"
AND
#2a: "all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A"

?

Am I correct in assuming that you are not opposing to #2 but want the additional constraint #2a?

Takashi

Wooten, David wrote:
> Except for some potential issues with speed traps, I think it would not be
> terribly difficult (conceptually) for the response back to A to follow a
> different path than the request from A even in the case of a unified
> transaction. I don't think the solution is worth trying. That's why I
> would like to have requests and responses from buses A to B follow the same
> path as requests and responses from B to A. To rephrase, although I know
> that other possibilities exist, am in favor of having ALL traffic between a
> given pair of buses going over the same path. I don't want the path of an
> asynchronous packet to be dependent on its type.
--
==========
Takashi Sato
Philips Research Tel: 914-945-6099
345 Scarborough Road Fax: 914-945-6580
Briarcliff Manor, NY 10510 E-mail: txs@philabs.research.philips.com
==========

From: "Bard, Steve" <steve.bard@intel.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 10:00:01 -0700

David and Takashi,

Please pardon me for being so simple, however, I have a short question for the sake of clarifying my understanding...

David, I think Takashi's statement ("#2a: all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A") implies loops would NOT be allowed. Am I correct?

Best Regards,

Steve Bard

-----Original Message-----

From: Takashi Sato [mailto:txs@philabs.research.philips.com]
Sent: Wednesday, July 08, 1998 8:35 AM
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

So, are you suggesting

#2: "a request and its corresponding response shall take the same path (in the opposite direction)"
AND
#2a: "all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A"

?

Am I correct in assuming that you are not opposing to #2 but want the additional constraint #2a?

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 12:02:39 -0500

Yes, yes, yes, yes (is that too many?).

David Wooten

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Wednesday, July 08, 1998 10:35 AM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> So, are you suggesting
>
> #2: "a request and its corresponding response shall take the same path (in the opposite direction)"
> AND
> #2a: "all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A"
>
> ?
>
> Am I correct in assuming that you are not opposing to #2 but want the additional constraint #2a?

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 12:50:59 -0500

Steve,

No.

David Wooten

Oh, I guess I can to better than that. Depends on what you mean by a loop.

Get out a piece of paper and draw three buses (A, B and C). Now connect each bus to each of the other buses (if you end up with more than three bridges, maybe you shouldn't be working on this.) I would expect that A could talk directly to B and C and B could talk directly to C (by inference, B can talk directly to A and C can talk directly to B and A). I would be tempted to call this a loop but...

David Wooten

> -----Original Message-----
> From: Bard, Steve [SMTP:steve.bard@intel.com]
> Sent: Wednesday, July 08, 1998 12:00 PM
> To: 'stds-1394-1@majordomo.ieee.org'
> Subject: RE: [P1394.1]: Voting-Item[2] the same path
>
> Please pardon me for being so simple, however, I have a short question for the sake of clarifying my understanding...
>
> David, I think Takashi's statement ("#2a: all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A") implies loops would NOT be allowed. Am I correct?

Date: Wed, 08 Jul 1998 13:56:20 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Hold on, David,

That is not exactly the same.

Take the following example: There are three buses, A, B, and C. There are two bridges X, and Y between buses A and B. There are one bridge Z between buses B and C.

All the request/response packets between bus A and bus B can go through bridge X, while all the request/response packets between bus A and bus C can go through bridge Y WITHOUT breaking the rules #2 and #2a. Still, this example constitutes a loop (between bus A and bus B), doesn't it?

Takashi

Wooten, David wrote:
> Yes, yes, yes, yes (is that too many?).
>
> > -----Original Message-----
> > From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> > Sent: Wednesday, July 08, 1998 10:35 AM
> > To: stds-1394-1@majordomo.ieee.org
> > Subject: Re: [P1394.1]: Voting-Item[2] the same path
> >
> > So, are you suggesting
> >
> > #2: "a request and its corresponding response shall take the same path (in the opposite direction)"
> > AND
> > #2a: "all the requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A"
> >
> > ?
>

> Am I correct in assuming that you are not opposing to #2 but want the
> additional constraint #2a?

From: "Bard, Steve" <steve.bard@intel.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 12:57:31 -0700

Hello David,

O.K. I did it. I have only three bridges (it was "hard" though - <chuckle>). There exists no loops in Bus A and no loops in Bus B and no loops in Bus C.

However, I am inclined to say that the three bridges create a "loop" of interconnected busses.

But, is that a problem here? (Comparative to a loop within a single bus?)

If a loop of busses can be created by interconnecting them via bridges):

1) responses to requests may (or may not) follow the exact (but opposite direction) path as the request: yes or no. If "no", then a "bunch" of work (rather complicated I think) must be done. If "yes", I am able to visualize a much simpler solution. At least I consider it simpler because I can visualize it and I have a much more difficult time visualizing a solution if separate paths are allowed.

I favor the notion that responses must follow the exact (but opposite direction) path as the request.

2) responses and requests from/to Bus A to Bus B may (or may not) follow the same path as responses and requests from/to Bus B to Bus A: yes or no. If "no" then, again, I see a "bunch" of work just like before and I think this work is rather difficult. If yes, I am able to visualize a much simpler solution (for the same reason as before).

I favor the notion that responses and requests from/to Bus A to Bus B must follow the same path as responses and requests from/to Bus B to Bus A.

One additional item regarding busses interconnected to for a "loop" via bridge connections: If allowed and "single path" (as discussed above) is accepted, then looped busses do not seem to me to be a problem. If, however, "separate" response/request paths are allowed in "looped busses" I think I see a whole bunch or rather "sticky" problems which must be resolved.

If looped busses are NOT allowed, then the "single versus separate path" discussion above is moot (I believe). BUT a whole bunch of OTHER problems must be resolved - such as: how to detect the "looped" busses and what to do about it when they can be detected. I think "non-looped busses/single path" issues may be a bit more complicated to resolve than "looped busses/single path" issues but NOT as difficult to resolve as "looped busses/separate path" issues.

I initially voted to NOT allow looped busses, BUT, if my understanding (as I outlined above) is accurate, I will (most likely) want to retract my vote until further the single versus separate path issue has been resolved.

Dick: Can I do that? It might go something like this: Looped busses: YES - BUT ONLY if response/request path issue resolves to be affirmative for single path. If the response/request path issue resolves to be affirmative for separate, then I vote NO for looped buses.

Best Regards,

Steve Bard

-----Original Message-----
From: Wooten, David [mailto:David.Wooten@compaq.com]
Sent: Wednesday, July 08, 1998 10:51 AM
To: 'stds-1394-1@majordomo.ieee.org'
Subject: RE: [P1394.1]: Voting-Item[2] the same path

No.

Oh, I guess I can to better than that. Depends on what you mean by a loop.
Get out a piece of paper and draw three buses (A, B and C). Now connect each bus to each of the other buses (if you end up with more than three bridges, maybe you shouldn't be working on this.) I would expect that A could talk directly to B and C and B could talk directly to C (by inference, B can talk directly to A and C can talk directly to B and A). I would be tempted to call this a loop but...

