Document number 2009007

IPv4 over IEEE-1394 Test Specification

November 19,2009

Sponsored by:

1394 Trade Association

Accepted for publication by

1394 Trade Association Board of Directors

Abstract

This specification defines compliance tests for IPv4 over IEEE-1394 specifically focusing on the features and functions specific to IPv4 over IEEE-1394 and not IPv4 in general.

Keywords

IEEE 1394, Serial Bus, IPv4

1394 Trade Association Specification

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in Working Groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
315 Lincoln, Suite E
Mukilteo, WA 98275
USA

Published by

1394 Trade Association
315 Lincoln Suite E
Mukilteo, WA 98275 USA

Printed in the United States of America

Contents

Tables

Figures

Annexes

EDITOR's NOTE –

IEEE Copyright

Portions of this specification are copied from published IEEE standards, by permission.

The source documents are:

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

IEEE Std 1394a-2000, Standard for a High Performance Serial Bus – Amendment 1

The IEEE copyright policy at http://standards.ieee.org/IPR/copyrightpolicy.html states, in part:

Royalty Free Permission

IEEE-SA policy holds that anyone may excerpt and publish up to, but not more than, ten percent (10%) of the entirety of an IEEE-SA Document (excluding IEEE SIN books) on a royalty-free basis, so long as:

1) proper acknowledgment is provided;

2) the 'heart' of the standard is not entirely contained within the portion being excerpted.

This included the use of tables, graphs, abstracts and scope statements from IEEE Documents

Foreword (This foreword is not part of 1394 Trade Association Specification 2009007)

This specification defines …

There is 1 annex in this specification. Annex A, is normative and part of this specification.

This specification was accepted by the Board of Directors of the 1394 Trade Association. Board of Directors acceptance of this specification does not necessarily imply that all board members voted for acceptance. At the time it accepted this specification, the 1394 Trade Association. Board of Directors had the following members:

Max Bassler, Chair
Peter Helfet, Vice-Chair
David Thompson, Secretary

Organization Represented *Name of Representative*

....

The Compliance and Interoperability Working Group, which developed and reviewed this specification, had the following members:

Richard Mourn, Chair
Richard Mourn, Secretary

Arul Paramasivam
Dave Thompson
Mike Gardner
Toni Ray
Richard Mourn
Bill Rose
David McCubbrey
Dimitris Staikos
Sam Liu
Morten Lave

Revision history

Revision 0.1 (December 10, 2008)

Revision 0.2 (February 2, 2009)

- Changed packet formats so they will show up when made into .pdf.

Revision 0.3 (March 30, 2009)

- Added document number.

Revision 0.4 (July 14, 2009)

- Addressed Ballot Comments.

Revision 1.0 (September 2, 2009)

- Addressed second ballot comments

IPv4 over IEEE-1394 Test Specification

1 Scope and purpose

1.1 Scope

This specification defines compliance tests for IPv4 over IEEE-1394 specifically focusing on the features and functions specific to IPv4 over IEEE-1394 and not IPv4 in general.

1.2 Purpose

This specification provide implementers and testers the necessary tests to verify IPv4 over IEEE-1394 implementation.

2 Normative references

2.1 Reference scope

The specifications and standards named in this section contain provisions, which, through reference in this text, constitute provisions of this 1394 Trade Association Specification. At the time of publication, the editions indicated were valid. All specifications and standards are subject to revision; parties to agreements based on this 1394 Trade Association Specification are encouraged to investigate the possibility of applying the most recent editions of the specifications and standards indicated below.

2.2 Approved references

The following approved specifications and standards may be obtained from the organizations that control them.

IEEE Std 1394-2008, Standard for a High Performance Serial Bus

The Internet Society (1999) RFC 2734, IPv4 over IEEE-1394

Large portions of the text used in this document were taken directly from the RFC 2734. Please see the Copyright (C) The Internet Society policy for details.

Throughout this document, the term “IEEE 1394" shall be understood to refer to IEEE Std 1394-2008.

2.3 References under development

At the time of publication, the following referenced specifications and standards were under development.

2.4 Reference acquisition

The references cited may be obtained from the organizations that control them:

1394 Trade Association, 315 Lincoln, Suite E, Mukilteo, WA 98275 USA; (425) 514-8454 / (425) 710-9971 (FAX); http://www.1394ta.org/

American National Standards Institute (ANSI), 11 West 42nd Street, New York, NY 10036, USA; (212) 642-4900 / (212) 398-0023 (FAX); http://www.ansi.org/

Institute of Electrical and Electronic Engineers (IEEE), 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, USA; (732) 981-0060 / (732) 981-1721 (FAX); http://www.ieee.org/

In addition, many of the documents controlled by the above organizations may also be ordered through a third party:

Global Engineering Documents, 15 Inverness Way, Englewood, CO 80112-5776; (800) 624-3974 / (303) 792-2192; http://www.global.ihs.com/

3 Definitions and notation

3.1 Definitions

3.1.1 Conformance

Several keywords are used to differentiate levels of requirements and optionality, as follows:

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

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

3.1.1.3 may: A keyword that indicates flexibility of choice with no implied preference.

3.1.1.4 reserved: A keyword used to describe objects (bits, bytes, quadlets, octlets 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 specifications. A reserved object shall be zeroed or, upon development of a future specification, set to a value specified by such a specification. The recipient of a reserved object shall ignore its value. The recipient of an object defined by this specification as other than reserved shall inspect its value and reject reserved code values.

3.1.1.5 shall: A keyword that indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to assure interoperability with other products conforming to this specification.

3.1.1.6 should: A keyword that denotes flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “is recommended.”

3.1.2 Glossary

The following terms are used in this specification:

3.1.2.1 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.

3.1.2.2 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.

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

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

3.1.2.5 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.

3.1.2.6 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.

3.1.2.7

3.1.2.8 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.

3.1.2.9 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.

3.1.2.10 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).

3.1.2.11 octet: Eight bits of data.

3.1.2.12 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.

3.1.2.13 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.

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

3.1.2.15 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.

3.1.3 Abbreviations

The following are abbreviations that are used in this specification:

XXX The "Normal Indent" paragraph style is used for abbreviations

CSR Control and status register [B1]

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

3.2 Notation

3.2.1 Numeric values

Decimal and hexadecimal are used within this specification. 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 without subscripts or by their English names. Hexadecimal numbers are represented by digits from the character set 0 – 9 and A – F followed by the subscript 16. When the subscript is unnecessary to disambiguate the base of the number it may be omitted. For the sake of legibility hexadecimal numbers are separated into groups of four digits separated by spaces.

As an example, 42 and $2A_{16}$ both represent the same numeric value.

3.2.2 Bit, byte and quadlet ordering

This specification uses the facilities of Serial Bus, IEEE 1394, and therefore uses the ordering conventions of Serial Bus in the representation of data structures. In order to promote interoperability with memory buses that may have different ordering conventions, this specification defines the order and significance of bits within bytes, bytes within quadlets and quadlets within octlets in terms of their relative position and not their physically addressed position.

Within a byte, the most significant bit, *msb*, is that which is transmitted first and the least significant bit, *lsb*, is that which is transmitted last on Serial Bus, as illustrated below. The significance of the interior bits uniformly decreases in progression from *msb* to *lsb*.

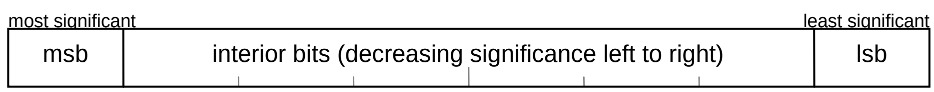

Figure 1 – Bit ordering within a byte

Within a quadlet, the most significant byte is that which is transmitted first and the least significant byte is that which is transmitted last on Serial Bus, as shown below.

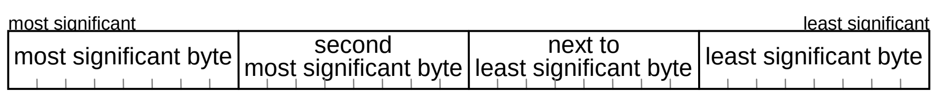

Figure 2 – Byte ordering within a quadlet

Within an octlet, which is frequently used to contain 64-bit Serial Bus addresses, the most significant quadlet is that which is transmitted first and the least significant quadlet is that which is transmitted last on Serial Bus, as the figure below indicates.

