Document number 2006021

Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable—Part 3: FCP and CMP over IPv4

July 12, 2008

Accepted for publication by

The 1394 Trade Association Board of Directors

Abstract

This technical specification standardizes a function control protocol (FCP/IP) and connection management procedures (CMP/IP) to be used in lieu of FCP and CMP as originally specified by IEC 61883-1. These methods, based upon IPv4, are intended for use by AV/C devices, in a network connected by L3 IP bridges, when at least one of the controller, talker (source) and listeners (sinks) is in a different IEEE 1394 cluster than the others.

Keywords

AV/C, CMP, FCP, IEC 61883-1, IEEE 1394, IPv4, isochronous, PCR, plug control register, Serial Bus

1394 Trade Association Technical Specification

1394 Trade Association Technical Specifications are developed within Working Groups of the 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 technical specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Technical Specification is voluntary. The existence of a 1394 Trade Association Technical 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 Technical Specification. Furthermore, the viewpoint expressed at the time a technical specification is accepted and published is subject to change brought about through developments in the state of the art and comments received from users of the technical specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Technical Specification.

Comments for revision of 1394 Trade Association Technical 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.

Questions might arise about the meaning of technical specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will prepare appropriate responses.

Comments on technical specifications and requests for interpretations should be addressed to the address below:

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

1394 Trade Association Technical Specifications are accepted by the association without regard to patents that might exist on articles, materials or processes or to other proprietary intellectual property that might exist within technical specifications. Acceptance of a technical specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the technical specifications. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights that might be infringed by conformance to this technical specification.

Published by

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

Copyright © 2008 by 1394 Trade Association
All rights reserved.

Printed in the United States of America

Contents

Tables

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Figures

Annexes

Copyright © 2008, 1394 Trade Association. All rights reserved.

Foreword (This foreword is not a normative part of 1394 Trade Association Specification 2006021)

During June, July and August of 2005, a 1394 Trade Association delegation visited a number of multiple system operators (MSOs). The purpose was straightforward: to determine MSO requirements for IEEE 1394 that, if satisfied, could speed the deployment of IEEE 1394 in the residential environment. Broadly speaking, the MSOs require a network capable of distributing multiple program streams from a set-top box (STB) equipped with multiple tuners to different televisions throughout the residence. In-depth discussion with the MSOs yielded the detailed requirements below:

- The network has to operate over existing coaxial cable infrastructure without disturbance to incumbent CATV services or services whose deployment is imminent. The MSOs postulated that a "typical" existing installation might have a diameter (*i.e.*, the maximum cable distance between any two wall plates) of 100 m and that the path between any two wall plates might pass through a 4-way signal splitter and up to two 2-way signal splitters. The network should be able to provide equal quality of service between all possible wall plate pairings—although remediation, *i.e.*, installation of higher-quality coaxial cable and higher-quality signal splitters, is tolerable in a small number of cases;
- The network requires sufficient bandwidth to transport up to four HD or SD program streams concurrently. Allowing for the increased bandwidth requirements of "trick" play, *e.g.*, fast-forward, single frame advance, *etc.*, the MSOs estimate a minimum requirement of 300 Mb/s of isochronous data;
- The network protocols should support transmission of the program guide from the STB to a television;
- The network has to interconnect clusters of Ethernet devices and permit them to communicate as if they were connected to the same local-area network; and
- The network has to support communications between IEEE 1394 AV disk drives, STBs, personal video recorders (PVRs) and televisions—but the MSOs are indifferent with respect to support for other IEEE 1394 devices.

Separate from the MSO requirements, FCC rules mandate that the network protocols support the transport of commands specified by CEA 775, "DTV 1394 Interface Specification" and CEA 931, "Remote Control Command Pass-through Standard for Home Networking".

December of that same year saw the launch of the High-Definition Audio-Video Network Alliance (HANA). The alliance intends to satisfy all of the MSO requirements enumerated above and, in addition, provide network support for legacy IEEE 1394 AV/C devices—although not necessarily for HANA's initial deployment. HANA is also the key proponent for CEA 2027-B, "A User Interface for Home Networks Using Web-based Protocols".

Through the development of a comprehensive family of technical specifications (of which this document is a part), the 1394 Trade Association plans to satisfy both MSO and HANA requirements. Because association member companies have a large infrastructure investment in contemporary, deployed IEEE 1394 AV/C devices that are not network-enabled, the newly designed network functionality must be at least as rich as the functionality available today, within a local cluster, to "legacy" devices. In particular:

- Network-enabled AV/C devices must a) be able to discover and operate with both network-enabled and legacy AV/C devices within the local cluster and b) additionally be able to discover and operate with network-enabled AV/C devices connected to the network but not to the local cluster;
- The functionality described above for network-enabled AV/C devices should be made available to legacy AV/C devices. This may require network-enabled "helper" devices that communicate across bridges on behalf of legacy AV/C devices;
- Mindful of the latency introduced by bridges, the network must be designed to connect IEEE 1394 clusters across a maximum of two intervening bridges connected by coaxial cable. If the accumulated latencies are

Copyright © 2008, 1394 Trade Association. All rights reserved.

small enough, it might be possible for remote devices to operate across more than two bridges—but this is neither guaranteed nor required; and

– The network and network-enabled AV/C devices must together provide digital content protection at least as robust as that available within a local IEEE 1394 cluster.

Because of the importance of the residential network to future growth in the deployment of IEEE 1394 devices, the 1394 Trade Association Wireless working group commenced development in 2006 of the following technical specifications organized under the family title " Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable ":

– Part 1: Continuous Pulsed Ultra-wideband (C-UWB) PHY
– Part 2: L3 IP Bridges
– Part 3: FCP and CMP over IPv4
– Part 4: AV/C Relay Agents

Part 1 specifies a physical layer (PHY), which is, in combination with the Medium Access Control (MAC) sublayer specified by IEEE Std 802.15.3b-2006, suitable for the coaxial cable portion of the residential network.

Part 2 specifies layer 3 (L3) bridges capable of connecting isolated IEEE 1394 clusters into a residential network *via* Internet protocol (IP) and subsequently accepting messages to configure the flow of additional isochronous data from one cluster to another.

Part 3 (this document) specifies methods to transport commands (and receive status in return) within the residential network by use of IPv4 rather than by IEEE 1394-specific methods. It also provides methods to program plug control registers *via* IP messages rather than by IEEE 1394-specific methods. These facilities enable AV/C to be used outside of the local cluster—even to be controlled from locations outside the residence.

Part 4 specifies how third-party devices acting as relay agents for legacy AV/C devices render them capable of interoperation with AV/C devices in remote clusters.

The Board of Directors of the 1394 Trade Association accepted this technical specification on July 27, 2008. Board of Directors acceptance of this technical specification does not necessarily imply that all board members voted for acceptance. At the time the 1394 Trade Association Board of Directors accepted this technical specification, it had the following members:

Eric Anderson, Chair
Max Bassler, Vice-Chair
Dave Thompson, Secretary

Organization Represented	*Name of Representative*
Apple	Eric Anderson
EqcoLogic	Peter Helfet
Interactive Technology	Max Bassler
LSI	Dave Thompson
Microsoft	Mark Slezak
Oxford Semiconductor	Don Harwood
Symwave	Burke Henehan
Texas Instruments	Will Harris
WJR Consulting	Bill Rose

Copyright © 2008, 1394 Trade Association. All rights reserved.

