                          IPv4 over IEEE 1394

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

ABSTRACT

   This document specifies how to use IEEE Std 1394-1995, Standard for a
   High Performance Serial Bus (and its supplements), for the transport
   of Internet Protocol Version 4 (IPv4) datagrams; it defines the
   necessary methods, data structures and codes for that purpose. These
   include not only packet formats and encapsulation methods for
   datagrams, but also an address resolution protocol (1394 ARP) and a
   multicast channel allocation protocol (MCAP). Both 1394 ARP and MCAP
   are specific to Serial Bus; the latter permits management of Serial
   Bus resources when used by IP multicast groups.

TABLE OF CONTENTS

1. INTRODUCTION

   This document specifies how to use IEEE Std 1394-1995, Standard for a
   High Performance Serial Bus (and its supplements), for the transport
   of Internet Protocol Version 4 (IPv4) datagrams. It defines the
   necessary methods, data structures and codes for that purpose and
   additionally defines methods for an address resolution protocol (1394
   ARP) and a multicast channel allocation protocol (MCAP)---both of
   which are specific to Serial Bus.

   The group of IEEE standards and supplements, draft or approved,
   related to IEEE Std 1394-1995 is hereafter referred to either as 1394
   or as Serial Bus.

   1394 is an interconnect (bus) that conforms to the CSR architecture,
   ISO/IEC 13213:1994. Serial Bus permits communications between nodes
   over shared physical media at speeds that range, at present, from 100
   to 400 Mbps. Both consumer electronic applications (such as digital
   VCRs, stereo systems, televisions and camcorders) and traditional
   desktop computer applications (e.g., mass storage, printers and
   tapes), have adopted 1394. Serial Bus is unique in its relevance to
   both consumer electronic and computer domains and is EXPECTED to form
   the basis of a home or small office network that combines both types
   of devices.

The CSR architecture describes a memory-mapped address space that
Serial Bus implements as a 64-bit fixed addressing scheme. Within the
address space, ten bits are allocated for bus ID (up to a maximum of
1,023 buses), six are allocated for node physical ID (up to 63 per
bus) while the remaining 48 bits (offset) describe a per node address
space of 256 terabytes. The CSR architecture, by convention, splits a
node's address space into two regions with different behavioral
characteristics. The lower portion, up to but not including 0xFFFF
F000 0000, is EXPECTED to behave as memory in response to read and
write transactions. The upper portion is more like a traditional IO
space: read and write transactions in this area usually have side
effects. Control and status registers (CSRs) that have FIFO behavior
customarily are implemented in this region.

Within the 64-bit address, the 16-bit node ID (bus ID and physical
ID) is analogous to a network hardware address---but 1394 node IDs
are variable and subject to reassignment each time one or more nodes
are added to or removed from the bus.

NOTE: Although the 16-bit node ID contains a bus ID, at present there
is no standard method to connect separately enumerated Serial Buses.
Active development of a standard for Serial Bus to Serial Bus bridges
is underway in the IEEE P1394.1 working group. Unless extended by
some future standard, the IPv4 over 1394 protocols specified by this
document may not operate correctly across bridges.

The 1394 link layer provides a packet delivery service with both
confirmed (acknowledged) and unconfirmed packets. Two levels of
service are available: "asynchronous" packets are sent on a best-
effort basis while "isochronous" packets are guaranteed to be
delivered with bounded latency. Confirmed packets are always
asynchronous but unconfirmed packets may be either asynchronous or
isochronous. Data payloads vary with implementations and may range
from one octet up to a maximum determined by the transmission speed
(at 100 Mbps, named S100, the maximum asynchronous data payload is
512 octets while at S400 it is 2048 octets).

NOTE: Extensions underway in IEEE P1394b contemplate additional
speeds of 800, 1600 and 3200 Mbps.

2. DEFINITIONS AND NOTATION

2.1 Conformance

   When used in this document, the keywords "MAY", "OPTIONAL",
   "RECOMMENDED", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD" and "SHOULD
   NOT" differentiate levels of requirements and optionality and are to
   be interpreted as described in RFC 2119.

   Several additional keywords are employed, as follows:

   EXPECTED: A keyword used to describe the behavior of the hardware or
   software in the design models assumed by this standard. Other
   hardware and software design models may also be implemented.

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

   RESERVED: A keyword used to describe either objects---bits, octets,
   quadlets and fields---or the code values assigned to these objects;
   the object or the code value is set aside for future standardization.
   A RESERVED object has no defined meaning and SHALL be zeroed by its
   originator 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 whose code
   values are defined by this standard 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.

   encapsulation header: A structure that precedes all IP data
   transmitted over 1394. See also link fragment.

   IP datagram: An Internet message that conforms to the format
   specified by STD 5, RFC 791.

link fragment: A portion of an IP datagram transmitted within a single 1394 packet. The data payload of the 1394 packet contains both an encapsulation header and its associated link fragment. It is possible to transmit datagrams without link fragmentation.

multicast channel allocation protocol: A method for multicast groups to coordinate their use of Serial Bus resources (channels) if multicast datagrams are transmitted on other than the default broadcast channel.

multicast channel owner: A multicast source that has allocated a channel for one or more multicast addresses and transmits MCAP advertisements to communicate these channel mapping(s) to other participants in the IP multicast group. When more than one source transmits MCAP advertisements for the same channel number, the source with the largest physical ID is the owner.

node ID: A 16-bit number that uniquely identifies a Serial Bus node within a group of multiple interconnected buses. The most significant ten bits are the bus ID and the least significant six 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 Serial Bus primary packets from 1394 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:

      1394 ARP Address resolution protocol (specific to 1394)
      CSR   Control and status register
      CRC   Cyclical redundancy checksum
      EUI-64   Extended Unique Identifier, 64-bits
      GASP   Global asynchronous stream packet
      IP   Internet protocol (within 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, which 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. IP-CAPABLE NODES

   Not all Serial Bus devices are capable of the reception and
   transmission of 1394 ARP requests/responses or IP datagrams. An IP-
   capable node SHALL fulfill the following minimum requirements:

   - it SHALL implement configuration ROM in the general format
     specified by ISO/IEC 13213:1994 and SHALL implement the bus
     information block specified by IEEE P1394a and a unit directory
     specified by this standard;

   - the max_rec field in its bus information block SHALL be at least 8;
     this indicates an ability to accept block write requests and
     asynchronous stream packets with data payload of 512 octets. The
     same ability SHALL also apply to read requests; that is, the node
     SHALL be able to transmit a block response packet with a data
     payload of 512 octets;

   - it SHALL be isochronous resource manager capable, as specified by
     IEEE P1394a;

   - it SHALL support both reception and transmission of asynchronous
     streams as specified by IEEE P1394a; and

4. LINK ENCAPSULATION AND FRAGMENTATION

   All IP datagrams (broadcast, unicast or multicast), 1394 ARP
   requests/responses and MCAP advertisements/solicitations that are
   transferred via 1394 block write requests or stream packets SHALL be
   encapsulated within the packet's data payload. The maximum size of
   data payload, in octets, is constrained by the speed at which the
   packet is transmitted.