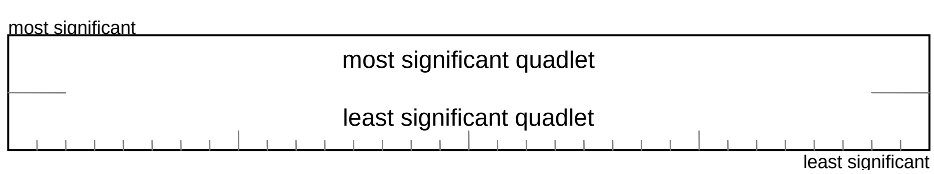

Figure 3 – Quadlet ordering within an octlet

When block transfers take place that are not quadlet aligned or not an integral number of quadlets, no assumptions can be made about the ordering (significance within a quadlet) of bytes at the unaligned beginning or fractional quadlet end of such a block transfer, unless an application has knowledge (outside of the scope of this specification) of the ordering conventions of the other bus.

4 IPv4 over 1394 Test Specification (informative)

4.1 Overview

This document specifies how to verify the use IEEE Std 1394-2008, Standard for a High Performance Serial Bus (and its supplements), for the transport of Internet Protocol Version 4 (IPv4) datagram's. It defines the tests to verify the methods, data structures and codes for that purpose and additionally defines tests to verify the methods for an address resolution protocol (1394 ARP) and a multicast channel allocation protocol (MCAP)---both of which are specific to 1394.

4.2 IEEE-1394 Background

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 3200 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.

5 IP-Capable Nodes

Not all Serial Bus devices are capable of the reception and transmission of 1394 ARP requests/responses or IP datagrams.

5.1 Minimum Requirements

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-1394-2008 and a unit directory specified in RFC-2734;
- 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 (IRM) capable, as specified by IEEE-1394-2008;
- it SHALL support both reception and transmission of asynchronous streams as specified by IEEE-1394-2008;

5.2 Minimum Requirements Test

ID	Question	Answer	Comments
IPC1	Is the Implementation Under Test (IUT) an IPv4 Capable Node?	Yes or No	
Check Point	**If you answered "No" to IPC1, IUT is not an IPv4 Capable Node. STOP and END TEST!**		
IPC2	Is the configuration ROM in the general format as specified by ISO/IEC 13213:1994?	Yes or No	See Base 1394 Test Suite Definition with Extension for 1394b for Configuration ROM General Format Test.
Check Point	**If you answered "No" to IPC2, IUT's configuration ROM is not in the correct format, FAILURE!**		
IPC3	Is the bus information block implemented as specified by IEEE-1394-2008?	Yes or No	See Base 1394 Test Suite Definition with Extension for 1394b for Bus Information Block Test.
Check Point	**If you answered "No" to IPC3, IUT's bus information block is not in the correct format, FAILURE!**		
IPC4	Is the max_rec field in its bus information block at least 8?	Yes or No	RFC2734:3

Check Point	**If you answered "No" to IPC4, IUT's max_rec field in the bus information block is not correct, FAILURE!**		
IPC5	Is the IUT IRM capable?	Yes or No	See Base 1394 Test Suite Definition with Extension for 1394b for IRM Capable Test.
Check Point	**If you answered "No" to IPC5, IUT is not IRM capable, FAILURE!**		
IPC6	Is IUT capable of transmission and reception of asynchronous stream packets?	Yes or No	
Check Point	**If you answered "No" to IPC6, IUT does not support reception and or transmission of asynchronous stream packets, FAILURE!**		

6 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.

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

Table 1 – Maximum data payload (octets)

6.1 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.

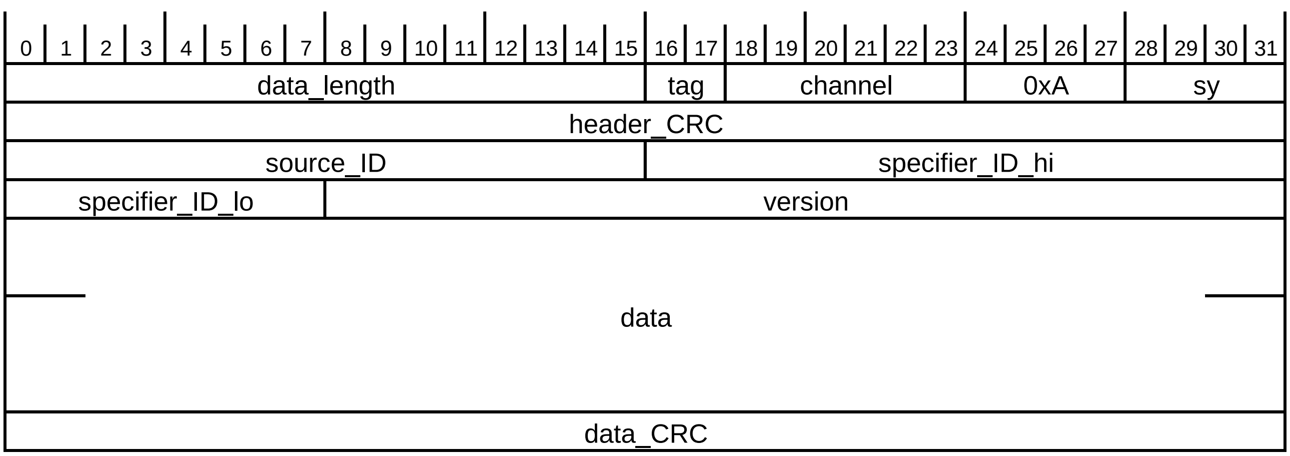

Figure 4 - GASP format

6.1.1 Source_ID field Test

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.

ID	Question	Answer	Comments
LEF1	Have IUT send an unfragmented GASP packet.	-	
LEF2	Is the source_ID field in the GASP packet the same as the most significant 16 bits of the sender’s NODE_IDS CSR register?	Yes or No	RFC2734:4.1
Check Point	**If you answered “No” to LEF2, IUT didn’t send its source_ID in the GASP packet! Failure**		

6.1.2 Specifier_ID_hi and lo Test

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.

ID	Question	Answer	Comments
LEF3	Have IUT send an unfragmented GASP packet.	-	
LEF4	Are the specifier_ID_hi and lo fields together in the GASP equal to 0x00 005E?	Yes or No	RFC2734:4.1
Check Point	**If you answered “No” to LEF4, IUT didn’t send the correct specifier_ID_hi and lo in the GASP packet! Failure**		

6.1.3 Version Test

The version field SHALL be one.

ID	Question	Answer	Comments
LEF5	Have IUT send an unfragmented GASP packet.	-	
LEF6	Is the version field in the GASP equal to 0x1?	Yes or No	RFC2734:4.1
Check Point	**If you answered “No” to LEF6, IUT didn’t send the correct version in the GASP packet! Failure**		

6.2 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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
lf		reserved														ether_type															

Figure 5 - Unfragmented encapsulation header format

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

6.2.1 Unfragmented Datagram lf Test

The lf field SHALL be zero.

ID	Question	Answer	Comments
LEF7	Have IUT send an unfragmented GASP packet.	-	
LEF8	Is the lf field in the GASP equal to 0x0?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF8, IUT didn't send the correct lf field in the GASP packet! Failure**		

6.2.2 Ether_type Test

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

Table 2 – Type of datagram

ID	Question	Answer	Comments
LEF9	Have IUT send a GASP with IPv4 datagram packet.	-	
LEF10	Is the ether_type field in the GASP equal to 0x0800?	Yes or No	RFC2734:4.1
Check	**If you answered "No" to LEF10, IUT didn't send the**		

Point	**correct ether_type in the GASP packet! Failure**		
LEF11	Have IUT send a GASP with 1394 ARP datagram packet.	-	
LEF12	Is the ether_type field in the GASP is equal to 0x0806?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF12, IUT didn't send the correct ether_type in the GASP packet! Failure**		
LEF13	Have IUT send a GASP with MCAP datagram packet.	-	
LEF14	Is the ether_type field in the GASP equal to 0x8861?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF14, IUT didn't send the correct ether_type in the GASP packet! Failure**		

6.3 Fragmented datagram test

6.3.1 First Fragmented Datagram

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.