From: "Churchill, Richard" <Richard.Churchill@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 15:05:28 -0500

Steve,

Nope, it doesn't ... A 'loop free' implementation is the simplest way to ensure that this is the case, but not the only one.

One way to state the rule is that for any bus X with one or more bridges attached, one and only one bridge will accept and relay a packet with a valid destination bus ID Y, where Y != X and Y != 0x3F, AND if bridge Z attached to bus X accepts and relays packets on bus X addressed to bus Y only bridge Z will accept and pass packets addressed to bus Y from bus X.
(The first part of the rule establishes the existence of a path for a packet X-to-Y, and the second part requires that path to be unique for all packets X-to-Y.) So, if we connect two buses X and Y with bridges Z0 through Zn (n
>= 0), there will be a unique bridge Zi (0 <= i <=n) that handles passing all packets addressed to bus Y from bus X, and a unique bridge Zj (0 <= j <= n) that passes packets addressed to bus X from bus Y, but i and j may or may not be equal. This easily generalizes to cases of more than two buses.

A case that involves a kind of loop yet obeys the rule is as follows. Buses B0, B1 and B2 are connected by bridges X(0,1), X(1,2) and X(2,0), where bridge X(a,b) directly connects buses Ba and Bb. If each bridge is unidirectional (accepts and passes packets not addressed to bus Ba to bus Bb) we have a 'loop' in the sense that responses to requests from bus Ba to bus Bb are passed via different bridges than were used to pass the requests.

Sincerely,

Richard Churchill,
(281)514-6984,
richard.churchill@compaq.com

> -----Original Message-----
> From: Bard, Steve [SMTP:steve.bard@intel.com]
> Sent: Wednesday, July 08, 1998 12:00 PM
> To: 'stds-1394-1@majordomo.ieee.org'

> Subject: RE: [P1394.1]: Voting-Item[2] the same path
>
> Please pardon me for being so simple, however, I have a short question for
> the sake of clarifying my understanding...
>
> David, I think Takashi's statement ("#2a: all the requests from Bus A to Bus
> B shall take the same path as all the requests from Bus B to Bus A")
> implies loops would NOT be allowed. Am I correct?

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 17:26:33 -0500

Takashi,

What is not the same??? I think my answers to your questions were not ambiguous. I am not against #2 but would like the additional #2a constraint.

Your example below doesn't look like a loop to me, it looks like redundant paths (X and Y). I don't believe that there is any particular benefit to having redundant paths between two buses. I don't think we can keep people from hooking them up (enumeration should not fail if this happens) but we shouldn't have both bridges enabled (might just enable the isochronous capabilities if that helps anything.)

David Wooten

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Wednesday, July 08, 1998 12:56 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> Hold on, David,
>
> That is not exactly the same.
>
> Take the following example: There are three buses, A, B, and C. There are two bridges X, and Y between buses A and B. There are one bridge Z between buses B and C.
>
> All the request/response packets between bus A and bus B can go through bridge X, while all the request/response packets between bus A and bus C can go through bridge Y WITHOUT breaking the rules #2 and #2a. Still, this example constitutes a loop (between bus A and bus B), doesn't it?

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 17:38:01 -0500

Richard,

Your description has convinced me. We will not make you write any portion of the specification. Come to think of it, I couldn't understand what Steve was muttering about either. Anyone care to draw a picture to make it clearer what we are talking about? ;-)

David Wooten

-----Original Message-----
From: Churchill, Richard
Sent: Wednesday, July 08, 1998 3:05 PM
To: 'stds-1394-1@majordomo.ieee.org'
Subject: RE: [P1394.1]: Voting-Item[2] the same path

Nope, it doesn't ... A 'loop free' implementation is the simplest way to ensure that this is the case, but not the only one.

One way to state the rule is that for any bus X with one or more bridges attached, one and only one bridge will accept and relay a packet with a valid destination bus ID Y, where Y != X and Y != 0x3F, AND if bridge Z attached to bus X accepts and relays packets on bus X addressed to bus Y only bridge Z will accept and pass packets addressed to bus Y from bus X. (The first part of the rule establishes the existence of a path for a packet X-to-Y, and the second part requires that path to be unique for all packets X-to-Y.) So, if we connect two buses X and Y with bridges Z0 through Zn (n >= 0), there will be a unique bridge Zi (0 <= i <=n) that handles passing all packets addressed to bus Y from bus X, and a unique bridge Zj (0 <= j <= n) that passes packets addressed to bus X from bus Y, but i and j may or may not be equal. This easily generalizes to cases of more than two buses.

A case that involves a kind of loop yet obeys the rule is as follows. Buses B0, B1 and B2 are connected by bridges X(0,1), X(1,2) and X(2,0), where bridge X(a,b) directly connects buses Ba and Bb. If each bridge is unidirectional (accepts and passes packets not addressed to bus Ba to bus Bb) we have a 'loop' in the sense that responses to requests from bus Ba to bus Bb are passed via different bridges than were used to pass the requests.

From: "Bard, Steve" <steve.bard@intel.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 16:56:30 -0700

David,

"Muttering" is a "good" word! I think I said I am OK that busses may create a "loop" when interconnected via bridges IF single paths for response/request were required. Said in "Bard speakease":

"Busses may be interconnected (via bridges) such that a "loop" may be formed. A request (and its corresponding response) shall use a single path (the response will use the same path but in the opposite direction).
Similarly, all requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A."

'Nuff said?

Best Regards,

Steve Bard

-----Original Message-----
From: Wooten, David [mailto:David.Wooten@compaq.com]
Sent: Wednesday, July 08, 1998 3:38 PM
To: 'stds-1394-1@majordomo.ieee.org'
Subject: RE: [P1394.1]: Voting-Item[2] the same path

Your description has convinced me. We will not make you write any portion of the specification. Come to think of it, I couldn't understand what Steve was muttering about either. Anyone care to draw a picture to make it clearer what we are talking about? ;-)

From: John Nels Fuller <jfuller@microsoft.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Wed, 8 Jul 1998 15:59:37 -0700

Hello all,

Time to put my 2 cents in.

I believe that many nodes already assume a constant path between itself and any other node it is communicating with (barring reconfigurations denoted by bus resets). Or stated another way, if node n can talk to node m at S400 then it is assumed that node m can talk to node n at S400 until a topology change is announced. In a bridged environment we would like to make that even simpler: if node n or bus A can talk to node m on bus B at S400 then node m on bus B can talk to node n on bus A at S400. To achieve this I believe that we need to require that the path between nA and mB is the inverse of the path between mB and nA for all packet types and that this path remain constant as much as possible. This leaves the question of is there any reason that we need to require that the path between nA and mB is the same as the path between nA and xB. Well, I think that trying to route packets based on both source node ID and destination node ID requires every bridge in the path to know the routing requirements for every node on both bus A and bus B. Since bridges would have to route not just between buses A and B but all buses then every bridge would have to have knowledge about every node on all buses just to route packets, this leads me to say that it is unworkable to do that. What I'm left with is a requirement that the path between bus A and bus B be constant and be the inverse of the path from bus B to bus A.