The Wireless Working Group, which developed and reviewed this technical specification, had the following participants:

Hans van der Ven, Chair
Michael Scholles, Vice-chair
Allen Heberling, Secretary

Eric Anderson
Yasaman Bahreini
Max Bassler
Les Baxter
Duncan Beadnell
Jack Chaney
Mike Conroy
Zephra Freeman
Robert Fust
Sergio Guillén
Will Harris
Allen Heberling
Peter Helfet
Peter Johansson
Hyunchin Kim
Todd Krein
Francesco Liburdi
Richard Mourn
Jalil Oraee
Steve Powers
Bill Rose
Kenji Sakamoto
John Santhoff
Hideoki Sato
Tsuyoshi Sawada
Michael Scholles
Mark Slezak
Dave Thompson
Masanori Tsuji
Koen Van den Brande
Hans Van der Ven
Colin Whitby-Strevens
Andy Yanowitz

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable— Part 3: FCP and CMP over IPv4

1 Scope and purpose

1.1 Scope

This is a technical specification whose scope is the use of Internet protocol as the basis for a) device and service announcement and discovery methods less burdensome than brute-force searches of device configuration ROM, b) a command and response transport protocol analogous to the IEC 61883-1 Function Control Protocol (FCP) and c) isochronous connection management procedures similar to the IEC 61883-1 Connection Management Procedures (CMP). These alternative methods provide a migration path for the AV/C command set towards networks based upon Internet protocol. The methods must function properly both for devices connected to the same IEEE 1394 cluster and for devices connected to different IEEE 1394 clusters within a network joined by L3 IP bridges (see 1.3). This technical specification standardizes the all aspects of these Internet protocol-based facilities in order to permit open-system interoperability amongst devices manufactured by different vendors.

NOTE – This specification defines the interfaces, functions and operations necessary to permit interoperability between conforming implementations. However, the designer should bear in mind that this specification is a functional description: an implementation may employ any design whose observable behavior conforms to this specification and any standards included by reference.

1.2 Purpose

IEEE 1394 is a cost-effective interconnection technology for two important groups of devices: desktop and notebook computers and their associated peripherals on the one hand and consumer electronic devices on the other. IEEE 1394 is increasingly a convergent interconnect between the two groups. However, the use of IEEE 1394 in other environments, *e.g.*, the transfer of high-speed digital video data between rooms of a house, is hampered by the lack of commercially viable architectural and protocol specifications for bridges that support the necessary quality of service. This technical specification addresses these challenges by leveraging existing protocols and physical infrastructure while at the same time preserving the core virtues of IEEE 1394 that render it desirable to product designers. The overall goal has been to produce a technically solid solution that is also pragmatic and readily deployable in order to enable a larger market for IEEE 1394 products.

1.3 Context

This technical specification is part of a larger collection of technical specifications grouped under the title, “Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable”. The documents that form the group are listed below:

Part 1: Continuous Pulsed Ultra-wideband (C-UWB) PHY

Part 2: L3 IP Bridges

Part 3: FCP and CMP over IPv4

Part 4: AV/C Relay Agents

1394 TRADE ASSOCIATION

Copyright © 2008, 1394 Trade Association. All rights reserved.

Implemented in their entirety, the technical specifications provide a comprehensive solution for networking AV devices in separate rooms of a house. The use of Internet protocol for command-and-control functions offers the possibility of extending control functionality outside the IEEE 1394 cluster to handheld or desktop Ethernet or WiFi devices. The isochronous MPLS capabilities of the L3 IP bridges, in combination with the C-UWB PHY throughput and the IEEE 802.15.3 MAC deterministic quality of service, extend IEEE 1394's high quality of service across the network. In addition, the provision for "relay agents" guarantees that deployed AV/C devices will not be stranded with little or no network connectivity.

Copyright © 2008, 1394 Trade Association. All rights reserved.

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 Technical 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 Technical 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.

[R1] 1394 Trade Association, TA Document 2004006, AV/C Digital Interface Command Set General Specification, Version 4.2

[R2] 1394 Trade Association, TA Document 2006019, Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable—Part 1: Continuous Pulsed Ultra-wideband (C-UWB) PHY

[R3] 1394 Trade Association, TA Document 2006016, Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable—Part 2: L3 IP Bridges

[R4] 1394 Trade Association, TA Document 2007004, Networking IEEE 1394 Clusters *via* UWB over Coaxial Cable—Part 4: AV/C Relay Agents

[R5] IEC 61883-1 (2003-01), Consumer audio/video equipment—Digital interface—Part 1: General

[R6] IEEE Std 1394-2008, Standard for a High Performance Serial Bus

[R7] IETF Draft-Cheshire-DNSExt-DNS-SD-04, DNS-Based Service Discovery, August 10, 2006

[R8] IETF Draft-Cheshire-DNSExt-MulticastDNS-06, Multicast DNS, August 10, 2006

[R9] IETF RFC 768, User Datagram Protocol, August 1980

[R10] IETF RFC 2734, IPv4 over IEEE 1394, December 1999

[R11] IETF RFC 2855, DHCP for IEEE 1394, June 2000

[R12] IETF RFC 3927, Dynamic Configuration of IPv4 Link-Local Addresses, May 2005

[R13] IETF RFC 5036, LDP Specification, October 2007

2.3 Reference acquisition

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

1394 Trade Association, 315 Lincoln, Suite 315, Mukilteo, Wash 98275USA; (817) 410-5750 / (817) 410-5752 (FAX); http://www.1394ta.org/

Consumer Electronics Association (CEA), 1919 South Eads St., Arlington, VA 22202, USA; (703) 907-7625 / (703) 907-7693 (FAX); http://www.ce.org/

1394 TRADE ASSOCIATION

Copyright © 2008, 1394 Trade Association. All rights reserved.

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/

International Electrotechnical Commission (IEC), Case Postale 131, 3, rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse; http://www.iec.ch/

Internet Engineering Task Force (IETF) Secretariat, c/o NeuStar, Inc., 46000 Center Oak Plaza, Sterling, VA 20166, USA; (571) 434-3500 / (571) 434-3535 (FAX); http://www.ietf.org/rfc.html

Copyright © 2008, 1394 Trade Association. All rights reserved.

3 Definitions and notation

3.1 Definitions

3.1.1 Conformance

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

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

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

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

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 and technical bulletins. A reserved object shall be zeroed or, upon development of a future specification or technical bulletin, set to a value specified by such a specification or technical bulletin. The recipient of a reserved object shall ignore its value. The recipient of an object defined by this technical specification as other than reserved shall inspect its value and reject reserved code values.

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 technical specification.

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 technical specification:

channel: A relationship that defines a group of local IEEE 1394 devices: zero or one source permitted to transmit stream packets for the channel and zero or more sinks configured to receive stream packets for the channel. A number between zero and 63, inclusive, identifies the group. Channel numbers are allocated cooperatively through isochronous resource management facilities.

controller: Within the context of FCP or FCP/IP, a device that transmits an FCP command frame to a target and receives one or more FCP response frames from the target. If FCP is the transport protocol, the controller and the target must be connected to the same IEEE 1394 cluster but if FCP/IP is used, the two devices may be part of the same cluster or may be connected to different clusters.

cycle master: On a particular IEEE 1394 cluster, the device that transmits the periodic cycle start packet 8000 times a second. In a network of interconnected clusters, there is a cycle master for each cluster.

downstream: An adjective used to describe the relationship of a node to a particular location within the network. When used in the context of cycle time synchronization with respect to the network cycle master, a downstream cycle master or a downstream port has more bridge ports between itself and the network cycle master than the port to which it is compared. Alternately, in the context of a particular isochronous path, within a bridge the downstream port is the one with more bridge ports between itself and the source.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

egress port: A description applied to a bridge port when it receives an IP datagram or IEEE 1394 stream SDU forwarded by an ingress port and transmits it over its connected medium, coaxial cable or IEEE 1394.

ingress port: A description applied to a bridge port when it receives an IP datagram or IEEE 1394 isochronous packet from its connected medium, coaxial cable or IEEE 1394, and forwards it to an egress port.

isochronous interval: a period that begins with the transmission of a cycle start packet and ends when a subaction gap is detected. During an isochronous interval, only isochronous packets may occur. An isochronous interval nominally begins every 125 µs.

isochronous resource manager: A node that implements the BUS_MANAGER_ID, BANDWIDTH_AVAILABLE, CHANNELS_AVAILABLE and BROADCAST_CHANNEL registers (some of which permit the cooperative allocation of isochronous resources). Subsequent to each bus reset, one isochronous resource manager is selected from all nodes capable of this function.