0–1	2–3	4–15	16–31
lf	rsv	datagram_size	ether_type
dgl			reserved

Figure 6 – First fragment encapsulation header format

6.3.1.1 Fragmented Datagram lf Test

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

Table 3 – Position of link fragment within IP datagram

ID	Question	Answer	Comments
LEF15	Have IUT send a GASP packet with a fragmented datagram. The packet shall be fragmented into exactly two fragments.	-	
LEF16	Is the lf field 0x1 in the first GASP packet?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF16, IUT didn't send the correct lf field in the GASP packet! Failure**		
LEF18	Is the lf field 0x2 in the last GASP packet?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF18, IUT didn't send the correct lf field in the GASP packet! Failure**		
LEF19	Have IUT send a GASP packet with a fragmented datagram. The packet shall be fragmented into more than two fragments.	-	
LEF20	Is the lf field 0x1 in the first GASP packet?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF20, IUT didn't send the correct lf field in the GASP packet! Failure**		
LEF21	Are the lf fields of the intermediate packets 0x3 in the GASP packets?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF14, IUT didn't send the correct lf field in the GASP packet! Failure**		
LEF22	Is the lf field 0x2 in the last GASP packet?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF22, IUT didn't send the correct lf field in the GASP packet! Failure**		

6.3.1.2 Datagram_size Test

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).

ID	Question	Answer	Comments
LEF23	Have IUT send a GASP packet with a fragmented datagram.	-	
LEF24	Is the datagram_size field the same for all the fragments of the fragmented GASP packet?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF24, IUT didn't send the correct datagram_size field in the GASP packet! Failure**		

6.3.1.3 Fragmented Ether_type Test

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

Please see 6.2.2 for test details.

6.3.1.4 Dgl Test

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.

ID	Question	Answer	Comments
LEF25	Have IUT send a GASP packet with a fragmented datagram.	-	
LEF26	Is the dgl field the same for fragments of the same IP datagram?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF26, IUT didn't send the correct dgl field in the GASP packets! Failure**		
LEF27	Have IUT send multiple GASP packets with fragmented datagrams?	-	
LEF28	Is the dgl field for each consecutive IP datagram incremented from the previous datagram?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF28, IUT didn't correctly increment dgl field in the GASP packets! Failure**		
LEF29	Have IUT send multiple GASP packets until 65,535 dgl number is reached then send at least one more.	-	
LEF30	Did the dgl field wrap from 65,535 back to zero?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF30, IUT didn't correctly wrap the dgl field in the GASP packets! Failure**		

6.3.2 Subsequent Fragmented Datagram

The second and subsequent link fragments (up to and including the last) SHALL conform to the format shown below.

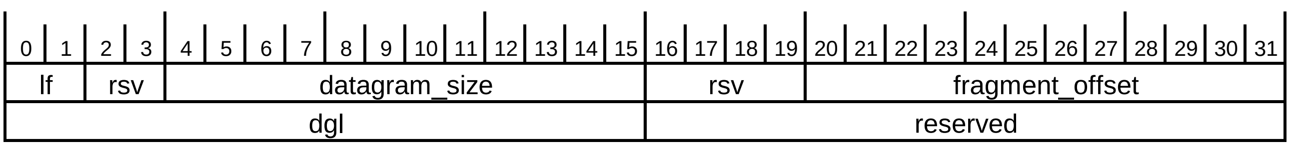

Figure 7 - Subsequent fragment(s) encapsulation header format

6.3.2.1 Fragment_offset Test

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.

ID	Question	Answer	Comments
LEF31	Have IUT send a GASP packet with a fragmented datagram.	-	
LEF32	Is the fragment_offset field correct for the 2nd to n subsequest link fragments?	Yes or No	RFC2734:4.2
Check Point	**If you answered "No" to LEF32, IUT didn't send the correct fragment_offset field in the subsequent GASP packets! Failure**		

6.4 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.

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.

Fragment reassembly is a very sensitive topic that has been heavily exploited in various forms of network attacks, so it is of primary importance to be very thorough in the related tests.

6.4.1 Fragmentation Cases

This section lists the *normal fragmentation and reassembly* test cases, that is cases that are legal and expected to occur in practice. However we must also make sure that some fragmentation patterns that are totally legal albeit a bit unusual are also handled correctly.

The table below is not a test description, just a list of the *normal fragmentation and reassembly* test cases, provided for ease of reference. Each case will have to be tested against GASP packets and asynchronous block writes. The full test procedure is provided at the end of section 6.4.

Code	Description	Comment
FG1	IP datagram is fragmented in two fragments.	Normal case.
FG2	IP datagram is fragmented in N>2 fragments.	Normal case.
FG3	IP datagram is fragmented in two fragments, with the 2nd fragment containing only 1 byte.	Rare, but can happen.
FG4	IP datagram is fragmented in N>2 fragments, with the Nth fragment containing only 1 byte.	Rare, but can happen.
FG5	IP datagram is fragmented in two fragments, with the 1st containing only 1 byte.	Most likely a fabricated attack, but still valid. It should reassemble OK.
FG6	IP datagram of N>2 bytes is fragmented in N-fragments of 1 byte.	Most likely a fabricated attack, but still valid. It should reassemble OK.

6.4.2 Concurrent Fragmented Packets

Naturally, an IUT implementation should be able to concurrently receive fragmented packets from different devices. However the IP1394 specification also permits the concurrent reception of multiple fragmented IP datagrams, originating from the **same** Source_ID, so this case needs to be tested too.

Depending on the implementation details of the sender node, the fragments may never appear interleaved in practice, for example the implementation fragments outgoing IP datagrams in a serialized fashion and then queues the packets for transmission in a batch operation. However this is an implementation detail. It is absolutely legal for fragments belonging to different IP datagrams, originating from the same device, to be transmitted in an interleaved fashion, so the IUT should be able to reassemble them all correctly.

6.4.3 Out-of-order

The IP1394 specification also permits that fragments be delivered out-of-order. An IUT implementation should be in the position to reassemble them properly.

6.4.4 Overlapping Fragments

In the case of overlapping fragments, overlapping can take several different forms, so each one, including corner cases, must be tested explicitly. The reference table below lists the possible cases that need to be tested for, giving each case a reference code that will be used in the descriptions of the actual tests, so as to keep the related descriptions short:

ID	Condition	Description
OV1	`(Offset_1 == Offset_2) &&` `(Size_1 == Size_2)`	Identical fragments.
OV2	`(Offset_1 == Offset_2) &&` `(Size_1 > Size_2)`	Fragments start at same offset, 2nd fragment smaller than the 1st.
OV3	`(Offset_1 == Offset_2) &&` `(Size_1 < Size_2)`	Fragments start at same offset, 2nd fragment larger than the 1st.
OV4	`(Offset_2 < Offset_1) &&` `((Offset_2 + Size_2) == (Offset_1 + 1))`	2nd fragment starts at earlier offset and overlaps with 1st fragment on the first byte.

OV5	`(Offset_2 < Offset_1) &&`
	`((Offset_2 + Size_2) > (Offset_1 + 1)) &&`
	`((Offset_2 + Size_2) < (Offset_1 + Size_1))`
		2nd fragment starts at earlier offset and overlaps with the middle portion of the 1st fragment (after the first byte, but before the last byte).

OV6	`(Offset_2 < Offset_1) &&` `((Offset_2 + Size_2) == (Offset_1 + Size_1))`	2nd fragment starts at earlier offset and fully overlaps with the 1st fragment, exactly up to the last byte.
OV7	`(Offset_2 < Offset_1) &&` `((Offset_2 + Size_2) > (Offset_1 + Size_1))`	2nd fragment starts at earlier offset and fully overlaps with the 1st fragment, past its last byte.
OV8	`(Offset_2 > Offset_1) &&` `((Offset_2 + Size_2) < (Offset_1 + Size_1))`	2nd fragment starts inside the 1st fragment and may extend up to, but not including, the last byte of the 1st fragment.
OV9	`(Offset_2 > Offset_1) &&` `((Offset_2 + Size_2) == (Offset_1 + Size_1))`	2nd fragment starts inside the 1st fragment and extends up to and including the last byte of the 1st fragment.
OV10	`(Offset_2 > Offset_1) &&` `((Offset_2 + Size_2) > (Offset_1 + Size_1))`	2nd fragment starts inside the 1st fragment and extends after the end of the 1st fragment.

6.4.5 Timeout

The IP1394 specification does not address the issue of timing out the IP datagram reassembly process, so this becomes an *implementation detail*. Still the IUT should be able to function properly should such a case occur and eventually release any buffer resources it might have allocated for the timed out datagram.

Obviously there is no straight-forward way to test for the timeout operation, because the result of having the IUT not handle this case correctly would probably mean that the IUT becomes unresponsive or has crashed.