_____

John Nels Fuller
Software Design Engineer, Windows NT
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Voice: (425) 703-3863
Fax: (425) 93 MSFAX

-----Original Message-----
From: Bard, Steve [mailto:steve.bard@intel.com]
Sent: Wednesday, July 08, 1998 12:58 PM
To: 'stds-1394-1@majordomo.ieee.org'
Subject: RE: [P1394.1]: Voting-Item[2] the same path

O.K. I did it. I have only three bridges (it was "hard" though - <chuckle>). There exists no loops in Bus A and no loops in Bus B and no loops in Bus C.

However, I am inclined to say that the three bridges create a "loop" of interconnected busses.

But, is that a problem here? (Comparative to a loop within a single bus?)

If a loop of busses can be created by interconnecting them via bridges):

1) responses to requests may (or may not) follow the exact (but opposite direction) path as the request: yes or no. If "no", then a "bunch" of work (rather complicated I think) must be done. If "yes", I am able to visualize a much simpler solution. At

least I consider it simpler because I can visualize it and I have a much more difficult time visualizing a solution if separate paths are allowed.

I favor the notion that responses must follow the exact (but opposite direction) path as the request.

2) responses and requests from/to Bus A to Bus B may (or may not) follow the same path as responses and requests from/to Bus B to Bus A: yes or no. If "no" then, again, I see a "bunch" of work just like before and I think this work is rather difficult. If yes, I am able to visualize a much simpler solution (for the same reason as before).

I favor the notion that responses and requests from/to Bus A to Bus B must follow the same path as responses and requests from/to Bus B to Bus A.

One additional item regarding busses interconnected to for a "loop" via bridge connections: If allowed and "single path" (as discussed above) is accepted, then looped busses do not seem to me to be a problem. If, however, "separate" response/request paths are allowed in "looped busses" I think I see a whole bunch or rather "sticky" problems which must be resolved.

If looped busses are NOT allowed, then the "single versus separate path" discussion above is moot (I believe). BUT a whole bunch of OTHER problems must be resolved - such as: how to detect the "looped" busses and what to do about it when they can be detected. I think "non-looped busses/single path" issues may be a bit more complicated to resolve than "looped busses/single path" issues but NOT as difficult to resolve as "looped busses/separate path" issues.

I initially voted to NOT allow looped busses, BUT, if my understanding (as I outlined above) is accurate, I will (most likely) want to retract my vote until further the single versus separate path issue has been resolved.

Dick: Can I do that? It might go something like this: Looped busses: YES - BUT ONLY if response/request path issue resolves to be affirmative for single path. If the response/request path issue resolves to be affirmative for separate, then I vote NO for looped buses.

July 9

Date: Thu, 09 Jul 1998 08:49:55 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

David,

What I meant is that a combination of the rules #2 and #2a does not necessarily imply 'no loop', as suggested by you to Steve.

In my opinion, it is a loop if there are more-than-one active bridges between two buses in parallel. Maybe, we should come up with a clear definition of 'a loop'.

I believe that one of the main reasons, if not the primary reason, why we want to allow 'loops' is to add redundant (I would rather call it alternate or supplemental) paths between buses so as to reduce congestion (isochronous or asynchronous) on particular bridges. The example I gave you falls into this category.

Regards,

Takashi

Wooten, David wrote:
> What is not the same??? I think my answers to your questions were not
> ambiguous. I am not against #2 but would like the additional #2a
> constraint.
>
> Your example below doesn't look like a loop to me, it looks like redundant
> paths (X and Y). I don't believe that there is any particular benefit to
> having redundant paths between two buses. I don't think we can keep people
> from hooking them up (enumeration should not fail if this happens) but we
> shouldn't have both bridges enabled (might just enable the isochronous
> capabilities if that helps anything.)

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Thu, 9 Jul 1998 09:04:29 -0500

Takashi,

As I mentioned in another email, I would not call having two brides between two buses a loop, I would call it redundant bridges. Using that terminology, I would say that I would not be in favor of having redundant paths between two buses except for the possibility of getting more isochronous resources. If there are multiple paths between two buses (due to either multiple direct or multiple indirect connections) only one path will be uses to carry traffic between the two buses. The reason for this is that I would like to make asynchronous routing dependent only on the destination_ID field of the packet.

I would like to leave the problem of redundant bridges to reduce congestion to the skill of bridge manufacturers. I believe that there are many 'simple' ways to solve this problem (e.g., just add more memory to an existing bridge) and I don't believe that we need to specify a way in which this must be done.

What might be 'interesting' for us to work on to deal with 'congestion' are the arbitration rules for a bridge. We might relieve the congestion on a bridge simply by giving it more shots at sending packets within a fairness interval. Deciding how to set the 'unfair' register in a bridge might become an interesting exercise.

David Wooten

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Thursday, July 09, 1998 7:50 AM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> What I meant is that a combination of the rules #2 and #2a does not necessarily imply 'no loop', as suggested by you to Steve.
>
> In my opinion, it is a loop if there are more-than-one active bridges between two buses in parallel. Maybe, we should come up with a clear definition of 'a loop'.
>
> I believe that one of the main reasons, if not the primary reason, why we want to allow 'loops' is to add redundant (I would rather call it alternate or supplemental) paths between buses so as to reduce congestion (isochronous or asynchronous) on particular bridges. The example I gave you falls into this category.

Date: Thu, 09 Jul 1998 11:30:22 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

David,

What you call redundant bridges are not always redundant, but sometimes necessary. Don't forget that there are natural limitations to bridge's capability. The number of isochronous streams and/or the data rate that one bridge can handle can be very limited, depending on its CPU power and its internal fabric. Especially, this is very true for wireless bridges, whose internal fabric (RF, etc.) is a scarce resource.

As for the routing, you can still make the routing solely based on the destination busID even if there are loops. What you have to do is to set all the routing tables on a bus so that there are no overlaps in their destination busIDs. For example, if you set bit X of the destination_busID register in one bridge, then you have to clear bit X of that register in all other bridges on the same bus. In other words, there is only one way out from each bus for each packet, depending on its destination busID. You just need to setup routing tables in the network so that there is no dead-end or traps for packets.

BTW, am I the only one who is receiving messages from this reflector after over a half hour later from when they are posted?

Regards,

Takashi

Wooten, David wrote:
> As I mentioned in another email, I would not call having two brides between
> two buses a loop, I would call it redundant bridges. Using that
> terminology, I would say that I would not be in favor of having redundant

> paths between two buses except for the possibility of getting more
> isochronous resources. If there are multiple paths between two buses (due
> to either multiple direct or multiple indirect connections) only one path
> will be uses to carry traffic between the two buses. The reason for this is
> that I would like to make asynchronous routing dependent only on the
> destination_ID field of the packet.
>
> I would like to leave the problem of redundant bridges to reduce congestion
> to the skill of bridge manufacturers. I believe that there are many
> 'simple' ways to solve this problem (e.g., just add more memory to an
> existing bridge) and I don't believe that we need to specify a way in which
> this must be done.
>
> What might be 'interesting' for us to work on to deal with 'congestion' are
> the arbitration rules for a bridge. We might relieve the congestion on a
> bridge simply by giving it more shots at sending packets within a fairness
> interval. Deciding how to set the 'unfair' register in a bridge might
> become an interesting exercise.