label: In MPLS terminology, the object that uniquely identifies a stream within the context of a particular link segment. For IEEE 802.15.3 coaxial cable media, the label is the 8-bit Stream Index while for IEEE 1394 media, the label is the 6-bit channel number.

legacy: An adjective applied to AV/C devices that use the FCP and CMP methods originally specified by IEC 61883-1, which are based on IEEE 1394 read, write and lock transactions. Legacy AV/C devices are not IP-capable nor can they communicate with AV/C devices connected to different IEEE 1394 clusters within the network.

local: An IEEE 1394 device is local with respect to another IEEE 1394 device if they are part of the same arbitration domain, *i.e.*, share the same root.

octet: Eight bits of data (synonymous with byte).

octlet: Eight bytes (64 bits) of data.

quadlet: Four bytes (32 bits) of data.

remote: An IEEE 1394 device is remote with respect to another IEEE 1394 device if one or more L3 IP bridges are on the route between the two devices.

sink: A device, or an application within a device, that receives isochronous stream data.

source: A device, or an application within a device, that transmits isochronous stream data.

stream: Either asynchronous or isochronous data originated by a source and received by zero or more sinks. An isochronous stream is uniquely identified by the source's EUI-64 and an index locally unique at the source. An isochronous stream's parameters include the payload, arbitration overhead and speed.

stream identifier: The concatenation of the source’s EUI-64 and the index of the source’s output plug used by the stream. Together these two data items uniquely identify a stream within the residential network.

target: Within the context of FCP or FCP/IP, a device that receives an FCP command frame from a controller and transmits one or two FCP response frames to the controller. If FCP is the transport protocol, the controller and the target must be connected to the same IEEE 1394 cluster but if FCP/IP is used, the two devices may be part of the same cluster (local) or may be connected to different clusters (remote).

upstream: An adjective used to describe the relationship of a node to a particular location within the network. When used in the context of cycle time synchronization with respect to the network cycle master, an upstream cycle master or an upstream port has fewer bridge ports between itself and the network cycle master than the port to which it is

Copyright © 2008, 1394 Trade Association. All rights reserved.

compared. Alternately, in the context of a particular isochronous path, within a bridge the upstream port is the one with fewer bridge ports between itself and the source.

3.1.3 Abbreviations and acronyms

The following abbreviations and acronyms are used in this technical specification:

CIP	Common isochronous packet (see [R5])
CRC	Cyclical redundancy checksum
DHCP	Distributed host control protocol
DNS	Domain name service
DNS-SD	DNS-based service discovery
FCP	Function control protocol (see [R5])
iPCR	Input plug control register
IRM	Isochronous resource manager (see [R6])
mDNS	Multicast DNS
OUI	Organizationally unique identifier
oMPCR	Output master plug control register
oPCR	Output plug control register
PCR	Plug control register (see [R5])
SSDP	Simple service discovery protocol
UDP	User datagram protocol (see [R9])

3.2 Notation

3.2.1 Numeric values

Decimal and hexadecimal numbers 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 ordering conventions of IEEE 1394 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 transmitted first and the least significant bit, *lsb*, is transmitted last on IEEE 1394, as illustrated below. The significance of the interior bits uniformly decreases in progression from *msb* to *lsb*.

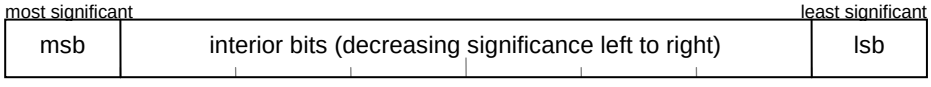

Figure 1 – IEEE 1394 bit order within a byte

Copyright © 2008, 1394 Trade Association. All rights reserved.

Within a quadlet, the most significant byte is transmitted first and the least significant byte is transmitted last on IEEE 1394, as shown below.

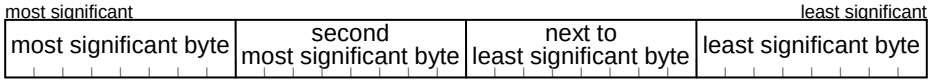

Figure 2 – IEEE 1394 byte order within a quadlet

Within an octlet, which is frequently used to contain 64-bit IEEE 1394 addresses, the most significant quadlet is transmitted first and the least significant quadlet is transmitted last on IEEE 1394, as the figure below indicates.

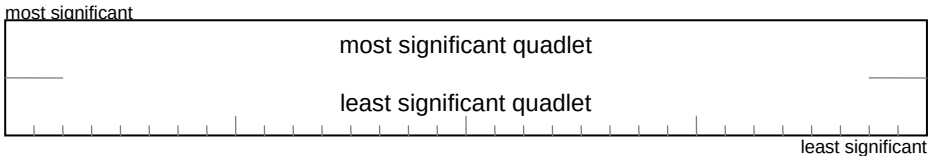

Figure 3 – IEEE 1394 quadlet order within an octlet

Without knowledge (outside of the scope of this specification) of the ordering conventions of the other bus, no assumptions can be made about the ordering (significance within a quadlet) of bytes at the unaligned beginning or fractional quadlet end of a block transfer that is not quadlet aligned or is not an integral number of quadlets.

Copyright © 2008, 1394 Trade Association. All rights reserved.

4 Overview (informative)

This document defines methods that permit AV/C devices to function as “network” devices that can discover each other and interoperate across L3 IP bridges (see [R3]). This requires protocols for service announcement and discovery, transport of commands and responses between devices and isochronous connection management procedures.

4.1 Service announcement and discovery

Consider the residential network illustrated by Figure 4. As an example, it shows IEEE 1394 clusters in three rooms, the living room, the den (home office) and a bedroom, interconnected by L3 IP bridges across a coaxial cable backbone. Each room has an AV/C controller (an HDTV in the living room and bedroom and a personal computer in the office), which will discover the other AV/C devices within its local IEEE 1394 cluster by reading each device's configuration ROM and searching for an AV/C unit directory. This is a brute force method which works adequately within a local cluster because there can be no more than 62 devices to examine, which number can be further reduced by intelligent examination of PHY self-ID packets. However, this method does not scale well over a network; not only is there potential for many more devices but the controller cannot determine how many devices are connected to a remote cluster and would have to search all possible addresses. The foregoing assumes, of course, that IEEE 1394 read and write requests cross L3 IP bridge boundaries, which they do not. For all these reasons, legacy AV/C device discovery is not suited to a network.

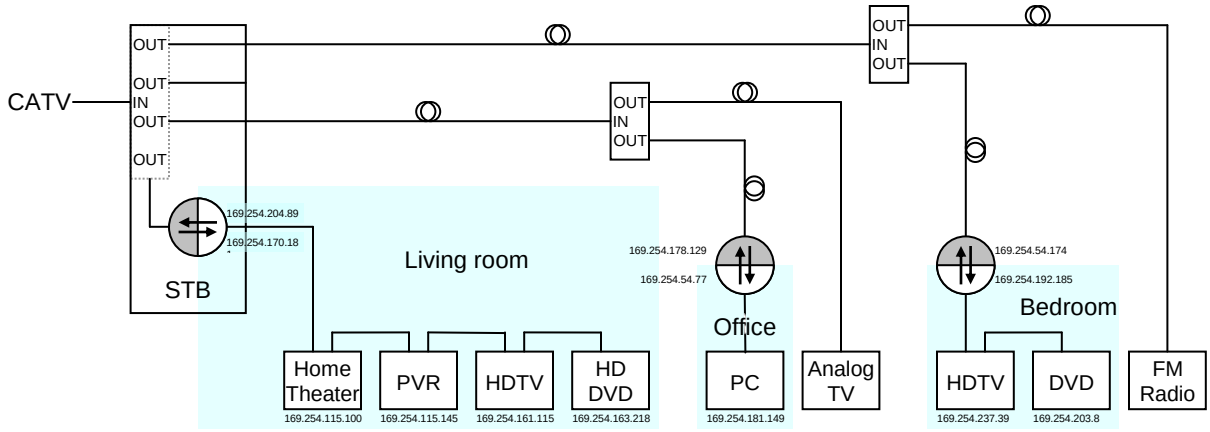

Figure 4 – Example residential network topology

A service announcement and discovery protocol suited to a network should meet several criteria:

- The protocol should be general-purpose and should leverage transport protocols already implemented in the devices, in this case Internet protocol;
- The burden on network resources should be minimized. In most cases, this means restricting device and service announcements to those instances where a previously unavailable device or service has become connected and available and restricting device and service queries to a "need to know" basis;
- An announcement or a response to a discovery request should contain both the unique identifier of the device offering the service and the routable network address used to communicate with the device; and
- The service announcement and discovery protocol should be as simple as possible.