6.4.6 Link Fragmented Tests

ID	**Question**	**Answer**	**Comments**
LEF33	**Have the IUT receive an IP datagram fragmented into exactly two fragments. The IP datagram is sent using GASP packets.**		**FG1**
LEF34	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF34 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF35	**Have the IUT receive an IP datagram fragmented into exactly two fragments. The IP datagram is sent using asynchronous block write packets.**		**FG1**
LEF36	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF36 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF37	**Have the IUT receive an IP datagram fragmented into more than two fragments. The IP datagram is sent using GASP packets.**		**FG2**
LEF38	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF38 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF39	**Have the IUT receive an IP datagram fragmented into more than two fragments. The IP datagram is sent using asynchronous block write packets.**		**FG2**
LEF40	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check	**If you answered "No" to LEF40 then the IUT didn't**		

Point	**correctly receive the IP datagram! Failure**		
LEF41	**Have the IUT receive an IP datagram fragmented in two fragments, with the 2nd fragment containing only 1 byte. The IP datagram is sent using GASP packets.**		**FG3**
LEF42	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF42 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF43	**Have the IUT receive an IP datagram fragmented in two fragments, with the 2nd fragment containing only 1 byte. The IP datagram is sent using asynchronous block write packets.**		**FG3**
LEF44	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF44 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF45	**Have the IUT receive an IP datagram fragmented in more than two fragments, with the last fragment containing only 1 byte. The IP datagram is sent using GASP packets.**		**FG4**
LEF46	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF46 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF47	**Have the IUT receive an IP datagram fragmented in more than two fragments, with the last fragment containing only 1 byte. The IP datagram is sent using asynchronous block write packets.**		**FG4**
LEF48	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF48 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF49	**Have the IUT receive an IP datagram fragmented in two fragments, with the 1st fragment containing only 1 byte. The IP datagram is sent using GASP packets.**		**FG5**
LEF50	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check	**If you answered "No" to LEF50 then the IUT didn't correctly receive the IP datagram! Failure**		

Point			
LEF51	Have the IUT receive an IP datagram fragmented in two fragments, with the 1st fragment containing only 1 byte. The IP datagram is sent using asynchronous block write packets.		FG5
LEF52	Is the datagram correctly received and handled?	Yes or No	RFC2734:4.3
Check Point	If you answered "No" to LEF52 then the IUT didn't correctly receive the IP datagram! Failure		
LEF53	Have the IUT receive an IP datagram of N>2 bytes fragmented in N fragments of 1 byte. The IP datagram is sent using GASP packets.		FG6
LEF54	Is the datagram correctly received and handled?	Yes or No	RFC2734:4.3
Check Point	If you answered "No" to LEF54 then the IUT didn't correctly receive the IP datagram! Failure		
LEF55	Have the IUT receive an IP datagram of N>2 bytes fragmented in N fragments of 1 byte. The IP datagram is sent using asynchronous block write packets.		FG6
LEF56	Is the datagram correctly received and handled?	Yes or No	RFC2734:4.3
Check Point	If you answered "No" to LEF56 then the IUT didn't correctly receive the IP datagram! Failure		
LEF57	Have the IUT receive fragmented IP datagrams from two devices at the same time. The fragments of the two IP datagrams are sent in an interleaved fashion. The IP datagrams are sent using GASP packets.		Version 0.3 LEF24 Concurrent Fragmented Packets
LEF58	Are the two datagrams from two different source_IDs correctly received and handled?	Yes or No	RFC2734:4.3
Check Point	If you answered “No” to LEF58, IUT didn’t correctly receive the two IP datagrams from two different nodes at the same time! Failure		
LEF59	Have the IUT receive fragmented IP datagrams from two devices at the same time. The fragments of the two IP datagrams are sent in an interleaved fashion. The IP datagrams are sent using asynchronous block write packets.		Version 0.3 LEF24 Concurrent Fragmented Packets

LEF60	**Are the two datagrams from two different source_IDs correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered “No” to LEF60, IUT didn’t correctly receive the two IP datagrams from two different nodes at the same time! Failure**		
LEF61	**Have the IUT receive two fragmented IP datagrams from the same device at the same time.** **The fragments of the two IP datagrams are sent in an interleaved fashion. The IP datagrams are sent using GASP packets.**		**Concurrent Fragmented Packets**
LEF62	**Are the two datagrams from the same source_ID correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered “No” to LEF62, IUT didn’t correctly receive the two IP datagrams from the same node at the same time! Failure**		
LEF63	**Have the IUT receive two fragmented IP datagrams from the same device at the same time. The fragments of the two IP datagrams are sent in an interleaved fashion. The IP datagrams are sent using asynchronous block write packets.**		**Concurrent Fragmented Packets**
LEF64	**Are the two datagrams from two different source_IDs correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered “No” to LEF64, IUT didn’t correctly receive the two IP datagrams from two different nodes at the same time! Failure**		
LEF65	**Have the IUT receive an IP datagram fragmented into exactly two fragments that are sent in reverse order. The IP datagram is sent using GASP packets.**		**FG1** **Out of Order Fragments**
LEF66	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check Point	**If you answered "No" to LEF66 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF67	**Have the IUT receive an IP datagram fragmented into exactly two fragments that are sent in reverse order. The IP datagram is sent using asynchronous block write packets.**		**FG1** **Out of Order Fragments**
LEF68	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check	**If you answered "No" to LEF68 then the IUT didn't correctly receive the IP datagram! Failure**		

Point			
LEF69	**Have the IUT receive an IP datagram fragmented into more than two fragments that are sent in reverse or random order. The IP datagram is sent using GASP packets.**		**FG2** **Out of Order Fragments**
LEF70	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check **Point**	**If you answered "No" to LEF70 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF71	**Have the IUT receive an IP datagram fragmented into more than two fragments that are sent in reverse or random order. The IP datagram is sent using asynchronous block write packets.**		**FG2** **Out of Order Fragments**
LEF72	**Is the datagram correctly received and handled?**	**Yes or No**	**RFC2734:4.3**
Check **Point**	**If you answered "No" to LEF72 then the IUT didn't correctly receive the IP datagram! Failure**		
LEF73	**Have the IUT receive a fragmented IP datagram, where two fragments are identical.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV1** **All the bytes of the IP datagram must be sent so as to make sure that the IUT would have all the data available to reassemble the datagram, in case the IUT did not correctly detect the overlapping condition.**
LEF74	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	**The IP1394 spec states "a fresh reassembly may be commenced" so the original question of LEF27 was changed to focus only on discarding the accumulated packets.**
Check **Point**	**If you answered "No" to LEF74 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF75	**Have the IUT receive a fragmented IP datagram, where two fragments are identical.** **All the data bytes of the IP datagram must be sent by**		**OV1**

	the sender. The IP datagram is sent using asynchronous block write packets.		
LEF76	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF76 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF77	**Have the IUT receive a fragmented IP datagram, where two fragments start at same offset, but the 2nd fragment is smaller than the 1st.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV2**
LEF78	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF78 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF79	**Have the IUT receive a fragmented IP datagram, where two fragments start at same offset, but the 2nd fragment is smaller than the 1st.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.**		**OV2**
LEF80	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF80 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF81	**Have the IUT receive a fragmented IP datagram, where two fragments start at same offset, but the 2nd fragment is larger than the 1st.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV3**
LEF82	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	

Check Point	If you answered "No" to LEF82 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF83	Have the IUT receive a fragmented IP datagram, where two fragments start at same offset, but the 2nd fragment is larger than the 1st. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.		OV3
LEF84	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF84 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF85	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV4. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.		OV4: (Offset_2 < Offset_1) && ((Offset_2 + Size_2) == (Offset_1 + 1))
LEF86	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF86 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF87	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV4. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.		OV4: (Offset_2 < Offset_1) && ((Offset_2 + Size_2) == (Offset_1 + 1))
LEF88	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF88 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF89	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV5.		OV5: (Offset_2 < Offset_1)

	All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.		**&&** **((Offset_2 + Size_2) > (Offset_1 + 1))** **&&** **((Offset_2 + Size_2) < (Offset_1 + Size_1))**
LEF90	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check **Point**	**If you answered "No" to LEF90 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF91	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV5.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.**		**OV5:** **(Offset_2 < Offset_1)** **&&** **((Offset_2 + Size_2) > (Offset_1 + 1))** **&&** **((Offset_2 + Size_2) < (Offset_1 + Size_1))**
LEF92	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check **Point**	**If you answered "No" to LEF92 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF93	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV6.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV6:** **(Offset_2 < Offset_1)** **&&** **((Offset_2 + Size_2) == (Offset_1 + Size_1))**
LEF94	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check	**If you answered "No" to LEF94 then the IUT didn't**		

