Network Working Group
Internet-Draft
<draft-johansson-ip1394-asynch-streams-00.txt>
Expires: May, 1999

P. Johansson
Congruent Software, Inc.

IEEE 1394 Asynchronous Streams

STATUS OF THIS DOCUMENT

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

ABSTRACT

This document proposes modifications to some of the packet formats defined by Internet-Draft draft-ietf-ip1394-ipv4-11, specifically those intended for transport by asynchronous stream packets.

TABLE OF CONTENTS

1. INTRODUCTION

This document proposes modifications to some of the packet formats defined by Internet-Draft draft-ietf-ip1394-ipv4-11, specifically those intended for transport by asynchronous stream packets.

The referenced Internet-Draft defines packet formats for Address Resolution Protocol (ARP) and Multicast Channel Allocation Protocol (MCAP) that utilize a Serial Bus channel for the broadcast of asynchronous stream packets to all IP-capable Serial Bus devices. These formats and methods have been tested and found to be interoperable between implementations of at least two different vendors.

Recent work in the IEEE P1394.1 working group, Serial Bus to Serial Bus bridges, suggests that the current ARP and MCAP formats will not be extensible when bridges are used to form nets of interconnected buses. In fact, the formats specified today will create problems when bridges are introduced.

Although the IETF working group responsible for the IPv4 over IEEE 1394 protocol has chosen to defer exact specification of operations in a bridged environment, the author of this document believes that the anticipated problems are serious enough to force a reexamination of these issues.

2. DEFINITIONS AND NOTATION

2.1 Conformance

When used in this document, the keywords "may", "optional", "recommended", "required", "shall" and "should" differentiate levels of requirements and optionality and are to be interpreted as described in RFC 2119.

Several additional keywords are employed, as follows:

ignored: A keyword that describes bits, octets, quadlets or fields whose values are not checked by the recipient.

reserved: A keyword used to describe objects---bits, octets, quadlets and fields---or the code values assigned to these objects in cases where either the object or the code value is set aside for future standardization. Usage and interpretation may be specified by future extensions to this or other standards. A reserved object shall be zeroed or, upon development of a future standard, set to a value specified by such a standard. The recipient of a reserved object shall not check its value. The recipient of an object defined by this standard as other than reserved shall check its value and reject reserved code values.

2.2 Glossary

The following terms are used in this standard:

address resolution protocol: A method for a requester to determine the hardware (1394) address of an IP node from the IP address of the node.

bus ID: A 10-bit number that uniquely identifies a particular bus within a group of multiple interconnected buses. The bus ID is the most significant portion of a node’s 16-bit node ID. The value 0x3FF designates the local bus; a node shall respond to requests addressed to its 6-bit physical ID if the bus ID in the request is either 0x3FF or the bus ID explicitly assigned to the node.

IP datagram: An Internet message that conforms to the format specified by RFC 791.

node ID: A 16-bit number that uniquely identifies a Serial Bus node within a group of multiple interconnected buses. The most significant 10 bits are the bus ID and the least significant 6 bits are the physical ID.

node unique ID: A 64-bit number that uniquely identifies a node among all the Serial Bus nodes manufactured worldwide; also known as the EUI-64 (Extended Unique Identifier, 64-bits).

octet: Eight bits of data.

packet: Any of the 1394 primary packets; these may be read, write or lock requests (and their responses) or stream data. The term "packet" is used consistently to differentiate 1394 packets from ARP requests/responses, IP datagrams or MCAP advertisements/solicitations.

physical ID: On a particular bus, this 6-bit number is dynamically assigned during the self-identification process and uniquely identifies a node on that bus.

quadlet: Four octets, or 32 bits, of data.

stream packet: A 1394 primary packet with a transaction code of 0x0A that contains a block data payload. Stream packets may be either asynchronous or isochronous according to the type of 1394 arbitration employed.