The IETF ZeroConf working group has specified a compact service announcement and discovery protocol based upon DNS. Since many IP-capable devices already interact with DNS as clients, it is a relatively modest change to add multicast DNS (mDNS [R8]) and DNS-based service discovery (DNS-SD [R7]) to AV/C controllers and targets.

Copyright © 2008, 1394 Trade Association. All rights reserved.

To return to the example network, devices that provide services, *i.e.*, AV/C targets would multicast their unique identities, available services and associated IP address subsequent to power reset. Any AV/C controllers receiving such a multicast would update their tables of known AV/C targets and their services. An mDNS service announcement transmitted by the PVR in the living room might appear as follows:

```
pvr.local                          A    169.254.115.145
_1394ta-fcp._udp.local.            PTR  pvr._1394ta-fcp._udp.local.
pvr._1394ta-fcp._udp.local.        SRV  1010 pvr.local.
                                   TXT  txtvers=1
                                        eui64=acde48234567abcd
                                        fcp_version=1
                                        unit_specifier_ID=00a02d
                                        unit_version=010001
```

The announcement consists of a set of DNS resource records (RR).[1] Taken together, the resource records inform listening devices that, within the residential network:

- there is at least one instance of a service, `_1394ta-fcp`, transported *via* UDP/IP;
- that the service is offered on port 1010 by a device named `pvr`;
- that `pvr` has an IP address of 169.254.115.145[2];
- that `pvr`'s EUI-64 is ACDE 4823 4567 $ABCD_{16}$; and
- that the service conforms to this specification and implements the AV/C command set.

The L3 IP bridges propagate the multicast announcement to the other rooms; any AV/C controller in the living room, office or bedroom that is listening for an mDNS announcement would obtain enough information to connect to the PVR and use command set-specific methods to determine the device type and whether it is of further interest.

Subsequent to power reset or bus reset, a device multicasts, at relatively short intervals, some number of mDNS announcements. Because it may have missed AV/C target announcements that occurred prior to its power reset, an AV/C controller should multicast an mDNS query in search of AV/C targets. A typical query might take the following form:

```
QTYPE=PTR
QNAME=_1394ta-fcp._udp.local.
```

Listening AV/C targets resolve the query by transmitting—either unicast or multicast—the announcement information described above. A unicast response is transmitted directly to the AV/C controller that originated the DNS query whereas all the other AV/C controllers within a subnet can opportunistically eavesdrop on the multicast response. Whatever information is obtained by an AV/C controller is used to update the controller's DNS cache.

Interoperability requires that all AV/C devices implement a common service announcement and discovery protocol. Because mDNS and DNS-SD are of modest complexity and implementation burden, they were chosen as that common protocol. However, this does not preclude implementation of alternate service announcement and discovery methods, *e.g.*, Simple Service Discovery Protocol (SSDP), necessary for conformance to external profiles outside the scope of this specification.

[1] The text does not show the actual format of the RR; it has been simplified to show only pertinent information for the different RR types. The normative formats of domain names, DNS RRs and DNS queries can be found in [B16] and [B17].

[2] The IP address shown in example happens to be a dynamically assigned link-local address, but it could just as readily be an address assigned to the AV/C target by a DHCP server. The source of an AV/C device's IP address has no effect on the operations of mDNS and DNS-SD.

Copyright © 2008, 1394 Trade Association. All rights reserved.

4.2 FCP and FCP over IPv4 (FCP/IP) transaction sequences

Legacy AV/C devices communicate with each other *via* Function Control Protocol (FCP), a single-threaded data transport protocol that permits command and response frames to be transferred between a controller and a target connected to the same IEEE 1394 cluster. Figure 5 illustrates an FCP transaction.[3]

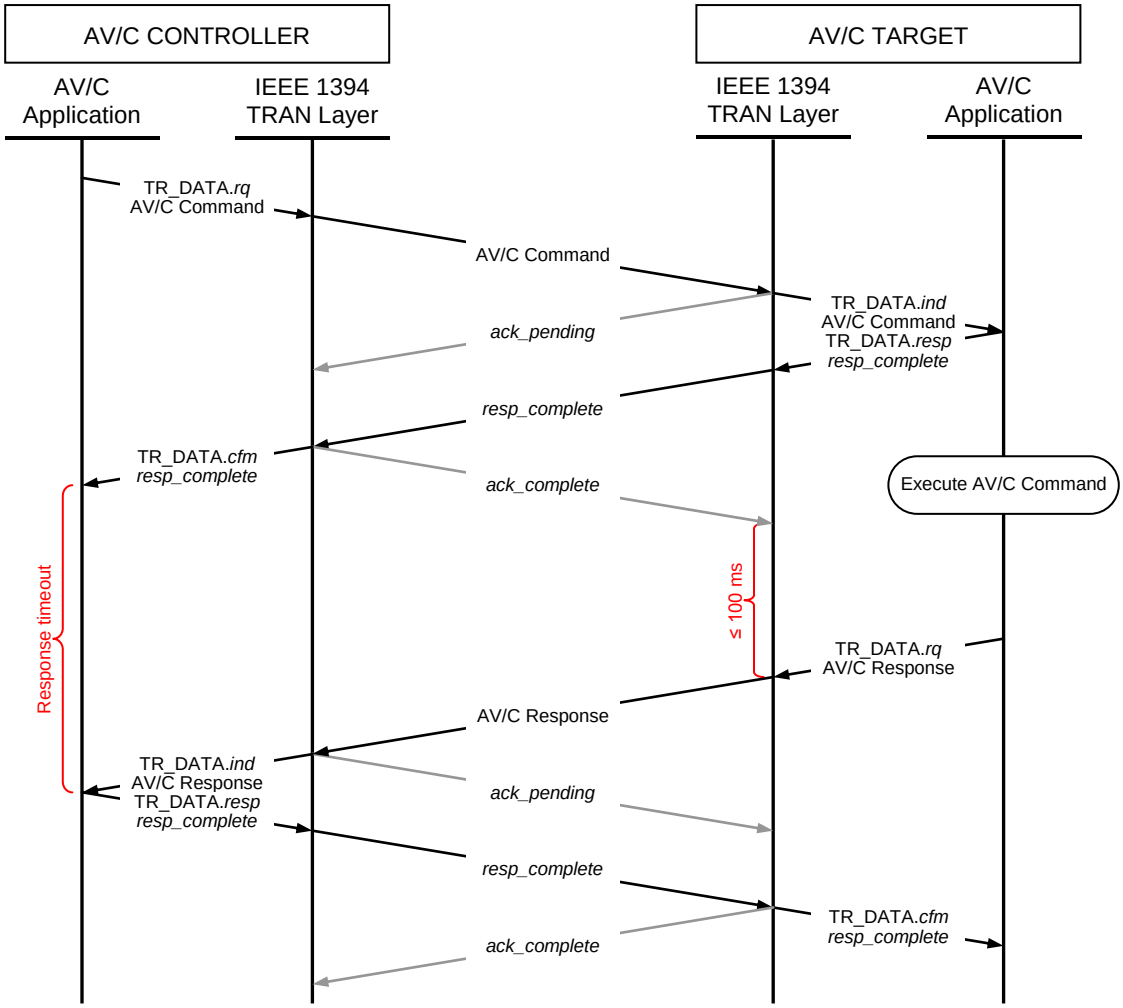

Figure 5 – FCP command/response transaction (IEEE 1394 split transactions)