Point	**correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF95	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV6.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.**		**OV6:** **(Offset_2 < Offset_1) &&** **((Offset_2 + Size_2) == (Offset_1 + Size_1))**
LEF96	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF96 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF97	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV7.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV7:** **(Offset_2 < Offset_1) &&** **((Offset_2 + Size_2) > (Offset_1 + Size_1))**
LEF98	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF98 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF99	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV7.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.**		**OV7:** **(Offset_2 < Offset_1) &&** **((Offset_2 + Size_2) > (Offset_1 + Size_1))**
LEF100	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF100 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF101	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV8.**		**OV8:** **(Offset_2 > Offset_1)**

	All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.		&& ((Offset_2 + Size_2) < (Offset_1 + Size_1))
LEF102	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF102 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF103	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV8. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.		OV8: (Offset_2 > Offset_1) && ((Offset_2 + Size_2) < (Offset_1 + Size_1))
LEF104	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF104 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF105	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV9. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.		OV9: (Offset_2 > Offset_1) && ((Offset_2 + Size_2) == (Offset_1 + Size_1))
LEF106	Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?	Yes or No	
Check Point	If you answered "No" to LEF106 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure		
LEF107	Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV9. All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.		OV9: (Offset_2 > Offset_1) && ((Offset_2 + Size_2) == (Offset_1 + Size_1))

LEF108	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF108 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF109	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV10.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using GASP packets.**		**OV10:** **(Offset_2 > Offset_1) &&** **((Offset_2 + Size_2) > (Offset_1 + Size_1))**
LEF110	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF110 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		
LEF111	**Have the IUT receive a fragmented IP datagram, with two fragments overlapping as described in case OV10.** **All the data bytes of the IP datagram must be sent by the sender. The IP datagram is sent using asynchronous block write packets.**		**OV10:** **(Offset_2 > Offset_1) &&** **((Offset_2 + Size_2) > (Offset_1 + Size_1))**
LEF112	**Did the IUT correctly determine if the fragments were overlapping and discarded the fragment(s) already accumulated in the reassembly buffer?**	**Yes or No**	
Check Point	**If you answered "No" to LEF112 then the IUT didn't correctly handled the fragmented IP datagram with overlapping fragments! Failure**		

6.4.7 Link Fragment Timeout Tests

ID	Question	Answer	Comments
LEF113	Have the IUT receive a fragmented IP datagram with at		

	least one fragment missing. The IP datagram is sent using GASP packets.		
LEF114	Have the IUT receive 10 fragmented and 10 non-fragmented IP datagrams, using a mix of GASP packets and asynchronous block write packets.		
LEF115	Are the 20 test IP datagrams correctly received and handled?	Yes or No	RFC2734:4.3
Check Point	If you answered "No" to LEF115 then the IUT may have become unresponsive or malfunctioning due to the handling of the reassembly timeout condition! Failure		
	Repeat steps LEF113 through LEF115 for ten times. Was each run successful?		FG1

7 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 below.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">hardware_type (0x0018)</td><td colspan="16">protocol_type (0x0800)</td></tr>
<tr><td colspan="8">hw_addr_len</td><td colspan="8">IP_addr_len</td><td colspan="16">opcode</td></tr>
<tr><td colspan="32" rowspan="2">sender_unique_ID</td></tr>
<tr></tr>
<tr><td colspan="8">sender_max_rec</td><td colspan="8">sspd</td><td colspan="16">sender_unicast_FIFO_hi</td></tr>
<tr><td colspan="32">sender_unicast_FIFO_lo</td></tr>
<tr><td colspan="32">sender_IP_address</td></tr>
<tr><td colspan="32">target_IP_address</td></tr>
</table>

Figure 8 - 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-1394-2008 (see also Figure 4). 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.

7.1 1394 ARP Request and Response Test

ID	Question	Answer	Comments
ARP1	Have IUT transmit a 1394 ARP request and broadcast IP datagrams.	-	
ARP2	Is the ARP request transmitted in the same way as the broadcast IP datagrams, i.e. using GASP packets?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP2, IUT didn't use the correct 1394 ARP request broadcast type! Failure**		
ARP3	Is the 1394 ARP request format the same as the shown in Figure 8? (Details of each field are tested in the following sections.)	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP3, IUT didn't implement the correct 1394 ARP format! Failure**		
ARP4	Is the 1394 ARP request encapsulated into a correctly GASP format packet?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP4, IUT didn't encapsulate the 1394 ARP request into a GASP format packet correctly! Failure**		
ARP5	Have IUT transmit a 1394 ARP response and broadcast IP datagrams.	-	
ARP6	Is the ARP response transmitted in the same way as the broadcast IP datagrams, i.e. using GASP packets?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP6, IUT didn't use the correct 1394 ARP response broadcast type! Failure**		
ARP7	Is the 1394 ARP response format the same as the shown in Figure 8? (Details of each field are tested in the following sections.)	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP7, IUT didn't implement the correct 1394 ARP format! Failure**		
ARP8	Is the 1394 ARP response encapsulated into a correctly GASP format packet?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP8, IUT didn't encapsulate the 1394 ARP response into a GASP format packet correctly! Failure**		

7.2 Hardware_type Test

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

ID	Question	Answer	Comments
ARP9	Have IUT transmit a 1394 ARP request.	-	
ARP10	Is the hardware_type field equal to 0x0018?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP10, the hardware_type field is incorrect! Failure**		

7.3 Protocol_type Test

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.

ID	Question	Answer	Comments
ARP11	Is the protocol_type field equal to 0x0800?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP11, the protocol_type field is incorrect! Failure**		

7.4 Hw_addr_len Test

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.

ID	Question	Answer	Comments
ARP12	Is the hw_addr_len field equal to 16?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP12, the hw_addr_len field is incorrect! Failure**		

7.5 IP_addr_len Test

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

ID	Question	Answer	Comments
ARP13	Is the IP_addr_len field equal to 4?	Yes or No	RFC2734:5
Check Point	**If you answered “No” to ARP13, the IP_addr_len field is incorrect! Failure**		

7.6 Opcode Test

opcode: This field SHALL be one to indicate a 1394 ARP request and two to indicate a 1394 ARP response.

ID	Question	Answer	Comments
ARP14	Is the opcode field equal to 0x0001?	Yes or No	RFC2734:5
Check Point	**If you answered “No” to ARP14, the opcode field is incorrect! Failure**		

7.7 Sender_unique_ID Test

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.

ID	Question	Answer	Comments
ARP15	Read the IUT’s unique ID from its bus information block.	-	
ARP16	Is the sender_unique_ID field equal to the unique ID retrieved from the IUT’s bus information block?	Yes or No	RFC2734:5
Check Point	**If you answered “No” to ARP16, the sender_unique_ID field is incorrect! Failure**		

7.8 Sender_max_rec Test

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

ID	Question	Answer	Comments
ARP17	Read the IUT's max_rec from its bus information block.	-	
ARP18	Is the sender_max_rec field equal to the max_rec retrieved from the IUT's bus information block?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP18, the sender_max_rec field is incorrect! Failure**		

7.9 Sspd Test

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.

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

Table 4 - Speed codes

ID	Question	Answer	Comments
ARP19	Read the IUT's link speed from its bus information block and determine its PHY speed.	-	
ARP20	Is the sspd field equal to the lesser of the IUT's link speed and PHY speed?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP20, the sspd field is incorrect! Failure**		

7.10 Sender_unicast_FIFO_hi and lo Test

sender_unicast_fifo_hi and 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 section 7 of RFC2734. The offset of a sender's unicast FIFO SHALL NOT change, except as the result of a power reset.