                  Table 1 - Maximum data payloads (octets)

                  Speed   Asynchronous   Isochronous
                  +----------------------------------+
                  | S100 | 512 | 1024 |
                  | S200 | 1024 | 2048 |
                  | S400 | 2048 | 4096 |
                  | S800 | 4096 | 8192 |
                  | S1600 | 8192 | 16384 |
                  | S3200 | 16384 | 32768 |
                  +----------------------------------+

   NOTE: The maximum data payloads at speeds of S800 and faster may be
   reduced (but will not be increased) as a result of standardization by
   IEEE P1394b.

   The maximum data payload for asynchronous requests and responses may
   also be restricted by the capabilities of the sending or receiving
   node(s); this is specified by max_rec in either the bus information
   block or 1394 ARP response.

   For either of these reasons, the maximum data payload transmissible
   between IP-capable nodes may be less than the default 1500 octet
   maximum transmission unit (MTU) specified by this document. This
   requires that the encapsulation format also permit 1394 link-level
   fragmentation and reassembly of IP datagrams.

   NOTE: IP-capable nodes may operate with an MTU size larger than the
   default, but the means by which a larger MTU is configured are beyond
   the scope of this document.

4.1 Global asynchronous stream packet (GASP) format

   Some IP datagrams, as well as 1394 ARP requests and responses, may be
   transported via asynchronous stream packets. When asynchronous stream
   packets are used, their format SHALL conform to the global
   asynchronous stream packet (GASP) format specified by IEEE P1394a.
   The GASP format illustrated below is INFORMATIVE and reproduced for
   ease of reference, only.

```
                        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                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
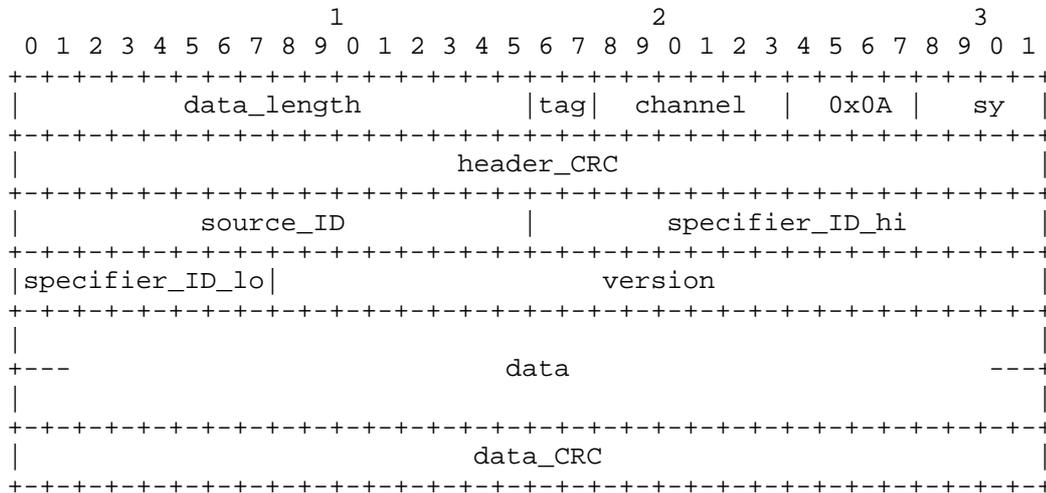
                        Figure 1 - GASP format

   The source_ID field SHALL specify the node ID of the sending node and
   SHALL be equal to the most significant 16 bits of the sender's
   NODE_IDS register.

   The specifier_ID_hi and specifier_ID_lo fields together SHALL contain
   the value 0x00 005E, the 24-bit organizationally unique identifier
   (OUI) assigned by the IEEE Registration Authority (RA) to IANA.

   The version field SHALL be one.

   NOTE: Because the GASP format utilizes the first two quadlets of data
   payload in an asynchronous stream packet format, the maximum payloads
   cited in Table 1 are effectively reduced by eight octets. In the
   clauses that follow, references to the first quadlet of data payload
   mean the first quadlet usable for an IP datagram or 1394 ARP request
   or response. When the GASP format is used, this is the third quadlet
   of the data payload for the packet.

4.2 Encapsulation header

   All IP datagrams transported over 1394 are prefixed by an
   encapsulation header with one of the formats illustrated below.

   If an entire IP datagram may be transmitted within a single 1394
   packet, it is unfragmented and the first quadlet of the data payload
   SHALL conform to the format illustrated 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | lf|         reserved          |          ether_type           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
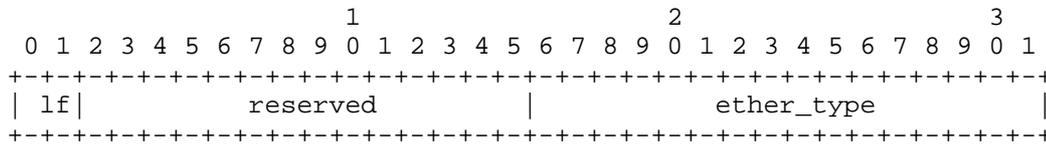
         Figure 2 - Unfragmented encapsulation header format

   The lf field SHALL be zero.

   The ether_type field SHALL indicate the nature of the datagram that
   follows, as specified by the following table.