Date: Thu, 9 Jul 1998 12:03:36 -0700
From: Jim_Busse@ccgate.sj.nec.com (Jim Busse)
Subject: Re[2]: [P1394.1]: Voting-Item[2] the same path
To: stds-1394-1@majordomo.ieee.org

David:

If loops are permitted, and the return path is not necessarily the send path, would it still be possible to specify the path for the send and return in order to optimize speed of transmission and throughput optimization? In some circumstances, I would like to take the fastest path for the quickest possible response.

Steve: "Busses" are members of my family. "Buses" are more than one "Bus". <BG>.

Jim Busse

_____ Reply Separator _____
Subject: Re: [P1394.1]: Voting-Item[2] the same path
Author: Takashi Sato <txs@philabs.research.philips.com> at ccgate
Date: 7/9/98 11:30 AM

What you call redundant bridges are not always redundant, but sometimes necessary. Don't forget that there are natural limitations to bridge's capability. The number of isochronous streams and/or the data rate that one bridge can handle can be very limited, depending on its CPU power and its internal fabric. Especially, this is very true for wireless bridges, whose internal fabric (RF, etc.) is a scarce resource.

As for the routing, you can still make the routing solely based on the destination busID even if there are loops. What you have to do is to set all the routing tables on a bus so that there are no overlaps in their destination busIDs. For example, if you set bit X of the destination_busID register in one bridge, then you have to clear bit X of that register in all other bridges on the same bus. In other words, there is only one way out from each bus for each packet, depending on its destination busID. You just need to setup routing tables in the network so that there is no dead-end or traps for packets.

BTW, am I the only one who is receiving messages from this reflector after over a half hour later from when they are posted?

Wooten, David wrote:

> As I mentioned in another email, I would not call having two brides between
> two buses a loop, I would call it redundant bridges. Using that
> terminology, I would say that I would not be in favor of having redundant
> paths between two buses except for the possibility of getting more
> isochronous resources. If there are multiple paths between two buses (due
> to either multiple direct or multiple indirect connections) only one path
> will be uses to carry traffic between the two buses. The reason for this is
> that I would like to make asynchronous routing dependent only on the
> destination_ID field of the packet.
>
> I would like to leave the problem of redundant bridges to reduce congestion
> to the skill of bridge manufacturers. I believe that there are many
> 'simple' ways to solve this problem (e.g., just add more memory to an
> existing bridge) and I don't believe that we need to specify a way in which
> this must be done.
>
> What might be 'interesting' for us to work on to deal with 'congestion' are
> the arbitration rules for a bridge. We might relieve the congestion on a
> bridge simply by giving it more shots at sending packets within a fairness
> interval. Deciding how to set the 'unfair' register in a bridge might
> become an interesting exercise.

From: "Churchill, Richard" <Richard.Churchill@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Thu, 9 Jul 1998 11:16:43 -0500

Takashi-san,

Topologically speaking, an easy way to say, "There are loops," is to say something like, "For a network of n interconnected buses, there are at least n bridges attached to separate buses at each end." There is no difference between a "redundant path directly between 'adjacent' nodes" and a loop.
This is related to the question of how you can tell when an individual bus has a loop ... For n nodes, you should have n-1 cables connecting them, and more than n-1 means you have a loop. Whether there are two bridges directly connecting buses Bi and Bj or some other configuration, the additional bridge creates a redundant path, and thus a loop, just as inserting an additional cable creates a loop in a bus.

I like the term "redundant," and don't see why we would want to say "supplemental" or "alternative" instead. Redundancy is a well defined and understood technical term, which in technical applications involving quality of service, safety, etc., is devoid of any pejorative sense. Calling a loop a redundant path or vice versa doesn't bother me at all. They are the same.

I also would like to allow loops for the reasons you cite. However, we need to recognize the fact that allowing unrestricted loops (as many additional bridges in a network as you want, wherever you want) creates problems in routing, and an additional burden in network management -- big problems and burdens. I don't see how, during the process of initializing paths in a CYCLIC network, we are going to be able to adequately guess how to configure the paths to correctly deal with the problems you are trying to solve. An acyclic tree is far easier to manage. So, as you are working to resolve the problems entailed by loops, keep in mind that we may need to drop support for loops (other than loop healing) in favor of finishing this standard in a timely manner, as well as to keep costs and complexity low.

Please see further comments embedded below.

> -----Original Message-----

> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Thursday, July 09, 1998 7:50 AM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> What I meant is that a combination of the rules #2 and #2a does not necessarily
> imply 'no loop', as suggested by you to Steve.
>
[RLC] As I understand them, rules 2 (a response from bus Bb to bus Ba will take the same -- but reversed -- path as the original request from bus Ba to bus Bb) and 2a (requests from bus Ba to bus Bb shall follow the same path as responses from bus Ba to bus Bb) taken together prevent logical loops (where packets passed between buses Ba and Bb could take more than one path), but do not preclude physical loops (there may be physical paths between any pair of buses that are not used or may be disabled). So, are you saying that these rules combined ALLOW LOGICAL loops? If so, one of us doesn't understand these rules or has them wrong ... and it may be me. Or are you saying that these rules allow PHYSICAL loops only?

> In my opinion, it is a loop if there are more-than-one active bridges between
> two buses in parallel. Maybe, we should come up with a clear definition of 'a loop'.
>
[RLC] Again, I agree with the terminology, as discussed above.

> I believe that one of the main reasons, if not the primary reason, why we want
> to allow 'loops' is to add redundant (I would rather call it alternate or
> supplemental) paths between buses so as to reduce congestion (isochronous or
> asynchronous) on particular bridges. The example I gave you falls into this
> category.

Date: Thu, 09 Jul 1998 14:02:30 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Richard,

I agree with your definition of "loops", and I can live with the term "redundant".

As for the difficulty of network management, however, I am unable to give you a quick answer. Breaking loops would possibly be as complex and burdensome as allowing loops. We all need more time to judge this.

> [RLC] As I understand them, rules 2 (a response from bus Bb to bus Ba will
> take the same -- but reversed -- path as the original request from bus Ba to
> bus Bb) and 2a (requests from bus Ba to bus Bb shall follow the same path as
> responses from bus Ba to bus Bb) taken together prevent logical loops (where
> packets passed between buses Ba and Bb could take more than one path), but
> do not preclude physical loops (there may be physical paths between any pair
> of buses that are not used or may be disabled). So, are you saying that
> these rules combined ALLOW LOGICAL loops? If so, one of us doesn't
> understand these rules or has them wrong ... and it may be me. Or are you
> saying that these rules allow PHYSICAL loops only?

Take a close look at the example I gave.

kind of packet:	bridge(s) it go through
request from A to B:	X
response from A to B:	X
request from B to A:	X

```
response from B to A:        X
request from A  to C:       Y, Z
response from A to C:       Y, Z
request from C to A:        Z, Y
response from C to A:        Z, Y
request from B to C:        Z
response from B to C:        Z
request from C to B:        Z
response from C to B:        Z
```

As you can see, this routing table satisfies both rules #2 AND #2a, however, there exists a LOGICAL LOOP (three bridges for three connected buses, according to your definition), doesn't it?