2.3 Abbreviations

The following are abbreviations that are used in this standard:

ARP	Address resolution protocol
CSR	Control and status register
CRC	Cyclical redundancy checksum
EUI-64	Extended Unique Identifier, 64-bits
IP	Internet protocol (within the context of this document, IPv4)
MCAP	Multicast channel allocation protocol

2.4 Numeric values

Decimal and hexadecimal numbers are used within this standard. By editorial convention, decimal numbers are most frequently used to represent quantities or counts. Addresses are uniformly represented by hexadecimal numbers. Hexadecimal numbers are also used when the value represented has an underlying structure that is more apparent in a hexadecimal format than in a decimal format.

Decimal numbers are represented by Arabic numerals or by their English names. Hexadecimal numbers are prefixed by 0x and represented by digits from the character set 0 – 9 and A – F. For the sake of legibility, hexadecimal numbers are separated into groups of four digits separated by spaces.

For example, both 42 and 0x2A represent the same numeric value.

3. PROBLEM STATEMENT

The problems are most easily understood in the context of ARP as defined by draft-ietf-ip1394-ipv4-11. The draft describes a means by which a Serial Bus channel is allocated by one of the IP-capable devices, the network protocol manager (NPM), which then communicates the channel number to all the IP-capable devices on the bus. When address resolution is necessary, the sender broadcasts an ARP packet on this channel. The pertinent information in the ARP request is:

- the IP address of the sender;

- the Serial Bus address of the sender; and

- the IP address of the device that the sender wishes to locate.

The fundamental challenge for IPv4 over IEEE 1394 is the fact that the 16-bit Serial Bus addresses (the node ID) are mutable with each bus reset. The insertion or removal of a Serial Bus device forces a bus reset; software may generate bus reset(s) at its discretion. Bus resets on one's local bus are manageable even if inconvenient; bus resets on some remote bus that lies beyond one or more bridges are particularly difficult. The only solution found by the P1394.1 working group to the management of remote node IDs is to virtualize them. Remote nodes are addressed by a *virtual node ID* that is intended to remain stable across bus resets.

NOTE: The reader is referred to a P1394.1 working document, BR034R00.pdf, which may be downloaded from the web site maintained by the working group at http://grouper.ieee.org/groups/1394/1/index.html. The detailed description of the virtual node ID concept is not repeated in this document.

The use of virtual node IDs places some demands on both the bridges and the data formats of the packets they route. The bridge immediately adjacent to a sender transforms the packet's *source_ID* from a physical (changeable) node ID to a virtual (stable) node ID. The bridge immediately adjacent to a recipient transforms the packet’s *destination_ID* from a virtual node ID to a physical node ID recognizable to the recipient. In order for bridges to perform these mappings, the packet format must be well known and the locations of *source_ID* and *destination_ID* fixed. This is the case for all asynchronous primary packets defined by IEEE Std 1394-1995.

The ARP request and response packets are carried by asynchronous stream packets, a P1394a extension of the isochronous packets defined by IEEE Std 1394-1995. The stream packet header contains neither *source_ID* nor *destination_ID*; consequently, a bridge is unable to apply any virtual node ID mapping to a stream packet as presently defined.

Is there any way that IP-capable nodes can embed enough information in an ARP packet for it to be useful on another, remote bus without any transformation by a bus? I think not and set forth the argument below.

Let us assume that IP-capable devices are split into two groups, those that are unaware of the existence of P1394.1 bridges and those that are cognizant of bridges. Let us further assume that the NODE_IDS register is *not* the method used by bridges to communicate unique bus IDs to Serial Bus devices.[1] As a result, devices unaware of bridges (let’s call them legacy devices) make use of bus ID 0x3FF, only. In order for an ARP request to be useful on the local bus of origination it must identify the sender by bus ID 0x3FF or else the legacy recipients will ignore the request. But, this same ARP request, when it is transported across a bridge to a remote bus, will be recognized by legacy devices on the remote bus---and recognized inappropriately! Unless there is some way for a bridge to either modify the ARP request or quarantine it to its local bus of origin, malfunctions will occur.