```
            ether_type   Datagram
          +--------------------------+
          |   0x0800    |  IPv4     |
          |   0x0806    |  1394 ARP |
          |   0x8861    |  MCAP     |
          +--------------------------+
```

   NOTE: Other network protocols, identified by different values of
   ether_type, may use the encapsulation formats defined herein but such
   use is outside of the scope of this document.

   In cases where the length of the datagram exceeds the maximum data
   payload supported by the sender and all recipients, the datagram
   SHALL be broken into link fragments; the first two quadlets of the
   data payload for the first link fragment SHALL conform to the format
   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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | lf|rsv|    datagram_size      |          ether_type           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |            dgl               |           reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
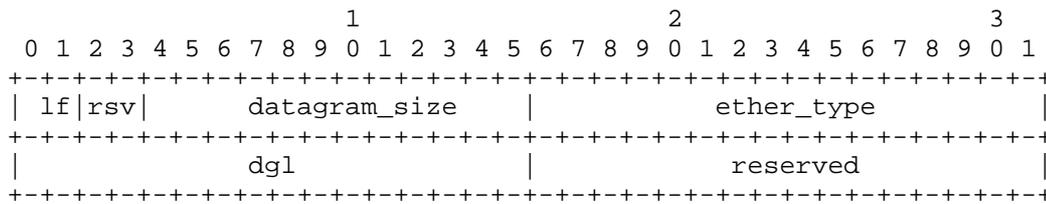
         Figure 3 - First fragment encapsulation header format

The second and subsequent link fragments (up to and including the
last) SHALL conform to the format 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | lf|rsv|      datagram_size    | rsv |     fragment_offset     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |            dgl               |             reserved           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
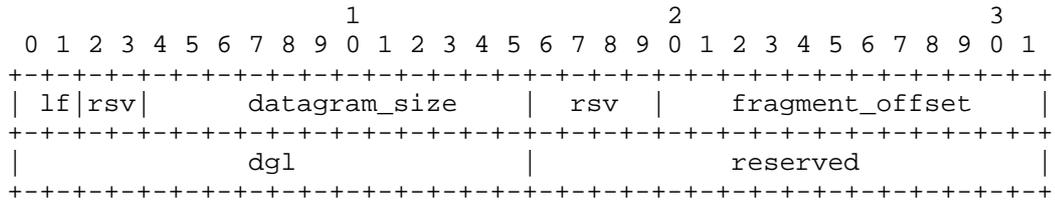
  Figure 4 - Subsequent fragment(s) encapsulation header format

The definition and usage of the fields is as follows:

   The lf field SHALL specify the relative position of the link
   fragment within the IP datagram, as encoded by the following
   table.

```
                    lf        Position
                 +-----------------------+
                 |   0    |  Unfragmented |
                 |   1    |  First        |
                 |   2    |  Last         |
                 |   3    |  Interior     |
                 +-----------------------+
```

   datagram_size: The encoded size of the entire IP datagram. The
   value of datagram_size SHALL be the same for all link fragments of
   an IP datagram and SHALL be one less than the value of Total
   Length in the datagram's IP header (see STD 5, RFC 791).

   ether_type: This field is present only in the first link fragment
   and SHALL have a value of 0x0800, which indicates an IPv4
   datagram.

   fragment_offset: This field is present only in the second and
   subsequent link fragments and SHALL specify the offset, in octets,
   of the fragment from the beginning of the IP datagram. The first
   octet of the datagram (the start of the IP header) has an offset
   of zero; the implicit value of fragment_offset in the first link
   fragment is zero.

   dgl: The value of dgl (datagram label) SHALL be the same for all
   link fragments of an IP datagram. The sender SHALL increment dgl
   for successive, fragmented datagrams; the incremented value of dgl
   SHALL wrap from 65,535 back to zero.

   All IP datagrams, regardless of the mode of transmission (block write
   requests or stream packets) SHALL be preceded by one of the above
   described encapsulation headers. This permits uniform software
   treatment of datagrams without regard to the mode of their
   transmission.

4.3 Link fragment reassembly

   The recipient of an IP datagram transmitted via more than one 1394
   packet SHALL use both the sender's source_ID (obtained from either
   the asynchronous packet header or the GASP header) and dgl to
   identify all the link fragments from a single datagram.

   Upon receipt of a link fragment, the recipient may place the data
   payload (absent the encapsulation header) within an IP datagram
   reassembly buffer at the location specified by fragment_offset. The
   size of the reassembly buffer may be determined from datagram_size.

   If a link fragment is received that overlaps another fragment
   identified by the same source_ID and dgl, the fragment(s) already
   accumulated in the reassembly buffer SHALL be discarded. A fresh
   reassembly may be commenced with the most recently received link
   fragment. Fragment overlap is determined by the combination of
   fragment_offset from the encapsulation header and data_length from
   the 1394 packet header.

   Upon detection of a Serial Bus reset, recipient(s) SHALL discard all
   link fragments of all partially reassembled IP datagrams and
   sender(s) SHALL discard all not yet transmitted link fragments of all
   partially transmitted IP datagrams.

5. SERIAL BUS ADDRESS RESOLUTION PROTOCOL (1394 ARP)

   Methods to determine the hardware address of a device from its
   corresponding IP address are inextricably tied to the transport
   medium utilized by the device. In the description below and
   throughout this document, the acronym 1394 ARP pertains solely to an
   address resolution protocol whose methods and data structures are
   specific to 1394.

   1394 ARP requests SHALL be transmitted by the same means as broadcast
   IP datagrams; 1394 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 1394 ARP request. A
1394 ARP request/response is 32 octets and SHALL conform to the
format illustrated by Figure 5.