FCP, as defined by [R5], uses IEEE 1394 block write requests to transport data. A controller initiates operations by requesting its IEEE 1394 transaction layer to transmit a block write request that contains a command frame to the target's well-known "mailbox" address for command frames, FFFF F000 $0B00_{16}$. Data written to this address causes the target's IEEE 1394 transaction layer to interrupt the AV/C application code to indicate receipt of a command frame, which the AV/C application code immediately confirms by signaling *resp_complete* to the transaction layer. This, in turn, causes the target's transaction layer to transmit a write response packet to the controller. When the controller's AV/C application code receives transaction completion confirmation, it starts a timeout period to await the response frame from the target. In the meantime, the target executes the command and, upon completion, requests its IEEE 1394 transaction layer to transmit the response frame as a block write request packet to the controller's well-known "mailbox" address for response frames, FFFF F000 $0D00_{16}$. This causes an interrupt that signals receipt

[3] Although the example shows split transactions, concatenated or unified transactions are permissible. (see [R6])

Copyright © 2008, 1394 Trade Association. All rights reserved.

of a response frame, at which time the controller concludes the command/response transaction with a write response packet of *resp_complete*.

Note two critical intervals, shown in red. The first is a 100 ms time limit within which the target shall transmit either an interim or final response frame. This interval is measured at the target's cable/PHY interface and starts when the target receives acknowledgment[4] of the write response packet transmitted to the controller to confirm receipt of the command frame. The second is a time limit (of a duration unspecified by [R1]) within which the target is expected to return a response frame. If the response timeout expires without receipt of a response frame, the controller institutes error recovery action.

Since IEEE 1394 transactions do not cross L3 IP bridges, an alternative transport for command and response frames, based upon Internet protocol, is specified by this document. This alternate, FCP over IPv4 (or FCP/IP for short) has a command/ response transaction structure intentionally analogous to FCP, as is evident in Figure 6.

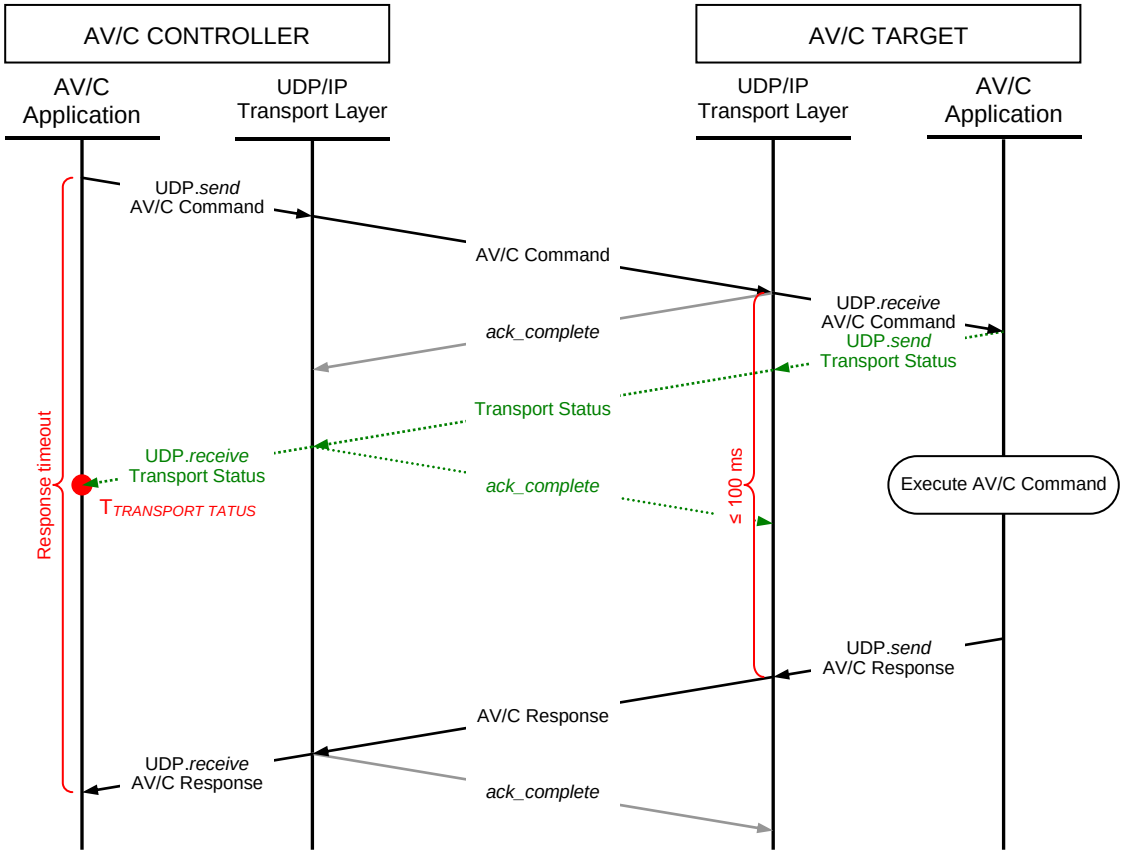

Figure 6 – FCP/IP command/response sequence

In lieu of IEEE 1394 block write requests, FCP/IP uses messages transported *via* user datagram protocol (UDP [R9]) over IP to deliver FCP command and response frames. The UDP transport layer interposes itself between the AV/C application and the IEEE 1394 transaction layer (not shown). An important consequence is that the AV/C application does not receive detailed information about IEEE 1394 transaction layer operations because these are processed by the UDP transport layer. A controller initiates operations by requesting its UDP transport layer to transmit a

[4] An AV/C target that uses unified transactions starts timing this interval upon receipt of the command frame, as the reference point used in the case of split transactions does not exist. The AV/C protocol would be more consistent if the same reference point were used in both cases.

Copyright © 2008, 1394 Trade Association. All rights reserved.

command frame within a UDP datagram to the target's IP unicast FIFO (see [R9]). Datagrams written to this FIFO are acknowledged and delivered to the target's AV/C application.

At this point, the analogy between native IEEE 1394 FCP and FCP/IP becomes somewhat strained. Because the UDP/IP transport layer manages recoverable link layer errors (*ack_busy_X*, *ack_busy_A*, *ack_busy_B*, *ack_conflict_error*, *ack_data_error* and *ack_tardy*), there is neither need nor possibility for the controller to manage these errors. However, a legacy AV/C target uses either a) repeated busy acknowledgments[5] or b) a transaction completion response of *resp_conflict_error* to signal that the device is busy and cannot accept another command. The optional FCP Transport Status data unit, shown in green with dotted lines in Figure 6, permits the target to communicate its busy condition to the controller. Unless response timeout considerations predominate, a target should not transmit a Transport Status data unit except to report an error.

There are additional consequences to the AV/C application's disconnection from the IEEE 1394 transaction layer confirmations: the lack of analogous reference points for the start of timeout intervals by the controller and target. In the case of the target's 100 ms response frame time limit, the difference is negligible and may be ignored. FCP/IP measures this time interval at the cable/PHY interface starting when the command frame datagram is received.

With respect to the controller's response timeout, the only starting reference point is the transmission of the command frame datagram. If both controller and target are locally connected, the difference between FCP and FCP/IP reference points is negligible and may be ignored. However, if controller and target are remote with respect to each other, the traversal of L3 IP bridges and the coaxial cable backbone might introduce delays that are a significant fraction of the controller's response timeout. Two approaches can mitigate this problem.

First, and most importantly, a controller's response timeout should be at least 200 ms; this is short enough to preserve system responsiveness and long enough to virtually eliminate unnecessary response timeouts and subsequent error recovery.