ID	Question	Answer	Comments
ARP21	Read the sender_unicast_FIFO_hi and lo from the 1394 ARP packet received from IUT.	-	
ARP22	Complete 1394 ARP protocol using the IUT's sender_unicast_FIFO_hi and lo address.	-	
ARP23	Was the 1394 ARP protocol successful using the sender_unicast_FIFO_hi and lo address?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP23, the sender_unicast_FIFO_hi and lo address was incorrect! Failure**		
ARP24	Initiate a 1394 bus reset	-	
ARP25	Have IUT initiate an ARP packet.	-	
ARP26	Is the sender_unicast_FIFO_hi and lo that same in the ARP packet before and after the bus reset?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP26, IUT changed its sender_unicast_FIFO_hi and lo address and it shouldn't have! Failure**		

7.11 Sender_IP_address Test

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

ID	Question	Answer	Comments
ARP27	Get the IUT's IP address from the OS or other independent means.	-	
ARP28	Is the sender_IP_address field equal to the IP address retrieved in step ARP27?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP28, the send_IP_address field is incorrect! Failure**		

7.12 Target_IP_address Test

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.

ID	Question	Answer	Comments
ARP29	Get the target's IP address from the OS or other independent means.	-	
ARP30	Have IUT send a 1394 ARP to the target.	-	
ARP31	Is the target_IP_address field equal to the IP address retrieved in step ARP29?	Yes or No	RFC2734:5
Check Point	**If you answered "No" to ARP31, the target_IP_address field is incorrect! Failure**		

8 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-1212-2001.

8.1 Unit_Spec_ID Test

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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x12								unit_spec_ID (0x00 005E)																							

Figure 9 – 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.

ID	Question	Answer	Comments
CR1	Read the Unit_Spec_ID entry from IUT's configuration ROM	-	
CR2	Is the Unit_Spec_ID entry read equal to: 0x1200 005E	Yes or No	RFC2734:6.1
Check Point	**If you answered "No" to CR2, the Unit_Spec_ID entry is incorrect! Failure**		

8.2 Unit_SW_Version Test

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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x13								unit_sw_version (0x00 0001																							

Figure 10 - 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.

ID	Question	Answer	Comments
CR3	Read the Unit_SW_Version entry from IUT's configuration ROM	-	
CR4	Is the Unit_SW_Version entry read equal to: 0x1300 0001	Yes or No	RFC2734:6.2
Check Point	**If you answered "No" to CR4, the Unit_SW_Version entry is incorrect! Failure**		

8.3 Textual Descriptors Test

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.

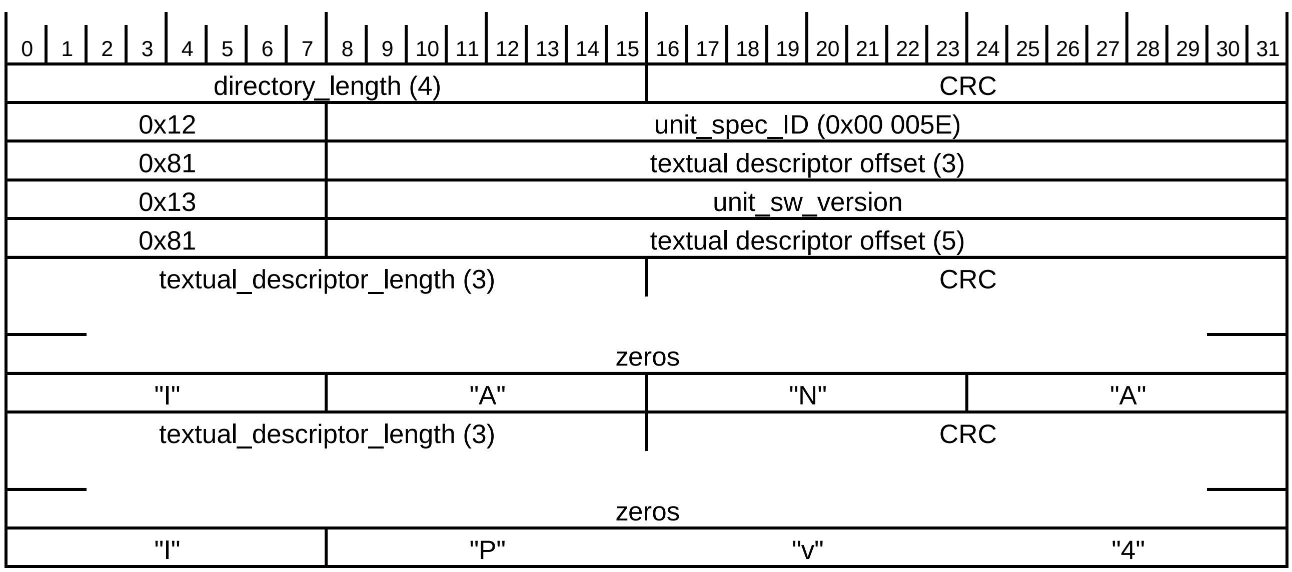

Figure 11 – Sample unit directory and textual descriptors

If textual descriptors are implemented the following test shall be executed.

ID	Question	Answer	Comments
CR5	Read the Unit_SW_Version entry from IUT's configuration ROM	-	
CR6	Is the Unit_SW_Version entry read equal to: 0x1300 0001	Yes or No	RFC2734:6.2
Check Point	**If you answered "No" to CR6, the Unit_SW_Version entry is incorrect! Failure**		

9 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:

- **Obsolete:** 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;
- 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.

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.

9.1 IP Unicast Test

ID	Question	Answer	Comments
IPU1	Setup an TCP Unicast connection.	-	
IPU2	Determine which primary packet type IUT uses for TCP Unicast connection communication.	-	Record for later use
IPU3	Is the payload size sent by the IUT the minimum of the largest data payload implemented by the sender, the recipient and the PHYs of all intervening nodes and the packet is transmitted at the maximum possible speed permitted by the path between the two nodes?	Yes or No	RFC2734:7
Check Point	**If you answered "No" to IPU3, the payload size is to large! Failure**		
IPU4	If the IUT primary packet type used is block write is the transactions addressed to the source_ID and sender_unicast_FIFO obtained from a 1394 ARP response?	Yes or No	RFC2734:7
Check Point	**If you answered "No" to IPU4, the datagram is incorrectly addressed! Failure**		

10 IP Broadcast

Broadcast IP datagrams are encapsulated according to the specifications of section 4 of RFC 2734 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.

ID	Question	Answer	Comments
IPB1	Send an IP Broadcast packet.	-	
IPB2	Is the channel number used by IUT equal to the channel field from the BROADCAST_CHANNEL register?	Yes or No	RFC2734:8
Check Point	**If you answered “No” to IPU2, IUT used the incorrect channel number! Failure**		
IPB3	Make IUT to not be IRM.	-	
IPB4	Initiate a 1394 bus reset	-	
IPB5	Did the IUT clear its valid bit in its BROADCAST_CHANNEL when the bus reset occurred?	Yes or No	RFC2734:8
Check Point	**If you answered “No” to IPB5, IUT didn’t correctly clear its valid bit! Failure**		
IPB6	Did the IUT read the BROADCAST_CHANNEL register after the bus reset to determine if its valid bit was set?	Yes or No	RFC2734:8
Check Point	**If you answered “No” to IPB6, IUT didn’t determine if IP Broadcast could be restarted! Warning**		
IPB7	Did the IUT send IP Broadcast packet(s) between the bus reset and before it set its valid bit in its BROADCAST_CHANNEL register?	Yes or No	RFC2734:8
Check Point	**If you answered “No” to IPB7, IUT sent IP Broadcast packet without valid bit being set! Failure**		

11 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.

11.1 Asynchronous Stream Support

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.

11.1.1 IP Multicast Test

ID	Question	Answer	Comments
IPM1	Setup an IP Multicast connection using IP address 224.0.0.1.	-	
IPM2	Is the channel number used by IUT equal to the channel field from the BROADCAST_CHANNEL register?	Yes or No	RFC2734:9
Check Point	**If you answered “No” to IPM2, IUT used the incorrect channel number! Failure**		
IPM3	Setup an IP Multicast connection using IP address 224.0.0.2.	-	
IPM4	Is the channel number used by IUT equal to the channel field from the BROADCAST_CHANNEL register?	Yes or No	RFC2734:9
Check Point	**If you answered “No” to IPM4, IUT used the incorrect channel number! Failure**		

11.2 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.