```
                      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                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
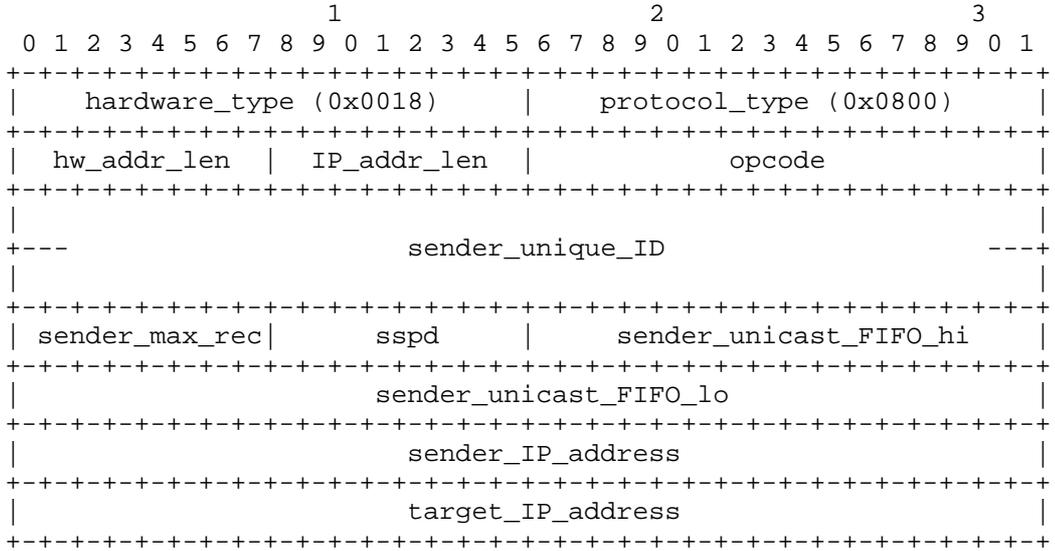
             Figure 5 - 1394 ARP request/response format

1394 ARP requests and responses transported by asynchronous stream
packets SHALL be encapsulated within the GASP format specified by
IEEE P1394a (see also 4.1). The recipient of a 1394 ARP request or
response SHALL ignore it unless the most significant ten bits of the
source_ID field (whether obtained from the GASP header of an
asynchronous stream packet or the packet header of a block write
request) are equal to either 0x3FF or the most significant ten bits
of the recipient's NODE_IDS register.

Field usage in a 1394 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 1394 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 a 1394 ARP request and
two to indicate a 1394 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
table below specifies the encoding used for sspd; all values not
specified are RESERVED for future standardization.

                      Table 2 - Speed codes

                      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 section 6. 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 a 1394 ARP request, this field SHALL specify
the IP address from which the sender desires a response. In a 1394
ARP response, it SHALL be IGNORED.

6. CONFIGURATION ROM

   Configuration ROM for IP-capable nodes SHALL contain a unit directory
   in the format specified by this standard. The unit directory SHALL
   contain Unit_Spec_ID and Unit_SW_Version entries, as specified by
   ISO/IEC 13213:1994.

   The unit directory may also contain other entries permitted by
   ISO/IEC 13213:1994 or IEEE P1212r.

6.1 Unit_Spec_ID entry

   The Unit_Spec_ID entry is an immediate entry in the unit directory
   that specifies the organization responsible for the architectural
   definition of the Internet Protocol capabilities of the device.

```
                      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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      0x12     |            unit_spec_ID (0x00 005E)           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
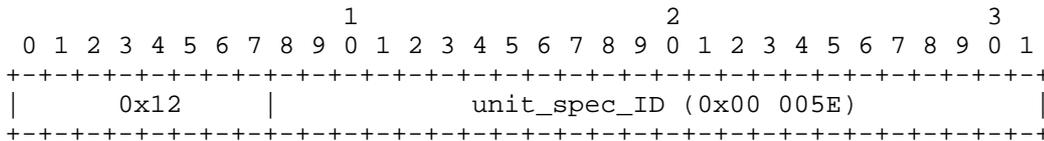
                   Figure 6 - Unit_Spec_ID entry format

   The value of unit_spec_ID SHALL be 0x00 005E, the registration ID
   (RID) obtained by IANA from the IEEE RA. The value indicates that the
   IETF and its technical committees are responsible for the maintenance
   of this standard.

6.2 Unit_SW_Version entry

   The Unit_SW_Version entry is an immediate entry in the unit directory
   that, in combination with the unit_spec_ID, specifies the document
   that defines the software interface of the unit.

```
                      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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      0x13     |          unit_sw_version (0x00 0001)          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
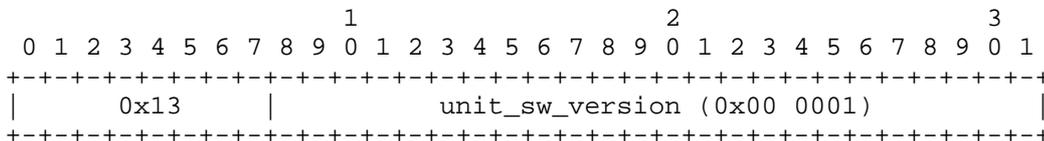
                  Figure 7 - Unit_SW_Version entry format

   The value of unit_sw_version SHALL be one, which indicates that the
   device complies with the normative requirements of this standard.

6.3 Textual descriptors

   Textual descriptors within configuration ROM are OPTIONAL; when
   present they provide additional descriptive information intended to
   be intelligible to a human user. IP-capable nodes SHOULD associate a
   textual descriptor with a content of "IANA" with the Unit_Spec_ID
   entry and a textual descriptor with a content of "IPv4" for the
   Unit_SW_Version entry.