The first possibility, in which the bridge modifies the ARP request, is discussed in the next section. The other possibility, quarantine, is more drastic and should be avoided. The problems with quarantine are twofold. First, because there is nothing unique about the stream packets that IPv4 over IEEE 1394 uses to carry ARP, bridges would be forced to quarantine all stream packets that have an unknown format. This would have a chilling effect on the use of asynchronous streams by other protocols that, perhaps, carry no node addressing information. The second reason to avoid quarantine may be more important to IP-capable devices: ARP is guaranteed to be nonextensible to a bridged environment. If the IETF working group adopts the current ARP solution described in draft-ietf-ip1394-ipv4-11, a new ARP solution will be required for bridges and IP-capable devices will have to issue two sets of ARP requests, one for their local bus and one for all remote buses.

[1] Similar arguments may be presented even if NODE_IDS is used and programmed to contain the unique bus IDs, but they are more complicated.

4. GLOBAL ASYNCHRONOUS STREAM PACKET FORMAT

That’s right, GASP! The P1394.1 working group proposes to standardize a new format for asynchronous stream packets that may be recognized by bridges and transformed by bridges as the stream packets flow from bus to bus. The proposed format takes the first two quadlets of the data payload and uses them as an extension to the stream packet header information, as shown below:

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          data_length          |tag|  channel  |  0x0A |   sy  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           header_CRC                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           source_ID           |        specifier_ID_hi        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |specifier_ID_lo|                    version                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +---                           data                          ---+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            data_CRC                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5 - GASP format

The definition and usage of the new fields not already specified by IEEE Std 1394-1995 and P1394a is as follows:

The *tag* field shall have a value of TO BE DETERMINED; this uniquely identifies the GASP format to P1394.1 bridges.

The *source_ID* field shall specify the address of the sender of the stream packet. When a GASP packet is received on a remote bus, the *source_ID* field shall continue a virtual ID placed in the packet by the first P1394.1 bridge to retransmit the packet.

The *specifier_ID* field shall contain a 24-bit organizationally unique identifier (OUI) assigned by the IEEE RAC. The owner of the OUI (company, accredited standards organization or industry group) shall be responsible to define the meaning and usage of the remainder of the data payload in the stream packet.

The meaning and usage of the *version* field shall be defined by the owner of *specifier_ID*.

Is this format sufficient to shield the IPv4 over IEEE 1394 work from any future changes in the P1394.1 draft standard? After all, the bridging work is still in progress and may not be fully stable for some time to come.

I believe that this format is both necessary and sufficient. The arguments as to its necessity were set out in the preceding section. Why is this format sufficient?

Assume for the moment that the progress made by P1394.1 with virtual node IDs is not the ultimate, stable solution. By necessity, the working group has to invent a solution for the routing of asynchronous request and response packets based on the information available in the headers of asynchronous primary packets. Of all of the fields in asynchronous packet headers, only *source_ID* and *destination_ID* have anything to do with the problems of routing through bridges. If a bridge solution is possible at all, it must work with the information available within these fields.

Now consider that asynchronous stream packets are broadcast packets; this is why they are used for ARP, MCAP and even DHCP. In what way is an asynchronous stream packet different from a conventional asynchronous broadcast write, other than in their transaction codes? In a broadcast write, the *destination_ID* is 0xFFFF. This is not a real address; it is only a token that indicates "broadcast". Asynchronous stream packets are already understood to be broadcast by virtue of their *tcode*; the lack of a *destination_ID* field in an asynchronous stream packet is of no importance. On the other hand, the *source_ID* in a broadcast write request is critical: it permits the recipient to address a subsequent transaction to the originator of the broadcast write. P1394.1 must design a bridge solution that works for broadcast writes and it must invent it out of the materials at hand (*source_ID*). As a consequence, if *source_ID* is added to asynchronous stream packets in a known location, the same solution that works for broadcast writes must work for asynchronous streams.

Both the P1394.1 and IETF working groups may confidently adopt the GASP format as a method to safely transport asynchronous stream data across bridges. We know it has to work at least as well as the solution for broadcast write.

5. ADDRESS RESOLUTION PROTOCOL (ARP)

ARP requests shall be transmitted by the same means as broadcast IP datagrams; ARP responses may be transmitted in the same way or they may be transmitted as block write requests addressed to the *sender_unicast_FIFO* address identified by the ARP request. An ARP request/response is 36 octets and shall conform to the format illustrated below.