0–15	16–23	24–31
length	reserved	opcode
message data		

Figure 12 - MCAP message format

11.2.1 MCAP Message Format Test

ID	Question	Answer	Comments
IPM5	Have IUT send MCAP messages.	-	
IPM6	Are the MCAP message not fragmented?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM6, IUT incorrectly fragmented MCAP messages! Failure**		
IPM7	Is the ether_type field within the MCAP messages equal to: 0x8861?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM7, the ether_type field is incorrect.**		

11.2.2 MCAP Message length Test

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

ID	Question	Answer	Comments
IPM8	Have IUT send MCAP messages.	-	
IPM9	Is the size of the MCAP message equal to the size indicated by the length field?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM9, IUT put the wrong size into the MCAP messages length field! Failure**		

11.2.3 MCAP Message opcode Test

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 corresponding channel number(s).
1	Solicit	Sent to request multicast channel owner(s) mapping(s) as soon as possible.

Table 5 MCAP Opcodes

ID	Question	Answer	Comments
IPM10	Have IUT send an Advertise MCAP messages.	-	
IPM11	Is the opcode of the Advertise MCAP message opcode equal to zero?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM11, IUT put the wrong opcode into the MCAP messages opcode field! Failure**		
IPM12	Have IUT send an Solicit MCAP messages.	-	
IPM13	Is the opcode of the Solicit MCAP message opcode equal to one?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM13, IUT put the wrong opcode into the MCAP messages opcode field! Failure**		

11.3 MCAP message data

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.

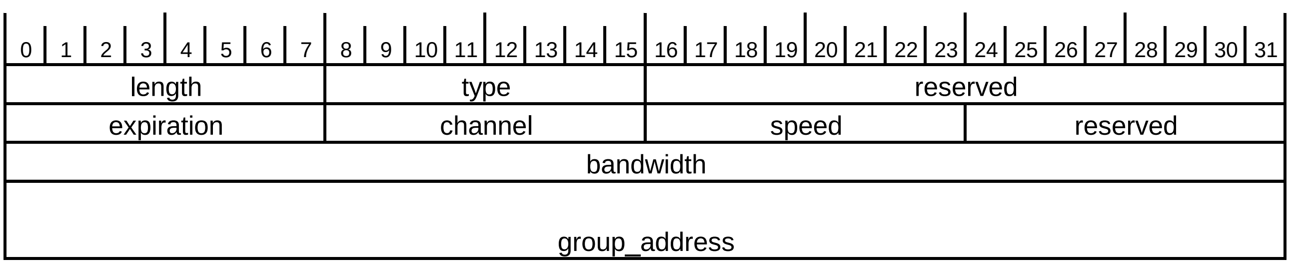

Figure 13 – MCAP group address descriptor format

11.3.1 MCAP Group Address Descriptor length Test

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

ID	Question	Answer	Comments
IPM14	Have IUT send a MCAP message with a group address descriptor.	-	
IPM15	Is the size of the MCAP group address descriptor equal to the size indicated by the length field?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM15, IUT put the wrong size into the MCAP messages length field! Failure**		

11.3.2 MCAP Group Address Descriptor type Test

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

ID	Question	Answer	Comments
IPM16	Have IUT send a MCAP message with a group address descriptor.	-	
IPM17	Is the type field of the MCAP group address descriptor equal to one?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM17, IUT put the wrong type into the MCAP messages type field! Failure**		

11.3.3 MCAP Group Address Descriptor expiration Test

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.

Please see the MCAP Advertise Message Time Test for test details.

11.3.4 MCAP Group Address Descriptor channel Test

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.

ID	Question	Answer	Comments
IPM18	Have IUT send an advertize MCAP message with a group address descriptor to channel 31.	-	
IPM19	Is the channel field of the MCAP group address descriptor equal to 31?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM19, IUT put the wrong channel into the MCAP messages channel field! Failure**		

11.3.5 MCAP Group Address Descriptor speed Test

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. - Speed codesTable 4 specifies the encoding used for speed.

ID	Question	Answer	Comments
IPM20	Have IUT send an advertize MCAP message with a group address descriptor at the maximum possible speed.	-	
IPM21	Is the speed field of the MCAP group address descriptor equal to the maximum possible speed?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM21, IUT put the wrong speed into the MCAP messages speed field! Failure**		

11.3.6 MCAP Group Address Descriptor bandwidth Test

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.

ID	Question	Answer	Comments
IPM22	Have IUT send a MCAP message with a group address descriptor.	-	
IPM23	Is the bandwidth field of the MCAP group address descriptor equal to zero?	Yes or No	RFC2734:9.1
Check Point	**If you answered "No" to IPM23, IUT put the wrong bandwidth into the MCAP messages bandwidth field! Failure**		

11.3.7 MCAP Group Address Descriptor group_address Test

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.

ID	Question	Answer	Comments
IPM24	Create a multicast group using a specific IP address. Have IUT send an MCAP message with a group address descriptor to that specific IP address.	-	
IPM25	Is the group_address field of the MCAP group address descriptor equal to the specific IP address?	Yes or No	RFC2734:9.1
Check Point	**If you answered “No” to IPM25, IUT put the wrong IP address into the MCAP messages group_address field! Failure**		

11.4 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.

11.4.1 MCAP Local Bus Address Test

ID	Question	Answer	Comments
IPM26	Send IUT an MCAP message with bus id (most significant ten bits of the source_ID field) not equal to 0x3FF or IUT local bus number.	-	
IPM27	Does IUT ignore this MCAP message?	Yes or No	RFC2734:9.2
Check Point	**If you answered “No” to IPM27, IUT didn’t ignore a non-local bus MCAP message! Failure**		

11.5 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.

11.5.1 Multicast Receive Get Channel Test

This is generally done manually so know test is defined at this time.

11.5.2 Multicast Receive Solicitation Request Rate Test

ID	Question	Answer	Comments
IPM28	If capable, have IUT issue solicitation requests to a specified channel. Make sure know response is issued to the solicitation request.	-	
IPM29	Does IUT issue solicitation requests at greater than 10 second intervals?	Yes or No	RFC2734:9.3
Check Point	**If you answered “No” to IPM29, IUT issue solicitation requests to quickly! Failure**		

11.5.3 MCAP Advertise Message Time Test

ID	Question	Answer	Comments
IPM30	If capable, have IUT Advertise message.	-	
IPM31	Are IUT Advertise messages sent at 10 second or less intervals?	Yes or No	RFC2734:9.3
Check Point	**If you answered "No" to IPM31, IUT doesn't issue advertisements quickly enough! Failure**		

11.5.4 Group Channel Number Determination

ID	Question	Answer	Comments
IPM32	If IUT is capable of group channel determination, have multiple devices send MCAP advertisements with the same group address. Each advertisement shall contain a different channel number.	-	
IPM33	Have IUT try to obtain group address/channel number.	-	
IPM34	Is the channel number obtained by the IUT equal to the channel number used by the device with the largest physical ID?	Yes or No	RFC2734:9.3
Check Point	**If you answered "No" to IPM34, IUT doesn't issue advertisements quickly enough! Failure**		

11.5.5 Reception on non-Default Channel Test

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.

11.6 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-1394-2008. 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.

11.6.1 Multicast Transmit Get Channel Test

This is generally done manually so know test is defined at this time.

11.6.2 Monitor Broadcast Channel for MCAP Advertisements Test

ID	Question	Answer	Comments
IPM35	Establish a group address and have it use a non-default channel. Have a device advertize the group address.	-	
IPM36	Have IUT attempt to transmit datagrams to group address established in the previous step.	-	
IPM37	Did IUT wait to transmit using the group address's channel until after is observed at least one advertisement?	Yes or No	RFC2734:9.4
Check Point	**If you answered "No" to IPM37, IUT doesn't appears to monitoring before starting to use multicast channel number! Failure**		

11.6.3 Monitor MCAP Advertisements to Refresh Expiration Test

ID	Question	Answer	Comments
IPM38	Establish a group address and have it use a non-default channel. Have a device advertize the group address.	-	
IPM39	Have IUT attempt to transmit datagrams to group address	-	

	established in the previous step.		
IPM40	Have device stop advertizing group address	-	
IPM41	Did IUT stop transmitting on the channel after its expiration timer expired?	Yes or No	RFC2734:9.4
Check Point	**If you answered "No" to IPM41, IUT didn't stop transmitting when it expiration timer expired! Failure**		

11.6.4 Allocation of Channel Test

ID	Question	Answer	Comments
IPM42	Make sure no devices are advertising for a specific channel mapping	-	
IPM43	Make sure the IUT is not IRM	-	
IPM44	Have IUT establish a channel mapping using a non-default channel.	-	
IPM45	Did IUT correctly allocate a channel number from the IRM using a compare/swap lock operation?	Yes or No	RFC2734:9.4
Check Point	**If you answered "No" to IPM45, IUT didn't correctly allocate a channel number from the IRM! Failure**		

11.6.5 Advertisement of Allocated Channel Test

ID	Question	Answer	Comments
IPM46	Follow on to Test ID IPM45	-	
IPM47	Did IUT correctly advertise the allocated channel mapping?	Yes or No	RFC2734:9.4
Check Point	**If you answered "No" to IPM47, IUT didn't correctly advertize the channel mapping! Failure**		

11.6.6 Time from First Advertisement to Use Test

ID	Question	Answer	Comments
IPM48	Follow on to Test ID IPM47	-	
IPM49	Did IUT wait at least 100ms from the initial advertisement	Yes or No	RFC2734:9.4

	before it started to use the channel?		
Check Point	**If you answered "No" to IPM49, IUT didn't wait long enough before starting to use a new channel! Failure**		

11.7 Advertisement of Channel Mapping

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 of the standard), 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.