   The figure below illustrates a unit directory implemented by an IP-
   capable node; it includes OPTIONAL textual descriptors. Although the
   textual descriptor leaves are not part of the unit directory, for the
   sake of simplicity they are shown immediately following the unit
   directory.

```
                        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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      directory_length (4)     |              CRC              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x12      |          unit_spec_ID (0x00 005E)            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x81      |          textual descriptor offset (3)       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x13      |              unit_sw_version                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x81      |          textual descriptor offset (5)       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | textual_descriptor_length (3) |              CRC              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +---                         zeros                           ---+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      "I"      |      "A"      |      "N"      |      "A"       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | textual_descriptor_length (3) |              CRC              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +---                         zeros                           ---+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      "I"      |      "P"      |      "v"      |      "4"       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
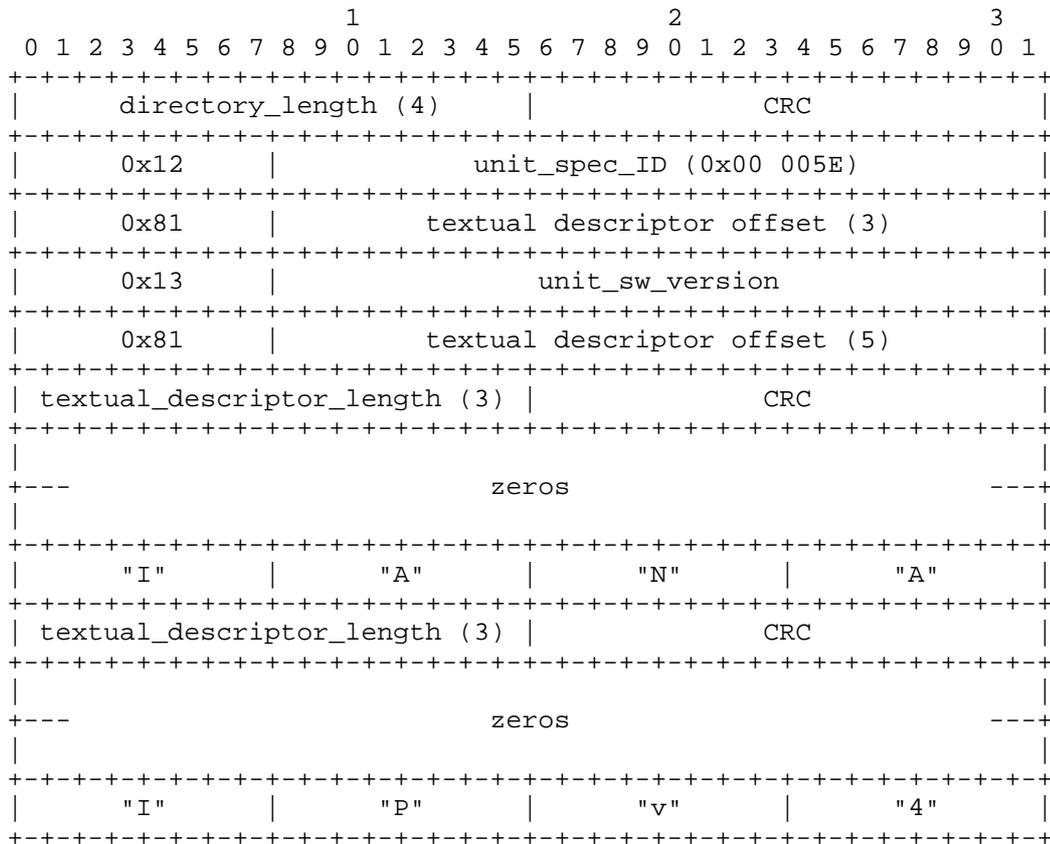
        Figure 9 - Sample unit directory and textual descriptors

7. IP UNICAST

   A unicast IP datagram may be transmitted to a recipient within a 1394
   primary packet that has one of the following transaction codes:

```
            tcode   Description      Arbitration
          +------------------------------------+
          |  0x01 | Block write    | Asynchronous |
          |  0x0A | Stream packet  | Isochronous  |
          |  0x0A | Stream packet  | Asynchronous |
          +------------------------------------+
```

   Block write requests are suitable when 1394 link-level
   acknowledgement is desired but there is no need for bounded latency
   in the delivery of the packet (quality of service).

   Isochronous stream packets provide quality of service guarantees but
   no 1394 link-level acknowledgement.

   The last method, asynchronous stream packets, is mentioned only for
   the sake of completeness. This method SHOULD NOT be used for IP
   unicast, since it provides for neither 1394 link-level acknowledgment
   nor quality of service---and consumes a valuable resource, a channel
   number.

   Regardless of the IP unicast method employed, asynchronous or
   isochronous, it is the responsibility of the sender of a unicast IP
   datagram to determine the maximum data payload that may be used in
   each packet. The necessary information may be obtained from:

   - the SPEED_MAP maintained by the 1394 bus manager, which provides
     the maximum transmission speed between any two nodes on the local
     Serial Bus. The bus manager analyzes bus topology in order to
     construct the speed map; the maximum transmission speed between
     nodes reflects the capabilities of the intervening nodes. The speed
     in turn implies a maximum data payload (see Table 1);

   - the sender_max_rec field in a 1394 ARP response; or

   - other methods beyond the scope of this standard.

   The maximum data payload SHALL be the minimum of the largest data
   payload implemented by the sender, the recipient and the PHYs of all
   intervening nodes (the last is implicit in the SPEED_MAP entry
   indexed by sender and recipient).

NOTE: The SPEED_MAP is derived from the self-ID packets transmitted
by all 1394 nodes subsequent to a bus reset. An IP-capable node may
observe the self-ID packets directly.

Unicast IP datagrams whose quality of service is best-effort SHALL be
contained within the data payload of 1394 block write transactions
addressed to the source_ID and sender_unicast_FIFO obtained from a
1394 ARP response.

If no acknowledgement is received in response to a unicast block
write request it is uncertain whether or not the data payload was
received by the target.

NOTE: An acknowledgment may be absent because the target is no longer
functional, may not have received the packet because of a header CRC
error or may have received the packet successfully but the
acknowledge sent in response was corrupted.

Unicast IP datagrams that require quality of service other than
best-effort are beyond the scope of this standard.

8. IP BROADCAST

Broadcast IP datagrams are encapsulated according to the
specifications of section 4 and are transported by asynchronous
stream packets. There is no quality of service provision for IP
broadcast over 1394. The channel number used for IP broadcast is
specified by the BROADCAST_CHANNEL register.

All broadcast IP datagrams SHALL use asynchronous stream packets
whose channel number is equal to the channel field from the
BROADCAST_CHANNEL register.

Although 1394 permits the use of previously allocated channel
number(s) for up to one second subsequent to a bus reset, IP-capable
nodes SHALL NOT transmit asynchronous stream packets at any time the
valid bit in their BROADCAST_CHANNEL register is zero. Since the
valid bit is automatically cleared to zero by a bus reset, this
prohibits the use of 1394 ARP or broadcast IP until the IRM allocates
a channel number.

9. IP MULTICAST

Multicast IP datagrams are encapsulated according to the
specifications of section 4 and are transported by stream packets.
Asynchronous streams are used for best-effort IP multicast; quality
of service other than best-effort is beyond the scope of this
standard.

By default, all best-effort IP multicast SHALL use asynchronous
stream packets whose channel number is equal to the channel field
from the BROADCAST_CHANNEL register. In particular, datagrams
addressed to 224.0.0.1 and 224.0.0.2 SHALL use this channel number.
Best-effort IP multicast for other IP multicast group addresses may
utilize a different channel number if such a channel number is
allocated and advertised prior to use, as described below.

IP-capable nodes may transmit best-effort IP multicast only if one of
the following two conditions is met:

- the channel number in the stream packet is equal to the channel
  number field in the BROADCAST_CHANNEL register and the valid bit in
  the same register is one; or

- for other channel number(s), some source of IP multicast has
  allocated and is advertising the channel number used.

The remainder of this section describes a multicast channel
allocation protocol (MCAP) employed by both IP multicast sources and
recipients whenever a channel number other than the default is used.
MCAP is a cooperative protocol; the participants exchange messages
over the broadcast channel used by all IP-capable nodes on a
particular Serial Bus.

CAUTION: This document does not define facilities and methods for
shared use of a single channel number (other than the default channel
number specified by the BROADCAST_CHANNEL register) by more than one
IP multicast address.

9.1 MCAP message format

MCAP messages, whether sent by a multicast channel owner or
recipient, are transported as the data portion of a GASP packet and
have the format illustrated below. The first four octets of the
message are fixed; the remainder consists of variable-length tuples,
each of which encodes information about a particular IP multicast
group. Individual MCAP messages SHALL NOT be fragmented and SHALL be
encapsulated within a stream packet as ether_type 0x8861.