Regards,

Takashi

Churchill, Richard wrote:
> Topologically speaking, an easy way to say, "There are loops," is to say
> something like, "For a network of n interconnected buses, there are at least
> n bridges attached to separate buses at each end." There is no difference
> between a "redundant path directly between 'adjacent' nodes" and a loop.
> This is related to the question of how you can tell when an individual bus
> has a loop ... For n nodes, you should have n-1 cables connecting them, and
> more than n-1 means you have a loop. Whether there are two bridges directly
> connecting buses Bi and Bj or some other configuration, the additional
> bridge creates a redundant path, and thus a loop, just as inserting an
> additional cable creates a loop in a bus.
>
> I like the term "redundant," and don't see why we would want to say
> "supplemental" or "alternative" instead. Redundancy is a well defined and
> understood technical term, which in technical applications involving quality
> of service, safety, etc., is devoid of any pejorative sense. Calling a loop
> a redundant path or vice versa doesn't bother me at all. They are the same.
>
> I also would like to allow loops for the reasons you cite. However, we need
> to recognize the fact that allowing unrestricted loops (as many additional
> bridges in a network as you want, wherever you want) creates problems in
> routing, and an additional burden in network management -- big problems and
> burdens. I don't see how, during the process of initializing paths in a
> CYCLIC network, we are going to be able to adequately guess how to configure
> the paths to correctly deal with the problems you are trying to solve. An
> acyclic tree is far easier to manage. So, as you are working to resolve the
> problems entailed by loops, keep in mind that we may need to drop support
> for loops (other than loop healing) in favor of finishing this standard in a
> timely manner, as well as to keep costs and complexity low.
>
> Please see further comments embedded below.

Date: Thu, 09 Jul 1998 11:51:35 -0700
To: stds-1394-1@majordomo.ieee.org
From: Dick Scheel <dicks@lsi.sel.sony.com>
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Hi Takashi,

In your example below, are the physical connections like this:?

bridges X & Y both between bus A and bus B bridge Z between bus B and bus C

A comment on this configuration: If the routing follows the paths below, then the routing decisions depend on both source and destination, not just on the destination. Look at a packet appearing on bus B destined for bus A. If it started on bus B, then bridge X forwards it to bus A. If it started on bus C (and arrived on bus B via bridge Z) then bridge Y forwards it to bus A. This is certainly possible to build, but takes more logic and more table space.

Regards,
Dick Scheel

At 02:02 PM 7/9/98 -0400, Takashi Sato wrote:
[snip]
>
>Take a close look at the example I gave.
>
>kind of packet: bridge(s) it go through
>request from A to B: X
>response from A to B: X
>request from B to A: X
>response from B to A: X
>request from A to C: Y, Z
>response from A to C: Y, Z
>request from C to A: Z, Y
>response from C to A: Z, Y
>request from B to C: Z
>response from B to C: Z
>request from C to B: Z
>response from C to B: Z
[snip]

Dick Scheel	Mgr, Tech Stds Development
Sony Electronics	Tel: (408) 982-5834
Technology Standards Office	Fax: (408) 982-5899
2350 Mission College Blvd, Ste. 982	Cel: (408) 307-1696
Santa Clara, CA 95054	email: dicks@lsi.sel.sony.com

Date: Thu, 09 Jul 1998 16:15:01 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Dick,

You are right.
You need to look at the sourceID, too.

Takashi

Dick Scheel wrote:
> In your example below, are the physical connections like this:?
>
> bridges X & Y both between bus A and bus B
> bridge Z between bus B and bus C
>
> A comment on this configuration: If the routing follows the paths below,

> then the routing decisions depend on both source and destination, not just
> on the destination. Look at a packet appearing on bus B destined for bus
> A. If it started on bus B, then bridge X forwards it to bus A. If it
> started on bus C (and arrived on bus B via bridge Z) then bridge Y forwards
> it to bus A. This is certainly possible to build, but takes more logic and
> more table space.

Takashi Sato
Philips Research Tel: 914-945-6099
345 Scarborough Road Fax: 914-945-6580
Briarcliff Manor, NY 10510 E-mail: txs@philabs.research.philips.com

---

From: bradley.saunders@rss.rockwell.com
To: stds-1394-1@majordomo.ieee.org
Date: Thu, 9 Jul 1998 19:38:34 -0700
Subject: RE: [P1394.1]: Voting-Item[2] the same path

I think that restricting all traffic between two given buses to use the same bridge is not good. We should allow for a bridge manager function to load balance between multiple redundant bridge paths as long as it manages not to establish "loops" and maintains symetric paths between any two given nodes (on different buses).

[I think that some of Richard's dissertation was along this line of thought, though I am not sure. :)]

Brad Saunders

"Bard, Steve" <steve.bard@intel.com> on 07/08/98 04:56:30 PM
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path

"Muttering" is a "good" word! I think I said I am OK that busses may create a "loop" when interconnected via bridges IF single paths for response/request were required. Said in "Bard speakease":

"Busses may be interconnected (via bridges) such that a "loop" may be formed. A request (and its corresponding response) shall use a single path (the response will use the same path but in the opposite direction).
Similarly, all requests from Bus A to Bus B shall take the same path as all the requests from Bus B to Bus A."

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Thu, 9 Jul 1998 14:06:52 -0500

Brad,

OK, you think its bad to limit the traffic to a single bridge. Ignoring isoc for a minute (they are channel routed and not destination_ID routed) can you tell me:

a) what do you see as the downside for limiting the traffic between two buses to a single path; and
b) how would you propose that someone figure out how to route through multiple bridges?

I thought Richard's dissertation was supporting only using one path (not parallel/redundant) between buses.

David Wooten

> -----Original Message-----
> From: bradley.saunders@rss.rockwell.com [SMTP:bradley.saunders@rss.rockwell.com]
> Sent: Thursday, July 09, 1998 9:39 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: RE: [P1394.1]: Voting-Item[2] the same path
>
> I think that restricting all traffic between two given buses to use the same bridge is not good. We should allow for a bridge manager function to load balance between multiple redundant bridge paths as long as it manages not to establish "loops" and maintains symetric paths between any two given nodes (on different buses).
>
> [I think that some of Richard's dissertation was along this line of thought, though I am not sure. :)]

From: bradley.saunders@rss.rockwell.com
To: stds-1394-1@majordomo.ieee.org
Date: Thu, 9 Jul 1998 14:05:06 -0700
Subject: RE: [P1394.1]: Voting-Item[2] the same path

David ... I take it all back!

I suppose one could easily argue that only one bridge between any two given buses should be a standard restriction. All that has to be required is that the bridge device be at least capable of handling all traffic between the two buses at the highest common bus speed, e.g. the bridge between a 1394a bus (A) & a 1394b bus (B) must be able to support S400 traffic since S400 is the highest possible common speed. With this speed capability and appropriate buffering for arbitration delays, the bridge should be able to keep up with all possible traffic between the two buses independent of any downstream/upstream bridges that may be attached to those given buses.

If we accept this restriction, if multiple bridges are inserted between any two given buses, we might need to have a detection feature which could disable all but the most capable bridge.