An additional strategy is to cause the controller's receipt of a Transport Status response, shown at time $T_{TRANSPORT\ STATUS}$, to reset the response timeout to its original value. This is less useful in practice than it sounds in theory. If a target executes a command promptly, there will be little or no time difference between the controller's receipt of the Transport Status response and its receipt of the response frame. In fact, if both FC PDUs are bundled within the same datagram, there will be no difference. The Transport Status response serves a useful purpose only in cases where the target consumes a significant part of the 100 ms time limit before it transmits a response frame. The benefit of "buying some time" to extend the controller's response timeout accrues only to those cases for which the sum of target delay and transport delay is greater than the unmodified response timeout. Targets should not transmit a Transport Status FC PDU except to report an error or if more than 80 ms are expected to elapse before transmission of a response frame.

The FCP/IP command/response transaction concludes when the controller receives a datagram containing the target's response frame. Just as it does for delivery of the command frame to the target, the UDP/IP transport layer manages any recoverable link errors that occur in the delivery of the response frame to the controller. Because a controller is expected to have sufficient resources to receive a response frame at any time, there is no necessity for the controller to transmit a Transport Status response to the target.

4.3 Connection Management Procedures over IPv4 (CMP/IP)

The Connection Management Procedures (CMP) specified by [R5] were architected for use within a single IEEE 1394 cluster and are unable to manage isochronous connections between devices in remote clusters. Supplemental procedures (CMP/IP) have been designed for isochronous connection management in a network environment.

[5] Persistently repeated busy acknowledgments eventually force a TR_DATA.*confirmation* request status of RETRY LIMIT at the controller. When FCP/IP is in use, this information will not reach the controller's AV/C application and the target must additionally transmit a Transport Status response to indicate the busy condition.

1394 TRADE ASSOCIATION Copyright © 2008, 1394 Trade Association. All rights reserved.

The principal reasons that CMP does not function across L3 IP bridges have less to do with the fact that the bridges do not transport IEEE 1394 read, write and lock transactions and more to do with the lack of notification of IEEE 1394 bus reset and, even if bus reset notification were provided, the timeliness of response. Bus reset is a fundamental part of CMP: all devices perceive it simultaneously and respond to it promptly. Bus reset obliterates all broadcast and point-to-point connections by zeroing the connection count fields in the output and input plug control registers. Bus reset releases all previously reserved isochronous resources. Each controller responsible for a connection between a source and sink device has one second to set things right, to reallocate previously allocated isochronous resources and to reestablish point-to-point or broadcast connections by lock operations to the various plug control registers. There is a flurry of activity within an IEEE 1394 cluster immediately after bus reset; it involves numerous individual read and lock transactions A protocol to encapsulate these IEEE 1394 transactions into a UDP/IP datagram could have been developed and probably could have met the generous 100 ms split-transaction timeout limits specified by IEEE 1394. Although individual transactions might complete within the time limits, taken together they would be terribly inefficient: the remotely located controllers would have slim probability of completing all the necessary housekeeping in less than one second after a bus reset.

However, the primary reason that precludes CMP from functioning across bridges—not only L3 IP bridges, but any sort of bridge—is that it is neither desirable nor practical to signal bus reset to the whole network when it occurs in a single IEEE 1394 cluster. This matter was studied extensively during the development of [B10] with the conclusion that any attempt to propagate bus reset throughout the network would result in self-perpetuating bus reset "storms" that would render the network useless.

Although CMP is restricted to a single cluster and cannot work in a network environment, its fundamental architecture is sound within the single cluster domain for which it was designed. Consequently, CMP/IP procedures, specified in [R3] and used by devices specified by this document, leverage CMP and make use of its design within each local cluster. The important difference is that the remotely located controller does not reallocate isochronous resources and reestablish connections subsequent to bus reset, but that the bridge port, as the controller's representative, performs these chores in a timely fashion after bus reset. In order to do this, the bridge requires information about the isochronous path's endpoints (connections) and isochronous resources utilized (channel and bandwidth within the IEEE 1394 cluster). CMP/IP provides a method for controllers to communicate this information to the L3 IP bridges at the same time the path's route is established and the necessary isochronous resources allocated along the way.

CMP/IP is accomplished by messages transmitted *via* Internet protocol; there is a message to create a path between a source device and a sink device (PATH REQUEST), a message to tear down a previously created path (PATH TEARDOWN) and a message to confirm the success or failure of either operation (PATH STATUS). CMP/IP is necessary in all of the following circumstances:

- The controller, source and sink are each connected to a different IEEE 1394 cluster;
- The controller and source are connected to the same cluster and the sink is connected to a different cluster;
- The controller and sink are connected to the same cluster and the source is connected to a different cluster; or
- The source and sink are connected to the same cluster and the controller is connected to a different cluster.

In other words, when at least one of the three device functions is connected to a different IEEE 1394 cluster than the other two, use CMP/IP in lieu of CMP. Contrariwise, when all three device functions are connected to the same cluster, CMP should be used.[6] This requires a controller to differentiate between local and remote devices, which is easily done if the controller maintains an inventory, organized by EUI-64, of all local devices and infers that other devices are remote.

Annex E contains examples, illustrated by message sequence charts (MSCs), of typical path establishment and path teardown operations. This is the same example that appears as an informative annex in [R3].

[6] It is possible, but not recommended, to use CMP/IP when all three device functions are connected to the same cluster if and only if an L3 IP bridge is connected to the same cluster. If no bridge is present, CMP is the only method available.

Copyright © 2008, 1394 Trade Association. All rights reserved.

5 Data structure formats

5.1 DNS messages and resource records

DNS is normatively specified by [B17], which should be consulted for more detailed information than is presented in the clauses that follow. The material that is reproduced here is intended to be beneficial to the reader's understanding of [R8] and [R7].

5.1.1 DNS message

All DNS communications share a common message format. A DNS message is divided into five parseable sections: the Header, Question, Answer, Authority and Additional Data sections. Except for the Header section, all the other sections, in certain circumstances, may be omitted from the message.

The Question section may be empty or may contain one or more entries, each of which consists of three parameters: QNAME, QTYPE and QCLASS. Together, each entry's parameters specify information sought from the DNS. QNAME shall specify a domain name. QTYPE shall specify the resource record TYPE that is eligible for return if its NAME field satisfies the QNAME query; a QTYPE value of FF_{16} (T_ANY) requests that matching resource records of any type be returned. QUEUE CLASS shall be IN, the Internet.

The remaining three sections, Answer, Authority and Additional Data contain zero or more resource records.

5.1.2 DNS resource records

DNS resource records (RR) share a common format specified by [B17] and reproduced in Figure 7 for the convenience of the reader. There are no alignment requirements for any of the fields of an RR.

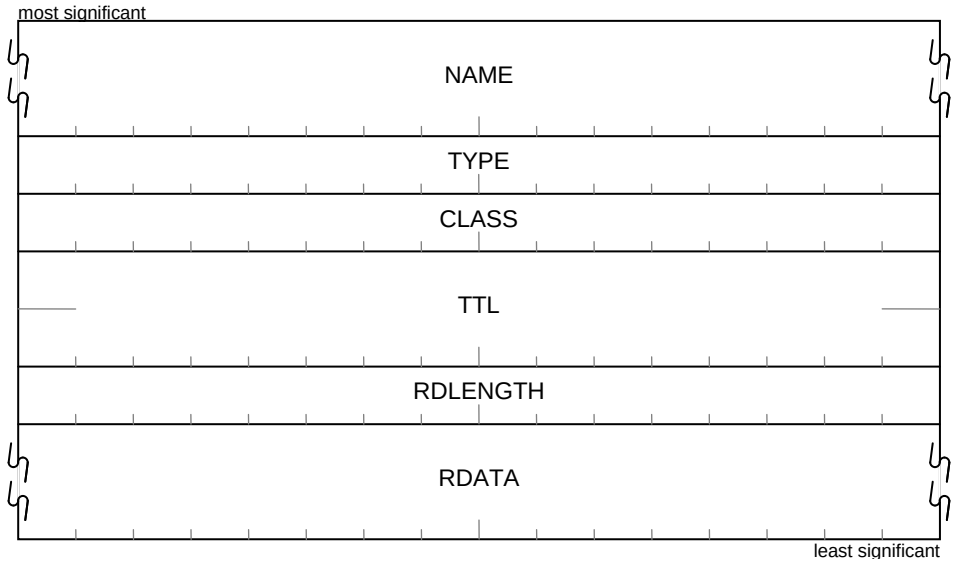

Figure 7 – DNS resource record (RR) message format

The variable-length NAME field shall contain the domain name to which the resource record pertains.