```
                               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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            length              |    reserved   |    opcode    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                        message data                           +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
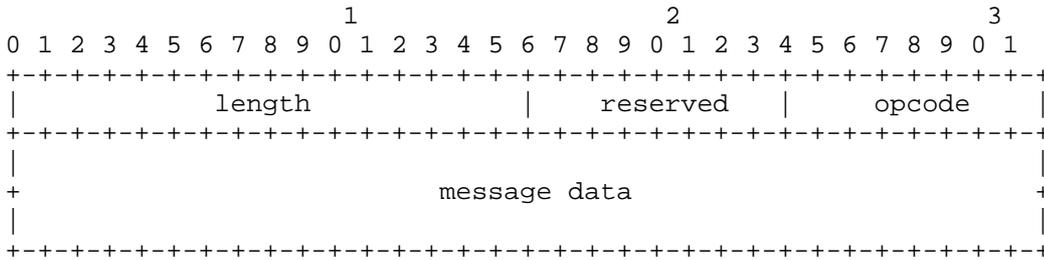
                     Figure 10 - MCAP message format

   Field usage in an MCAP message is as follows:

      length: This field SHALL contain the size, in octets, of the
      entire MCAP message.

      opcode: This field SHALL have one of the values specified by the
      table below.

```
       opcode     Name        Comment
      +-----------------------------------------------------------------+
      |   0    | Advertise | Sent by a multicast channel owner to       |
      |        |           | broadcast the current mapping(s) from one  |
      |        |           | or more group addresses to their           |
      |        |           | corresponding channel number(s).           |
      |   1    |  Solicit  | Sent to request multicast channel owner(s) |
      |        |           | to advertise the indicated channel         |
      |        |           | mapping(s) as soon as possible.            |
      +-----------------------------------------------------------------+
```

      message data: The remainder of the MCAP message is variable in
      length and SHALL consist of zero or more group address descriptors
      with the format illustrated 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      length       |       type        |         reserved      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    expiration     |     channel       |     speed     | reserved |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           bandwidth                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                              |
   +                         group_address                        +
   |                                                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

            Figure 11 - MCAP group address descriptor format

length: This field SHALL contain the size, in octets, of the MCAP
group address descriptor.

type: This field SHALL have a value of one, which indicates a
group address descriptor.

expiration: The usage of this field varies according to opcode.
For solicit messages the expiration field SHALL be IGNORED.
Otherwise, for advertisements, this field SHALL contain a time-
stamp, in seconds, that specifies a future time after which the
channel number specified by channel may no longer be used.

channel: This field is valid only for advertise messages, in which
case it SHALL specify an allocated channel number, in the range
zero to 63 inclusive. All other values are RESERVED.

speed: This field is valid only for advertise messages, in which
case it SHALL specify the speed at which stream packets for the
indicated channel are transmitted. Table 2 specifies the encoding
used for speed.

bandwidth: This field SHALL be zero; it is allocated in the group
address descriptor to accommodate future extensions to MCAP that
specify quality of service and utilize the isochronous
capabilities of Serial Bus.

group_address: This variable length field SHALL specify the IP
address of a particular IP multicast group. The length of
group_address, in octets, is derived from the length of the group
address descriptor by subtracting 12 from the length field.

9.2 MCAP message domain

   MCAP messages carry information valid only for the local Serial Bus
   on which they are transmitted. Recipients of MCAP messages SHALL
   IGNORE all MCAP messages from other than the local bus, as follows.
   The source_ID of the sender is contained in the GASP header that
   precedes the encapsulated MCAP message. A recipient of an MCAP
   message SHALL examine the most significant ten bits of source_ID from
   the GASP header; if they are not equal to either 0x3FF or the most
   significant ten bits of the recipient's NODE_IDS register, the
   recipient SHALL IGNORE the message.

   Within an MCAP message domain, the owner of a channel mapping is
   identified by the source_ID field in the GASP header of an MCAP
   advertisement. The owner is the node with the largest physical ID,
   the least significant six bits of source_ID.

9.3 Multicast receive

   An IP-capable device that wishes to receive multicast data SHALL
   first ascertain the channel mapping (if any) that exists between a
   group address and a channel number other than the default channel
   specified by the BROADCAST_CHANNEL register. Such a device may
   observe the MCAP advertisements on the broadcast channel for the
   desired channel mapping(s).

   An intended multicast recipient may transmit MCAP solicitation
   requests in order to request multicast channel owner(s) to broadcast
   advertisements sooner than the next ten second interval. Originators
   of MCAP solicitation requests SHALL limit the rate at which they are
   transmitted. Subsequent to sending a solicitation request, the
   originator SHALL NOT send another MCAP solicitation request until ten
   seconds have elapsed.

   In either case, if a mapping exists for the group address for other
   than the default channel, an MCAP advertise message is EXPECTED
   within ten seconds. Upon receipt of an MCAP advertise message that
   describes one or more channel mappings, the intended multicast
   recipient may receive IP datagrams on the indicated channel number(s)
   until the expiration time.