Brad

"Wooten, David" <David.Wooten@compaq.com> on 07/09/98 12:06:52 PM
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path

OK, you think its bad to limit the traffic to a single bridge. Ignoring isoc for a minute (they are channel routed and not destination_ID routed) can you tell me:

a) what do you see as the downside for limiting the traffic between two buses to a single path; and
b) how would you propose that someone figure out how to route through multiple bridges?

I thought Richard's dissertation was supporting only using one path (not parallel/redundant) between buses.

From: "Bard, Steve" <steve.bard@intel.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>

Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Thu, 9 Jul 1998 14:48:38 -0700

Brad,

How do you propose to make certain a bridge portal is always connected to a port on a bus which supports the highest speed that particular bus can support? In other words, how do you prevent a bridge from becoming attached to an S100 only port when there are nodes on the bus that can handle S800 or above?

Also, there is some value in the opportunity to have additional isoch channels through a second bridge connected to the same bus the first bridge is connected to.

Visualize: Bus A is bridged to Bus B via bridge X and Bridge Y. Bus B is also bridged to Bus C via bridge Z. Bus A has a number of isoch players and consumes all isoch channels available in bridge Y. However, additional isoch channels are needed. It would be good to allocate additional isoch channels to Bus A via bridge X. The isoch streams could be used by nodes in Bus C as well as nodes in Bus B.

Best Regards,

Steve Bard

-----Original Message-----
From: bradley.saunders@rss.rockwell.com [mailto:bradley.saunders@rss.rockwell.com]
Sent: Thursday, July 09, 1998 2:05 PM
To: stds-1394-1@majordomo.ieee.org
Subject: RE: [P1394.1]: Voting-Item[2] the same path

David ... I take it all back!

I suppose one could easily argue that only one bridge between any two given buses should be a standard restriction. All that has to be required is that the bridge device be at least capable of handling all traffic between the two buses at the highest common bus speed, e.g. the bridge between a 1394a bus (A) & a 1394b bus (B) must be able to support S400 traffic since S400 is the highest possible common speed. With this speed capability and appropriate buffering for arbitration delays, the bridge should be able to keep up with all possible traffic between the two buses independent of any downstream/upstream bridges that may be attached to those given buses.

If we accept this restriction, if multiple bridges are inserted between any two given buses, we might need to have a detection feature which could disable all but the most capable bridge.

Date: Thu, 09 Jul 1998 18:17:37 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Brad,

PLEASE read this again!

> What you call redundant bridges are not always redundant, but sometimes
> necessary. Don't forget that there are natural limitations to bridge's
> capability. The number of isochronous streams and/or the data rate that one
> bridge can handle can be very limited, depending on its CPU power and its
> internal fabric. Especially, this is very true for wireless bridges, whose
> internal fabric (RF, etc.) is a scarce resource.

I strongly believe it shall be up to the manufacturer to choose bridge's capabilities. If you enforce such a restriction, it will be impossible to make low-cost bridges and/or wireless bridges. I don't think that would be a wise business choice. What we must do is to define a mechanism that allows limited-capability bridges as well as super bridges.

Regards,

    Takashi

bradley.saunders@rss.rockwell.com wrote:
> David ... I take it all back!

> I suppose one could easily argue that only one bridge between any two given
> buses should be a standard restriction. All that has to be required is
> that the bridge device be at least capable of handling all traffic between
> the two buses at the highest common bus speed, e.g. the bridge between a
> 1394a bus (A) & a 1394b bus (B) must be able to support S400 traffic since
> S400 is the highest possible common speed. With this speed capability and
> appropriate buffering for arbitration delays, the bridge should be able to
> keep up with all possible traffic between the two buses independent of any
> downstream/upstream bridges that may be attached to those given buses.
>
> If we accept this restriction, if multiple bridges are inserted between any
> two given buses, we might need to have a detection feature which could
> disable all but the most capable bridge.

From: bradley.saunders@rss.rockwell.com
To: stds-1394-1@majordomo.ieee.org
Date: Thu, 9 Jul 1998 16:39:28 -0700
Subject: RE: [P1394.1]: Voting-Item[2] the same path

Steve & Takashi ... I think that you both see it the way I do. I was just taking David's counterpoint a bit farther to illustrate some points. I voted in favor of allowing multiple bridge connections between buses for purposes of load sharing. Whether or not to allow complete transactions between any two given nodes (on different buses) to be able to use varying paths (in the presence of multiple bridges or virtual loops) is still a debatable issue.

Brad

"Bard, Steve" <steve.bard@intel.com> on 07/09/98 02:48:38 PM
 To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path

How do you propose to make certain a bridge portal is always connected to a port on a bus which supports the highest speed that particular bus can support? In other words, how do you prevent a bridge from becoming attached to an S100 only port when there are nodes on the bus that can handle S800 or above?

Also, there is some value in the opportunity to have additional isoch channels through a second bridge connected to the same bus the first bridge is connected to.

Visualize: Bus A is bridged to Bus B via bridge X and Bridge Y. Bus B is also bridged to Bus C via bridge Z. Bus A has a number of isoch players and consumes all isoch channels available in bridge Y. However, additional isoch channels are needed. It would be good to allocate additional isoch channels to Bus A via bridge X. The isoch streams could be used by nodes in Bus C as well as nodes in Bus B.

July 10

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: Re[2]: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 08:58:18 -0500

Jim,

We have problems to solve if the with return path is not the same as the send path. John Fuller pointed out some in a previous email and then there's the problem of what do you do with a unified write response (the device returns ack_complete instead of ack_pending.) The bridge that is supposed to handle the response was not part of the transaction. Now, we can make it a part of the transaction if we require that it snoop the bus and look for write requests that have a return address that they are supposed to handle. Then, when the ack_complete is seen, it takes responsibility for sending the response. I think that's a lousy thing to do.

So, before we wander too far down the path of alternate response paths, can you explain why you think that alternate paths would speed anything up?

David

> -----Original Message-----
> From: Jim_Busse@ccgate.sj.nec.com [SMTP:Jim_Busse@ccgate.sj.nec.com]
> Sent: Thursday, July 09, 1998 2:04 PM
> To: stds-1394-1@majordomo.ieee.org; Takashi Sato
> Subject: Re[2]: [P1394.1]: Voting-Item[2] the same path
>
> If loops are permitted, and the return path is not necessarily the send path, would it still be possible to specify the path for the send and return in order to optimize speed of transmission and throughput optimization? In some circumstances, I would like to take the fastest path for the quickest possible response.

From: "Churchill, Richard" <Richard.Churchill@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 09:20:45 -0500

Takashi-san,

Yes, loop healing is problematic, too, but not as problematic as is the case for an individual bus. In that circumstance, the reset process never makes it out of Tree ID, making it very difficult to determine the topology so as to optimally break the loops and keep contact with all nodes. You can't get all the information needed without at least seeing the Self-ID packets, though you should be able to come pretty close.