11.7.1 Periodic Advertisement Timing Test

11.7.1.1 Timing

ID	Question	Answer	Comments
IPM50	Follow on to Test ID IPM47	-	
IPM51	Does IUT advertise the channel mapping approximately every 10 seconds or less?	Yes or No	RFC2734:9.5
Check Point	**If you answered "No" to IPM51, IUT doesn't periodically send channel mapping advertisement correctly! Failure**		

11.7.1.2 Format

ID	Question	Answer	Comments
IPM52	Follow on to Test ID IPM51	-	
IPM53	Does IUT advertisement consist of a single MCAP message with an opcode of zero and one or more group address descriptors?	Yes or No	RFC2734:9.5
Check Point	**If you answered "No" to IPM53, IUT advertisement format isn't correct! Failure**		

11.7.1.3 Expiration Timing

ID	Question	Answer	Comments
IPM54	Follow on to Test ID IPM53	-	
IPM55	Is the expiration time in each intermediate advertisement greater than or equal to 60 seconds in the future from when the advertisement was transmitted?	Yes or No	RFC2734:9.5
Check Point	**If you answered "No" to IPM55, IUT expiration time isn't correct! Failure**		

11.8 Overlapped Channel Mapping

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 of the standard.

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.

11.8.1 Overlapped Channel Mapping Test

ID	Question	Answer	Comments
IPM56	Have IUT establish a channel map and start advertising this map.	-	
IPM57	Have a tester, with a larger physical_ID than IUTs, advertise the same channel as IUT with expiration time of less than 60 seconds.	-	
IPM58	Does IUT continue to issue MCAP advertisements and datagrams using the established channel number.	Yes or No	RFC2734:9.6
Check Point	**If you answered "No" to IPM58, IUT incorrectly stopped issuing using overlapped channel when**		

	expiration was less than 60 seconds! Failure		
IPM59	Have a tester, with small physical_ID than IUTs, advertise the same channel as IUT with expiration time greater than 60 seconds.	-	
IPM60	Does IUT continue to issue MCAP advertisements and datagrams using the established channel number.	Yes or No	RFC2734:9.6
Check Point	**If you answered “No” to IPM60, IUT incorrectly stopped issuing using overlapped channel when its physical_ID is larger! Failure**		
IPM61	Have a tester, with larger physical_ID than IUTs, advertise the same channel as IUT with expiration time greater than 60 seconds.	-	
IPM62	Does IUT stop issueing MCAP advertisements and datagrams using the established channel number.	Yes or No	RFC2734:9.6
Check Point	**If you answered “No” to IPM62, IUT incorrectly didn’t stopped issuing using overlapped channel when its physical_ID was smaller! Failure**		

11.9 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 of the standard. 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 of the standard. 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 of the standard.

11.9.1 Transfer of Channel Ownership Test

ID	Question	Answer	Comments
IPM63	Have Tester establish a channel map and start advertising this map.	-	
IPM64	Have IUT start using the channel map.	-	
IPM65	Have Tester lower it expiration time below 60 seconds in 10 second intervals.	-	
IPM66	Does IUT start to issue MCAP advertisements for channel mapping?	Yes or No	RFC2734:9.7
Check Point	**If you answered "No" to IPM66, IUT didn't start advertising a channel it is using when the owners expiration was below 60 seconds! Failure**		
IPM67	Have IUT establish a channel map and start advertising the map.	-	
IPM68	Have Tester start using the channel map.	-	
IPM69	Have IUT stop using the channel map	-	
IPM70	Does IUT reduce the expiration time below 60 seconds?	Yes or No	RFC2734:9.7
Check Point	**If you answered "No" to IPM70, IUT didn't correctly reduce its expiration time when it stops using a channel map! Failure**		
IPM71	Continue on from IPM71:	-	
IPM72	Did IUT not deallocate the channel number being used?	Yes or No	RFC2734:9.7
Check Point	**If you answered "No" to IPM72, IUT incorrectly deallocated a channel used by another device! Failure**		

11.10 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 of the standard 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.

Test for this case are provided in other sections.

11.11 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.

11.11.1 Expired Channel Mapping Test

ID	Question	Answer	Comments
IPM73	Have IUT establish a channel map and start advertising this map.	-	
IPM74	Have stop using the channel map.	-	
IPM75	Does IUT continue to advertise the channel map with a zero expiration time for an additional 30 seconds?	Yes or No	RFC2734:9.9
Check Point	**If you answered "No" to IPM75, IUT didn't correctly implement expiration of the channel mapping! Failure**		
IPM76	Continue from IPM75	-	
IPM77	After the additional 30 seconds did IUT deallocate the channel from the CHANNELS_AVAILABLE register	Yes or No	RFC2734:9.9
Check Point	**If you answered "No" to IPM77, IUT didn't correctly implement deallocation of the channel! Failure**		

11.12 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 of the standard to allocate a channel number and advertise the channel mapping.

11.12.1 Bus Reset Test

ID	Question	Answer	Comments
IPM78	Have IUT establish a channel map and start advertising this map. Establish a Tester as the IRM.	-	
IPM79	Initiate a bus reset.	-	
IPM80	Does IUT reallocate the channel number from the CHANNELS_AVAILABLE register within one second of the bus reset ?	Yes or No	RFC2734:9.10
Check Point	**If you answered “No” to IPM80 skip to ID 81.**		
Check Point	**If you answered “Yes” to IPM80 IUT reallocated using the same channel number skip to ID 82**		
IPM81	Did IUT wait at least 1 second to allocation a new channel?	Yes or No	RFC2734:9.10
Check Point	**If you answered “No” to IPM81 IUT allocated a new channel to early! Failure**		
IPM82	Initiate a bus reset	-	
IPM83	Have IUT attempt to establish a channel mapping within one second of the bus reset. Tester as the IRM.	-	
IPM84	Did IUT wait at least 10 seconds from the bus reset before it attempts to allocate the channel?	Yes or No	RFC2734:9.10
Check Point	**If you answered “No” to IPM84 IUT didn’t wait long enough to allocated a new channel after a bus reset! Failure**		

12 Point-to-Point and Network Tests

Test ID	Test Description	Answer
PP1	Connect IUT to IP 1394 capable test device. Was the reference device correctly listed in IUT's device manager/registry?	Yes or No
PP2	Have IUT write one giga-byte or larger file to the test device. Was file transfer completed successfully?	Yes or No or N/A
PP3	Have IUT read one giga-byte or larger file from the test device. Was file transfer completed successfully?	Yes or No or NA
PP4	Repeat steps PP2 and PP3 4 times. Was each file transfer completed successfully?	Yes or No or NA
PP5	Have the reference device write one giga-byte or larger file to IUT. Was file transfer completed successfully?	Yes or No or N/A
PP6	Have the IUT read one giga-byte or larger file from the reference device. Was file transfer completed successfully?	Yes or No or NA
PP7	Repeat steps PP5 and PP6 4 times. Was each file transfer completed successfully?	Yes or No or NA
PP8	Have the reference device and IUT write and read one giga-byte or larger file to each other at the same time. Were the file transfers completed successfully?	Yes or No or N/A
PP9	Repeat step PP8 4 times. Was each file transfer completed successfully?	Yes or No or NA

Annex A
(informative)

Bibliography

[B1] IEEE Std 1212-2001, Standard for a Control and Status Registers (CSR) Architecture for microcomputer buses

[B2] IEEE Std 1394-2008, Standard for a High Performance Serial Bus

[B3] The Internet Society (1999) RFC 2734, IPv4 over IEEE-1394