   If multiple MCAP advertise messages are observed that specify the
   same group address, the channel number SHALL be obtained from the
   advertisement message with the largest physical ID, which SHALL be
   obtained from the least significant six bits of source_ID from the
   GASP header.

If no MCAP advertise message is received for a particular group
address within ten seconds, no multicast source(s) are active for
channel(s) other than the default. Either there is there is no
multicast data or it is being transmitted on the default channel.

Once a multicast recipient has observed an advertisement for the
desired group address, it MAY receive multicast data on either the
default broadcast channel or the channel number(s) indicated but it
SHALL continue to monitor the default broadcast channel for MCAP
advertisements for the same group address in order to refresh the
expiration time of channel number(s) in use.

9.4 Multicast transmit

An IP-capable device that wishes to transmit multicast data on other
than the default channel SHALL first ascertain whether or not another
multicast source has already allocated a channel number for the group
address. The intended multicast source may transmit an MCAP
solicitation request with one or more group address descriptors.

Whether or not a solicitation request has been transmitted, the
intended multicast source SHALL monitor the broadcast channel for
MCAP advertisements. If a channel mapping already exists for the
group address, an MCAP advertisement SHOULD be received within ten
seconds. In this case the intended multicast source may commence
transmission of IP datagrams on the indicated channel number(s) and
may continue to do so until their expiration time. The multicast
source SHALL monitor MCAP advertisements in order to refresh the
expiration time of channel number(s) in use.

When no other multicast source has established a channel mapping for
the group address, the intended multicast source may attempt to
allocate a channel number from the isochronous resource manager's
CHANNELS_AVAILABLE register according to the procedures described in
IEEE P1394a. If the channel number allocation is successful, the
multicast source SHALL advertise the new channel mapping(s) as soon
as possible. Once 100 ms elapses subsequent to the initial
advertisement of a newly allocated channel number , the multicast
source may transmit IP datagrams using the channel number advertised.

Multicast IP datagrams may be transmitted on the default channel
until the sender observes (or transmits) an advertisement that
specifies non- default channel mapping(s) for the multicast
addresses. This permits the smooth transition of multicast from the
default channel to an explicitly allocated channel.

Once a multicast source has advertised a channel mapping, it SHALL
continue to transmit MCAP advertisements for the channel mapping
unless it either a) transfers ownership to another multicast source,
b) permits the channel mapping to expire without transfer or c) in
the case of overlapped channel mappings, relinquishes control of the
channel mapping to another multicast source.

9.5 Advertisement of channel mappings

Each multicast source SHALL periodically broadcast an advertisement
of all IP multicast group addresses for which it has allocated a
channel number different from the default multicast channel number.
An advertisement SHALL consist of a single MCAP message with an
opcode of zero that contains one or more group address descriptors
(one for each group address assigned a channel number other than that
specified by the BROADCAST_CHANNEL register).

Within each group address descriptor, the group_address and channel
fields associate an IP multicast group address with a Serial Bus
channel number. The speed field specifies the maximum 1394 speed at
which any of the senders within the IP multicast group is permitted
to transmit data. The expiration field specifies the current time or
a future time after which the channel mapping(s) are no longer valid.
Except when a channel owner intends to relinquish ownership (as
described in 9.7 below), the expiration time SHALL be at least 60
seconds in the future measured from the time the advertisement is
transmitted.

No more than ten seconds SHALL elapse from the transmission of its
most recent advertisement before the owner of a channel mapping
initiates transmission of the subsequent advertisement. The owner of
a channel mapping SHOULD transmit an MCAP advertisement in response
to a solicitation as soon as possible after the receipt of the
request.

9.6 Overlapped channel mappings

When two intended multicast sources wish to transmit to the same IP
multicast group and no channel mapping exists for the group address,
there is a chance that both will allocate channel numbers and both
will advertise the channel mappings. These channel mappings overlap,
i.e., the same group address is mapped to more than one channel
number in MCAP advertisements with nonzero expiration times.

Multicast channel owners SHALL monitor MCAP advertisements in order
to detect overlapped channel mappings. MCAP advertisements whose
expiration field has a value less than 60 SHALL be ignored for the
purpose of overlapped channel detection. When an overlapped channel

mapping is detected, the owner with the largest physical ID (as
determined by the least significant six bits of source_ID from the
GASP header) is NOT REQUIRED to take any action. The channel numbers
advertised by owners with smaller physical IDs are invalid; their
owners SHALL cease transmission of both IP datagrams and MCAP
advertisements that use the invalid channel numbers. As soon as these
channel mappings expire , their owners SHALL deallocate any unused
channel numbers as described in 9.8 below.

Recipients of MCAP advertisements that detect overlapped channel
mappings SHALL ignore the advertisements from multicast channel
owner(s) with the smaller physical IDs and SHALL NOT transmit IP
datagrams that use the invalid channel number. It is possible for
some channel mappings in a single MCAP advertisement to be valid even
if others SHALL be IGNORED as a result of overlap.

9.7 Transfer of channel ownership

The owner of a channel mapping may cease multicast transmission on a
particular channel, in which case it SHOULD invalidate the channel
mapping and in some cases deallocate the channel number. Because
other multicast sources may be using the same channel mapping, an
orderly process is defined to transfer channel ownership.

The owner of an existing channel mapping that wishes to release the
mapping SHALL commence a timer to measure the time remaining before
the anticipated release of the mapping and its associated channel.
Until the timer counts down to zero, the owner SHOULD continue to
transmit MCAP advertisements for the affected channel but SHALL
adjust expiration in each advertisement to reflect the time remaining
until the channel is to be deallocated. If the owner is unable to
transmit MCAP advertisements until the timer reaches zero, it SHALL
initiate a bus reset. Otherwise, the sequence of expiration times
transmitted by the owner intending to release the mapping SHALL
decrease with each succeeding advertisement. If other multicast
source(s) are using the same channel mapping and observe an
expiration time less than or equal to 60 seconds, they SHALL commence
transmitting MCAP advertisements for the channel mapping with
refreshed expiration times greater than or equal to 60 seconds that
maintain the channel mapping. Any contention that occurs between
multiple sources that attempt to claim ownership of the channel
mapping SHALL be resolved as described in 9.8. If the original owner
observes an MCAP advertisement for the channel to be relinquished
before its own timer has expired, it SHALL NOT deallocate the channel
number.