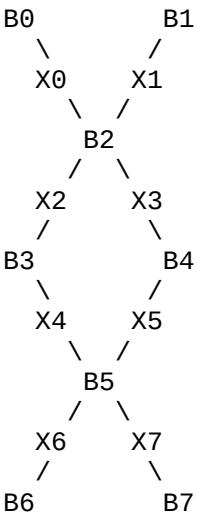
Bridges should be smarter, and the individual buses shouldn't fail to complete resets due to redundant bridges. (OK, so if you attached the same bridge at two points in the same bus you have a problem that I don't want to think about at the moment.) You can get the individual buses up and running, and then the bridges should be able to figure out (via GUIDs, etc.) when connections between adjacent buses are redundant.

As for the presence of loops in the case described, it depends on where you sit. Using both rules 2 and 2a, from any one bus the network seems to be an acyclic tree. Requests

and responses exchanged with any other bus in the network always travel the same acyclic tree-structured paths. From a different bus, the network may look like a different tree, but it still appears acyclic. Viewed externally, we may be able to see loops such as you describe -- physical loops having bridges that are not used in exchanges between specific pairs of buses but are used by one or more other pairs, or which may be disabled and thus are not logically present in the network tree as viewed from any bus.

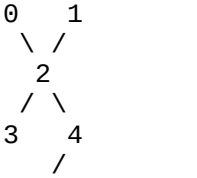
The case you describe has one problem ... The path taken by a packet is dependent upon both the source and destination. A packet originating on bus A can be routed to bus C by a different bridge than packets addressed to bus B, but when a packet originating on either bus B or C reaches bridges X and Y, both will see the destination field contains the ID for A, and pass the packet to bus A, putting two copies of the same packet on A, which is not a good thing ... unless you are also decoding the source ID, which I don't think a good idea either, since that seems to imply a separate routing map for each bus on the network in every bridge.

Consider the following topology instead. (Sorry about the crude text "drawing." If you have a choice, view this using a fixed width font. Buses are denoted by "Bi" and bridges by "Xj.")



If both rules 2 and 2a apply, and we route based only upon the destination ID, a request or response from B0 to B7 must follow the same path from B2 onward as is used for requests or responses from B1 to B7 ... They'll all go either via bus B3 or B4. Let's say they all go via bus B3. Responses from B7 must therefore also back-track the same way, via B4. Now, if we tried to route requests and responses from B0 and B1 to B6 via B3, we would have a problem, since the response from B6 would, upon reaching B5 be routed back to B0 and B1 via B4, and we haven't satisfied our rules. So, all requests and responses with a common destination bus ID passing through two common buses must share a common path between those two common points. (In this case, for packets passing between buses B0/1, and buses B6/7, the physical connection from B2 via B3 to B5 is logically broken in X2, and the return path is logically broken in X4.)

Now let's assume that the path from buses B0/1 to B3 use bridge X2. We know that the logical connection between buses B0/1 and buses B4/5 must follow the same path as those for buses B6/7, so the view of the bus from buses B0, 1 and 2 is the same, as sketched below.



```
  5
 / \
6   7
```

What happens when we view the network from bus B3? It looks like there are four places, bridges X2, 3, 4 and 5, where the physical loop B2-3-4-5 may be broken. With the stipulated three rules, we must accommodate paths that already exist, so breaking the logical loop at X3 gives a topology as viewed from B3 that is different from that described in the previous paragraph, as illustrated below.

```
0   1
 \ /
  2
 /
3   4
 \ /
  5
 / \
6   7
```

So, we have a physical loop, even though we don't have a logical one. The view of the network from buses B5, 6 and 7 must also comply with the stipulated three rules, so we have the following from their perspective.

```
0   1
 \ /
  2
   \
3   4
 \ /
  5
 / \
6   7
```

So, we have three distinct logical views of the bus ... Other structures are possible. This indicates that the three rules allow physical loops but not logical ones.

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Thursday, July 09, 1998 1:03 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> I agree with your definition of "loops", and I can live with the term
> "redundant".
>
> As for the difficulty of network management, however, I am unable to give you a
> quick answer. Breaking loops would possibly be as complex and burdensome as
> allowing loops. We all need more time to judge this.
>
> > [RLC] As I understand them, rules 2 (a response from bus Bb to bus Ba will
> > take the same -- but reversed -- path as the original request from bus Ba to
> > bus Bb) and 2a (requests from bus Ba to bus Bb shall follow the same path as
> > responses from bus Ba to bus Bb) taken together prevent logical loops (where
> > packets passed between buses Ba and Bb could take more than one path), but
> > do not preclude physical loops (there may be physical paths between any pair
> > of buses that are not used or may be disabled). So, are you saying that
> > these rules combined ALLOW LOGICAL loops? If so, one of us doesn't
> > understand these rules or has them wrong ... and it may be me. Or are you
> > saying that these rules allow PHYSICAL loops only?
>

> Take a close look at the example I gave.
>
> kind of packet: bridge(s) it go through
> request from A to B: X
> response from A to B: X
> request from B to A: X
> response from B to A: X
> request from A to C: Y, Z
> response from A to C: Y, Z
> request from C to A: Z, Y
> response from C to A: Z, Y
> request from B to C: Z
> response from B to C: Z
> request from C to B: Z
> response from C to B: Z
>
> As you can see, this routing table satisfies both rules #2 AND #2a, however,
> there exists a LOGICAL LOOP (three bridges for three connected buses, according
> to your definition), doesn't it?

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 09:31:33 -0500

I don't understand the issue about wireless bridges and redundant connections. Is the problem that we may have multiple wireless bridges where we are trying to load-share? Or is it that we have a wireless connection to some things and a copper/glass/plastic connection in parallel?
If the latter, why wouldn't we route everything through the non-wireless connection and only send things into the wireless domain when we need to access a device that can't be accessed any other way?

David Wooten

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Thursday, July 09, 1998 5:18 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> PLEASE read this again!
>
> > What you call redundant bridges are not always redundant, but sometimes
> > necessary. Don't forget that there are natural limitations to bridge's
> > capability. The number of isochronous streams and/or the data rate that one
> > bridge can handle can be very limited, depending on its CPU power and its
> > internal fabric. Especially, this is very true for wireless bridges, whose
> > internal fabric (RF, etc.) is a scarce resource.
>
> I strongly believe it shall be up to the manufacturer to choose bridge's
> capabilities. If you enforce such a restriction, it will be impossible to make
> low-cost bridges and/or wireless bridges. I don't think that would be a wise
> business choice. What we must do is to define a mechanism that allows
> limited-capability bridges as well as super bridges.

Date: Fri, 10 Jul 1998 11:37:30 -0400
From: Takashi Sato <txs@philabs.research.philips.com>

To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

David,

Pardon me if I confused you.

Almost any bridge (wireless or wired) has its limitations to some extent unless it can handle 64 isochronous streams at a total of 400 Mbps.

The redundant bridge argument usually applies to wired bridges.

However, if wireless bridges use different frequency bands or media (e.g., RF and IR), it can apply to wireless bridges, too.

I'm not discussing the situation where wired and wireless bridges are connection the same buses because it won't make much sense.