NOTE: Just as draft-ietf-ip1394-ipv4-11 omits the link encapsulation format, it should be understood that the diagram below omits the two-quadlet GASP header which precedes the ARP request when it is broadcast as an synchronous stream packet.

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    hardware_type (0x0018)     |    protocol_type (0x0800)     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  hw_addr_len  |  IP_addr_len  |            opcode             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +---                    sender_unique_ID                     ---+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | sender_max_rec|     sspd      |    sender_unicast_FIFO_hi     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    sender_unicast_FIFO_lo                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       sender_IP_address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       target_IP_address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5 - ARP request/response format

Field usage in an ARP request/response is as follows:

hardware_type: This field indicates 1394 and shall have a value of 0x0018.

protocol_type: This field shall have a value of 0x0800; this indicates that the protocol addresses in the ARP request/response conform to the format for IP addresses.

hw_addr_len: This field indicates the size, in octets, of the 1394-dependent hardware address associated with an IP address and shall have a value of 16.

IP_addr_len: This field indicates the size, in octets, of an IP version 4 (IPv4) address and shall have a value of 4.

opcode: This field shall be one to indicate an ARP request and two to indicate an ARP response.

sender_unique_ID: This field shall contain the node unique ID of the sender and shall be equal to that specified in the sender’s bus information block.

sender_max_rec: This field shall be equal to the value of *max_rec* in the sender’s configuration ROM bus information block.

sspd: This field shall be set to the lesser of the sender’s link speed and PHY speed. The link speed is the maximum speed at which the link may send or receive packets; the PHY speed is the maximum speed at which the PHY may send, receive or repeat packets. The encoding used for *sspd* is specified by the table below; all values not specified are reserved.

Value	Speed
0	S100
1	S200
2	S400
3	S800
4	S1600
5	S3200

sender_unicast_FIFO_hi and *sender_unicast_FIFO_lo*: These fields together shall specify the 48-bit offset of the sender’s FIFO available for the receipt of IP datagrams in the format specified by draft-ietf-ip1394-ipv4-11. The offset of a sender’s unicast FIFO shall not change, except as the result of a power reset.

sender_IP_address: This field shall specify the IP address of the sender.

target_IP_address: In an ARP request, this field shall specify the IP address from which the sender desires a response. In an ARP response, it shall be ignored.

Note that *sender_node_ID* has been removed from the ARP format; the recipient of an ARP packet shall obtain this information from the *source_ID* field in either and asynchronous primary packet or a GASP packet. Also note that the *sender_xxx* fields hold the requester’s or responder’s information according to *opcode* (ARP request or response). In both cases the information in these fields pertains to the transmitter of the packet. The *target_IP_address* is meaningful only in an ARP request; the IP address of the responder is contained in *sender_IP_address* when *opcode* indicates an ARP response.

6. SECURITY CONSIDERATIONS

This document pertains to the use of an unsecured link layer, Serial Bus, for the transport of IPv4 datagrams. Serial Bus is vulnerable to denial of service attacks; it is also possible for devices to eavesdrop on data or present forged identities. Implementers who utilize Serial Bus for IPv4 should consider appropriate counter-measures within application or other layers.

7. ACKNOWLEDGEMENTS

This document builds upon work in progress by the IP/1394 Working Group. The author wishes to acknowledge his debt to all the effort of others that preceded this proposal.

8. REFERENCES

[1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus

[2] ISO/IEC 13213:1994, Control and Status Register (CSR) Architecture for Microcomputer Buses

[3] IEEE Project P1394a, Draft Standard for a High Performance Serial Bus (Supplement)

[4] draft-ietf-ip1394-ipv4-11, IPv4 over IEEE 1394

9. AUTHOR’S ADDRESS

Peter Johansson
Congruent Software, Inc.
98 Colorado Avenue
Berkeley, CA 94602

(510) 527-3926
(510) 527-3856 FAX
pjohansson@aol.com