Otherwise, if the owner's timer expires without the observation of a
MCAP advertisement by another node, the owner of the channel number
SHALL subsequently deallocate the channel as described in 9.8. If the
intended owner of the channel mapping observes an MCAP advertisement
whose expiration field is zero, orderly transfer of the channel(s)
from the former owner has failed. The intended owner SHALL either
stop reception and transmission on the expired channel number(s) or
allocate different channel number(s) as specified by 9.4.

9.8 Redundant channel mappings

When ownership of a channel mapping is transferred from one multicast
source to another, it is possible for more than one device to claim
ownership. This results in redundant MCAP advertisements, transmitted
by different sources, each of which specifies the same multicast
group address and channel. A procedure similar to that of 9.6 SHALL
resolve the contention for channel ownership.

Multicast channel owners SHALL monitor MCAP advertisements in order
to detect redundant channel mappings. MCAP advertisements whose
expiration field has a value less than 60 SHALL be ignored for the
purpose of redundant channel detection. When a redundant channel
mapping is detected, the owner with the largest physical ID (as
determined by the least significant six bits of source_ID from the
GASP header) is NOT REQUIRED to take any action. The owner(s) with
smaller physical IDs SHALL cease transmission of MCAP advertisements
for the redundant channel number but SHALL NOT deallocate the channel
number.

9.9 Expired channel mappings

A channel mapping expires when expiration seconds have elapsed since
the most recent MCAP advertisement. At this time, multicast
recipients SHALL stop reception on the expired channel number(s).
Also at this time, the owner of the channel mapping(s) SHALL transmit
an MCAP advertisement with expiration cleared to zero and SHALL
continue to transmit such advertisements until 30 seconds have
elapsed since the expiration of the channel mapping. Once this
additional 30-second period has elapsed, the owner of the channel
mapping(s) SHALL deallocate the channel number(s) and indicate their
availability in the isochronous resource manager's CHANNELS_AVAILABLE
register.

If an IP-capable device observes an MCAP advertisement whose
expiration field is zero, it SHALL NOT attempt to allocate any of the
channel number(s) specified until 30 seconds have elapsed since the
most recent such advertisement.

9.10 Bus reset

   A bus reset SHALL invalidate all multicast channel mappings and SHALL
   cause all multicast recipients and senders to zero all MCAP
   advertisement interval timers.

   Prior owners of multicast channel mappings may reallocate a channel
   number from the isochronous resource manager's CHANNELS_AVAILABLE
   register and resume broadcast of MCAP advertisements as soon as a
   channel is allocated. If channel reallocation is attempted, the prior
   owner SHOULD use the same channel number allocated prior to the bus
   reset and may commence reallocation immediately upon completion of
   the bus reset so long as the same channel number is reused. If the
   prior owner elects to allocate a different channel number, it SHALL
   wait until at least one second has elapsed since the completion of
   the bus reset before attempting to allocate a new channel number.

   Intended or prior recipients or transmitters of multicast on other
   than the default channel SHALL NOT transmit MCAP solicitation
   requests until at least ten seconds have elapsed since the completion
   of the bus reset. Multicast data on other than the default channel
   SHALL NOT be received or transmitted until an MCAP advertisement is
   observed or transmitted for the IP multicast group address.

   Intended or prior transmitters of multicast on other than the default
   channel that did not own a channel mapping for the IP multicast group
   address prior to the bus reset SHALL NOT attempt to allocate a
   channel number from the isochronous resource manager's
   CHANNELS_AVAILABLE register until at least ten seconds have elapsed
   since the completion of the bus reset. Subsequent to this ten second
   delay, intended or prior transmitters of multicast may follow the
   procedures specified by 9.4 to allocate a channel number and
   advertise the channel mapping.

10. IANA CONSIDERATIONS

   This document necessitates the creation and management of a new name
   space (registry) by IANA. The need for such a registry arises out of
   the method by which protocol interfaces are uniquely identified by
   bus standards compliant with ISO/IEC 13213:1994, CSR Architecture.
   This is explained in more detail in section 6; the essence is that a
   globally unique 48-bit number SHALL identify the document that
   specifies the protocol interface. The 48-bit number is the
   concatenation of 0x00 005E (a registration ID, or RID, granted to
   IANA by the IEEE Registration Authority) and a second 24-bit number
   administered by IANA.

The IEEE RA RECOMMENDS that the policy for management of the second
24-bit number be chosen to maximize the quantity of usable numbers
with the range of possible values. In particular, the IEEE RA
RECOMMENDS that the assignment scheme not apply a structure to the
number (e.g., the allocation of a version field within the number)
since this would tend to waste large portions of the range.

The new name space is "CSR Protocol Identifiers". The values zero and
0xFF FFFF are reserved and SHALL NOT be allocated by IANA. The value
one is allocated to this document. The remaining numbers SHALL be
managed by IANA and allocated as necessary to identify Internet-
Drafts that become IESG standards track documents.

Regardless of the assignment method elected by IANA, a registry of
all assigned version numbers SHOULD be maintained at one or more
Internet sites and should clearly identify the relevant standard
identified by the combination of the RID and version number.

11. SECURITY CONSIDERATIONS

This document specifies 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.

12. ACKNOWLEDGEMENTS

This document represents the efforts of the IP/1394 Working Group.
The editor wishes to acknowledge the contributions made by all the
active participants, either on the reflector or at face-to-face
meetings, which have advanced the technical content.

13. REFERENCES

   Normative reference to standards under development at the time of
   this document's publication shall utilize the most current draft
   until such time as it is replaced by an approved standard.

   [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] IEEE Project P1394b, Draft Standard for a High Performance Serial
       Bus (Supplement)

   [5] Postel, J., "Internet Protocol Darpa Internet Program Protocol
       Specification", RFC 791, September 1981.

   [6] Bradner, S., "Key words for use in RFCs to Indicate Requirement
       Levels", RFC 2119, March 1997.

14. EDITOR'S ADDRESS

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

   Phone: (510) 527-3926
   Fax: (510) 527-3856
   EMail: pjohansson@aol.com

15. Full Copyright Statement

Acknowledgement