The value of the TYPE field shall be equal to one of the type codes enumerated in the table below:

Copyright © 2008, 1394 Trade Association. All rights reserved.

TYPE	Value	Description
A	1	Host address record
PTR	12	Domain name pointer record
TXT	16	Text strings record
SRV	33	Server record

The value of the CLASS field shall be one to indicate that the network class is Internet.

The value of the 32-bit TTL field shall be a signed integer that specifies the duration, in seconds, that the information in the resource record may be cached.

The RDLENGTH field shall specify the length, in octets, of the RDATA field.

The variable-length RDATA field is a string of octets that describes the resource. The format of the RDATA is determined by the value of the CLASS and TYPE fields.

5.1.2.1 Host address (A) resource record

The NAME field of an A record shall contain the AV/C target's host name obtained *via* mDNS or a DHCP server. The RDATA field shall contain the target's 32-bit IPv4 address.

5.1.2.2 Pointer (PTR) resource record

The NAME field of a PTR record shall contain the service name `_1394ta-fcp._udp.local`.

The RDATA field of a PTR record shall contain a service instance name. There shall be a SRV record for the specified service instance name.

5.1.2.3 Server (SRV) resource record

The NAME field of an SRV record shall contain the service name `_1394ta-fcp._udp.local` prepended by an instance name. The combined service instance name identifies an AV/C target instantiation of the FCP/IP service specified by this document. For example, `pvr_.1394ta-fcp._udp.local` could describe a PVR's instantiation of the service.

The RDATA field of an SRV record shall contain the variable-length character string subfields designated Priority, Weight, Port and Target.

The Priority and Weight subfields shall be equal to zero. SRV records, as used by FCP/IP, shall not determine which AV/C target instance of the service is selected by an AV/C controller. That choice shall depend upon the target's unit and subunit characteristics, as determined by command set-dependent means beyond the scope of this document. Nonzero values for these subfields would preempt the AV/C controller's ability to use its own selection criteria.

The Port subfield shall specify the AV/C target port number, encoded as a 16-bit unsigned integer in network byte order, to which UDP/IP datagrams containing FC PDUs should be addressed. If an AV/C target implements more than one unit, each shall use a different port number.

The Target subfield shall specify the AV/C target's host name. There shall be at least one address record for the specified host name. When responding to an mDNS query, the target should return its address record in the additional data section.

Copyright © 2008, 1394 Trade Association. All rights reserved.

5.1.2.4 Text (TXT) resource record

The NAME field of a TXT record shall contain, explicitly or by inheritance from the preceding RR, the service instance name of the SRV record to which it pertains.

The RDATA field of a TXT record shall contain the parameters enumerated in the table below. The parameters shall be encoded as length-delimited character strings consisting of the parameter name separated from the parameter value by an equal sign.

Parameter	Value
`txtvers`	The `txtvers` parameter shall be equal to UTF-8 "1" and shall be the first character string in the RDATA field. This indicates that the format of the TXT record conforms to [R7], *i.e.*, the RDATA contains only length-delimited character strings.
`eui64` [a]	The `eui64` parameter shall be equal to the value of the EUI-64 of the AV/C target identified by the Target subfield in the SRV record to which this TXT record pertains.
`fcp_version`	The `fcp_version` parameter shall be equal to UTF-8 "1" to indicate that the AV/C target conforms to this specification.
`model_ID` [a]	The optional `model_ID` parameter shall be equal to the 24-bit value of the Model_ID entry, if present, in the AV/C target's root directory or AV/C unit directory.
`model_descr`	The optional `model_descr` parameter shall be a Unicode string equivalent to the textual descriptor, if present, that immediately follows the Model_ID entry. UTF-8 is the recommended encoding.
`unit_specifier_ID` [a]	The `unit_specifier_ID` parameter shall be equal to the 24-bit value $00A02D_{16}$.
`unit_version` [a]	The `unit_version` parameter shall be equal to the 24-bit value 010001_{16}.
`utf_encoding`	The `utf_encoding` parameter shall be equal to UTF-8 "8", "16" or "32", which specifies the Unicode encoding, UTF-8, UTF-16 or UTF-32, respectively, of any `model_descr` or `vendor_descr` parameter that follows in the left-to-right RDATA parse order. More than one `utf_encoding` parameter may be present within a TXT record; if omitted, the default is UTF-8 encoding.
`vendor_ID` [a]	The `vendor_ID` parameter shall be equal to the 24-bit value of the Vendor_ID entry in the AV/C target's root directory.
`vendor_descr`	The optional `vendor_descr` parameter shall be a Unicode string equivalent to the textual descriptor, if present, that immediately follows the Vendor_ID entry. UTF-8 is the recommended encoding.

[a] The parameter's value shall be encoded as a sequence of UTF-8 characters, each of which represents a hexadecimal digit that corresponds to four bits of the parameter's numeric value.

5.2 Function control protocol data unit (FC PDU)

Individual FCP command and response frames shall be encapsulated within a protocol data unit whose format conforms to Figure 8. The FC PDU format also permits a controller to initiate "tunneled" IEEE 1394 read, write or lock requests addressed to a target; the response format enables a target to communicate the transport status of the command frame or request most recently transmitted to the target. FC PDUs are transmitted *via* UDP/IP; an IP datagram may contain more than one FC PDU addressed to the same recipient.

Copyright © 2008, 1394 Trade Association. All rights reserved.

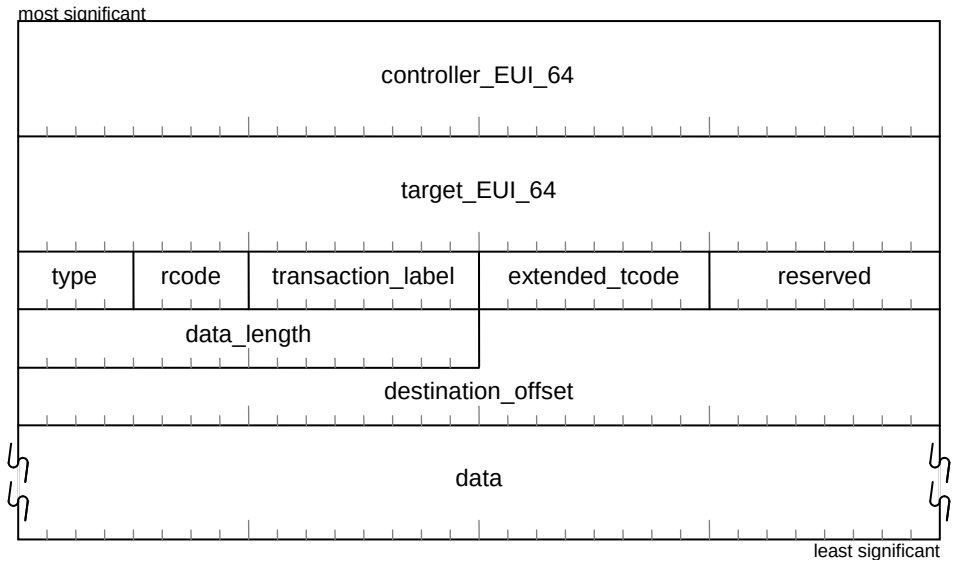

Figure 8 – FCP/IP protocol data unit format

The value of the *controller_EUI_64* field shall be equal to the EUI-64 reported by the controller's configuration ROM bus information block.