Regards,

Takashi

Wooten, David wrote:
> I don't understand the issue about wireless bridges and redundant
> connections. Is the problem that we may have multiple wireless bridges
> where we are trying to load-share? Or is it that we have a wireless
> connection to some things and a copper/glass/plastic connection in parallel?
> If the latter, why wouldn't we route everything through the non-wireless
> connection and only send things into the wireless domain when we need to
> access a device that can't be accessed any other way?

Date: Fri, 10 Jul 1998 15:51:39 -0400
From: Takashi Sato <txs@philabs.research.philips.com>
To: stds-1394-1@majordomo.ieee.org
Subject: Re: [P1394.1]: Voting-Item[2] the same path

Richard,

Your example serves better to show that there can be a solution that uses only destinationIDs while satisfying the rules #2 and #2a. Thanks.

I wonder if you can prove that there always exists a solution for any physical network topologies.

I also wonder if somebody can come up with an algorithm that solves these seemingly very complex routing puzzles.

Regards,

Takashi

Churchill, Richard wrote:
> Yes, loop healing is problematic, too, but not as problematic as is the case
> for an individual bus. In that circumstance, the reset process never makes
> it out of Tree ID, making it very difficult to determine the topology so as
> to optimally break the loops and keep contact with all nodes. You can't get
> all the information needed without at least seeing the Self-ID packets,
> though you should be able to come pretty close.
[snip]

Takashi Sato
Philips Research Tel: 914-945-6099
345 Scarborough Road Fax: 914-945-6580
Briarcliff Manor, NY 10510 E-mail: txs@philabs.research.philips.com

---

From: "Wooten, David" <David.Wooten@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 16:35:30 -0500

Richard,

What if there was a table that indicated which nodes on a bus that the bridge would accept requests/responses from? Seems that this could be used to open up the network somewhat. From a bridge perspective, a bus can be divided into multiple segments each with its own bus number. If we do this right, the nodes will not need to know what bus they are on so they would only need to use 3FF as the bus number in the source ID and the bridge would translate. I'm not sure this would give us all the routing capability we might imagine but it might be worth the additional cost.

Now, find some serious problems with this idea so that we can say we considered it an eliminated it due to major breakage.

David Wooten

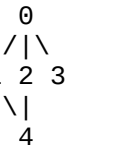
> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Friday, July 10, 1998 2:52 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> Your example serves better to show that there can be a solution that uses only destinationIDs while satisfying the rules #2 and #2a. Thanks.
>
> I wonder if you can prove that there always exists a solution for any physical network topologies.
>
> I also wonder if somebody can come up with an algorithm that solves these seemingly very complex routing puzzles.

From: "Churchill, Richard" <Richard.Churchill@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>
Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 18:05:15 -0500

Takashi-san,

The fact that a network of buses is a connected graph, but not a directed graph (our bridges are bi-directional), means that there is always way of getting from any one bus to any other ... It is a property of connected graphs. The simplest connected graph is an acyclic tree, and all we've done with rules 2 and 2a plus using only destination IDs for routing is to overlay a set of logical acyclic trees on top of a potentially cyclic graph, with certain properties for the logical trees derived from the convergence of paths imposed by the three operative rules.

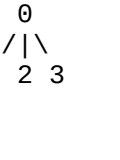
I don't think the problem is to create an acyclic tree from a cyclic network, but to create an optimal one. If you select any bus in the network as a starting point, enumerate the buses in a breadth-first sequence (see the following diagram, best viewed in fixed width fonts), then construct ordered pairs representing each bridge by (x,y), where x < y, eliminate redundant pairs, then construct a first order connectivity matrix, loops will show up as buses with connections to more than one bus of lower ordinal number. To eliminate the loops, just throw out the pairs with higher x ordinal values.



(0,1), (0,2), (0,3), (1,4), (2,4)

	0	1	2	3	4
0	X	1	1	1	0
1	1	X	0	0	1
2	1	0	X	0	1
3	1	0	0	X	0
4	0	1	1	0	X

Eliminate connection (2,4), and we have an acyclic tree.



It will be a little tricky making sure all graphs obey our three rules, if we wish to make use of physical loops, but I think an iterative process can be developed without driving anyone too crazy. But again, making these connections OPTIMAL gives me serious heart-burn.

Sincerely,

Richard Churchill,
(281)514-6984,
richard.churchill@compaq.com

> -----Original Message-----
> From: Takashi Sato [SMTP:txs@philabs.research.philips.com]
> Sent: Friday, July 10, 1998 2:52 PM
> To: stds-1394-1@majordomo.ieee.org
> Subject: Re: [P1394.1]: Voting-Item[2] the same path
>
> Your example serves better to show that there can be a solution that uses only
> destinationIDs while satisfying the rules #2 and #2a. Thanks.
>
> I wonder if you can prove that there always exists a solution for any physical
> network topologies.
>
> I also wonder if somebody can come up with an algorithm that solves these
> seemingly very complex routing puzzles.

From: "Churchill, Richard" <Richard.Churchill@compaq.com>
To: "'stds-1394-1@majordomo.ieee.org'" <stds-1394-1@majordomo.ieee.org>

Subject: RE: [P1394.1]: Voting-Item[2] the same path
Date: Fri, 10 Jul 1998 18:07:37 -0500

Sir:

Well, it's too late in the day for that, so it will have to wait until tomorrow ... or Bath.

Sincerely,

Richard Churchill,
(281)514-6984,
richard.churchill@compaq.com

> -----Original Message-----
> From: Wooten, David
> Sent: Friday, July 10, 1998 4:36 PM
> To: 'stds-1394-1@majordomo.ieee.org'
> Subject: RE: [P1394.1]: Voting-Item[2] the same path
>
> What if there was a table that indicated which nodes on a bus that the
> bridge would accept requests/responses from? Seems that this could be used
> to open up the network somewhat. From a bridge perspective, a bus can be
> divided into multiple segments each with its own bus number. If we do this
> right, the nodes will not need to know what bus they are on so they would
> only need to use 3FF as the bus number in the source ID and the bridge would
> translate. I'm not sure this would give us all the routing capability we
> might imagine but it might be worth the additional cost.
>
> Now, find some serious problems with this idea so that we can say we
> considered it an eliminated it due to major breakage.

July 12

From: PJohansson@aol.com
Date: Sun, 12 Jul 1998 15:49:10 EDT
To: stds-1394-1@ieee.org
Subject: Re: [P1394.1]: Minutes + vote announcement

In a message dated 98-07-02 08:37:47 EDT, saito@optsys1.optsys.cl.nec.co.jp writes:
<<There are no discussion on the reflector about vote as shown below. I understand the voting items of 1) and 2) are essentially whether the loop bridge are allowed or not, as Neil wrote. I believe that the loop bridge has many advantages and many disadvantages. However these are not cleared. I think more discussion is needed, and I feel these voting items are still open issues.>>

I agree with Saito-san's opinion that it is premature to be taking formal votes on these issues. I think concrete, detailed proposals are necessary first.

That said, I'm not suggesting we should slow progress by doing nothing. I think straw polls would be in order, to be followed shortly with technical documents.

Regards,

Peter Johansson

Congruent Software, Inc.
3998 Whittle Avenue
Oakland, CA 94602
(510) 531-5472
(510) 531-2942 FAX
pjohansson@aol.com