The value of the *target_EUI_64* field shall be equal to the EUI-64 reported by the target's configuration ROM bus information block.

NOTE – When relay agents (see [R4]) participate in the FCP/IP command/response exchange, the values of the *controller_EUI_64* and *target_EUI_64* do not necessarily identify the IP host that originates FCP command frames or the IP host that originates FCP response frames, respectively.

The *type* field shall specify the type of PDU, as specified by the table below:

type	**Description**
0	The *data* field contains an FCP command frame.
1	The *data* field contains an FCP response frame
2	Request the target to read its own memory and return the information in the *data* field.
3	Request the target to write the information in the *data* field to its own memory.
4	Request the target to perform the lock operation specified by *extended_tcode*; the *data* field contains the arguments for the lock operation.
5	The *rcode* field contains transport status information for the request identified by *transaction_label* and the *data* field may contain response data.
$6 – F_{16}$	Reserved for future standardization.

The value of the *rcode* field is meaningful only when the *type* field is equal to five, in which case it shall indicate the transport status of the most recent FCP command frame or request transmitted by the controller, as encoded by the table below.

Copyright © 2008, 1394 Trade Association. All rights reserved.

rcode	Name	Description
0	Complete	The most recent attempt to deliver an FCP command frame or request to the target succeeded.
1 – 3		Reserved for future standardization.
4	Conflict error	A resource conflict prevented the target from accepting the FCP command frame or request; the controller may resend the PDU.
5	Data error	A data CRC or hardware error prevented the target from accepting the FCP command frame or request; the controller may resend the PDU.
6	Type error	The *type* field in the FC PDU was set to a reserved value.
$7 – F_{16}$		Reserved for future standardization

The controller determines the value of the *transaction_label* field, which shall uniquely identify each request outstanding at a target. The value of *transaction_label* need not be globally unique but shall be unique within the context of the target. The target shall copy the *transaction_label* field from a received FC PDU and shall return it in the corresponding response frame or transport status FC PDUs. So long as a command or request identified by a particular value of *transaction_label* remains outstanding at a particular target, the controller shall not use the same *transaction_label* value in an FC PDU transmitted to the same target. A command is outstanding until the controller receives the corresponding frame from the target or the controller's response timeout expires.

The value of the *extended_tcode* field is meaningful when the FC PDU specifies a lock operation, in which case the meaning and usage of *extended_tcode* are as specified by [R6].

The *data_length* field shall specify the size, in bytes, of the *data* field. The value of *data_length* shall not include pad bytes, if any, at the end of the *data* field.

The value of the destination_offset field is meaningful when the FC PDU specifies a read, write or lock request, in which case the field shall contain the least significant 48 bits of the destination address of the request.

When *data_length* is zero, the *data* field shall be omitted from the FC PDU.

5.3 Path management messages

FCP/IP controllers and targets use a subset of the path management messages defined by [R3]; the table below enumerates the path management messages and identifies whether FCP/IP controllers and targets originate (transmit) or process (receive) individual message types.

		Controller		Target	
Message	***message_type***	**Transmit**	**Receive**	**Transmit**	**Receive**
PATH MAPPING	$3E40_{16}$	Not applicable		Not applicable	
PATH REQUEST	$3E41_{16}$	Mandatory	Not applicable		
PATH TEARDOWN	$3E42_{16}$	Mandatory	Not applicable		
PATH NOTIFICATION	$3E43_{16}$	Not applicable	Mandatory		
TIME OFFSET	$3E50_{16}$	Optional			

NOTE – Although FCP/IP controller and target are distinct functional modes, they may coexist within a single device.

Copyright © 2008, 1394 Trade Association. All rights reserved.

6 Operations

6.1 Service announcement and discovery

6.1.1 Link-local address assignment

Although obtaining an IP address is not, strictly speaking, part of service announcement and discovery, it is a prerequisite for participating in the service announcement and discovery protocols.

When a DHCP server is unavailable within the residential network, devices shall use the method specified by [R12] to claim and defend an IP address unique within the domain of the residential network. [B24] provides guidance for the detection of a DHCP server.

[R12] specifies that initial selection of an IP address rely on a pseudorandom number generator whose output has uniform distribution in the range 169.254.1.0 through 169.254.255.255, inclusive. This specification additionally requires the device's EUI-64 be used as the initial seed value for the pseudorandom number generator.

Once a device successfully claims and defends a link-local IP address in the 169.254/16 address family, the device should store it in nonvolatile memory. If the IP address is subsequently abandoned and a new address claimed, the new address should replace the old in nonvolatile memory. Subsequent to power reset, a device with nonvolatile memory should attempt to claim its most recently assigned address. Otherwise, the device shall seed its pseudorandom number generator with its EUI-64 and generate an IP address to claim and defend.

6.1.2 Host name assignment

Once a device has successfully obtained an IP address, it shall use the method specified by [R8] to claim and defend a host name. Although the choice of a host name is implementation-dependent, a useful host name should meet the following criteria:

- The host name will appear in menus for selection by a human user but will be typed rarely, if ever, by a user. As a consequence, a host name should be neither cryptic nor terse and should be both readable by and intelligible to a human user;
- The host name should be descriptive of the device, *i.e.*, it should not be a challenge to match a host name with a physical device; and
- The preferred host name should have a low probability of collision with another device's host name. The device should construct the preferred host name from the device manufacturer’s name and the device's model number. For example, “Acme-DVP-NS71HP” might be the host name for a particular model DVD player manufactured by Acme Consumer Electronics.

If the preferred host name conflicts with the same name already claimed by another device, the unsuccessful claimant should add a suffix to the host name, *e.g.*, "-1", and attempt to claim the modified host name.

NOTE – [B9] defines "writable" configuration ROM, a facility that could provide a standard method to establish or change a user-modifiable "nickname", *i.e.*, a locally unique host name. Devices with a rich user interface, *e.g.*, an HDTV, could present an interface that would enable a user to modify the configuration ROM of another device that lacks a user interface, *e.g.*, an STB.

6.1.3 Service announcement

Initial service announcement occurs after probes have determined that none of the names in the device's unique[7] RR set conflict with names claimed in other devices’ unique RR sets. Once this prerequisite is satisfied, the device shall

[7] Address (A), SRV and TXT records are part of a device's unique RR set, while PTR records are part of its shared RR set.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

multicast a gratuitous[8] mDNS response that contains all of its RRs in the response's Answer section. The device shall transmit least two gratuitous responses, one second apart, and may send up to eight gratuitous responses, subject to the requirement that the interval between responses doubles for each response after the second.

Once the initial announcement completes, the device shall not transmit additional gratuitous mDNS responses unless it undergoes power reset, information in any of its RRs changes or cluster topology changes (*i.e.*, a device is disconnected from or connected to the cluster). In any of these cases, the device shall repeat its initial service announcement as described above.

6.1.4 Service discovery

A device requests information about all instances of a particular service within the residential network by multicasting an mDNS query whose Question section contains the service name.

In the case of AV/C controllers seeking AV/C targets, the QNAME field shall specify `_1394ta-fcp._udp.local` and the QTYPE field shall be equal to PTR. This causes all AV/C targets to return, either by multicast or unicast, a DNS response record whose Answer section contains the PTR record that satisfies the query and whose Additional Data section contains the associated A, SRV and TXT records that describe the service instance's IP address and port number. The AV/C controller may connect to the FCP/IP service and use AV/C commands to determine the target's unit and subunit information.

6.2 AV/C transactions *via* FCP/IP

An AV/C transaction consists of a command frame that originates from an AV/C controller and is transmitted to a single AV/C target or broadcast to all AV/C targets, followed by the controller's receipt of one or more response frames returned by the target or targets. FC PDUs (see 5.1) that contain AV/C command or response frames shall be transmitted as an IP datagram *via* UDP/IP.

Command frame FC PDUs transmitted by an AV/C controller shall be addressed to the AV/C target IP address and port number announced by the target in an mDNS response message. Response frame and Transport Status FC PDUs transmitted by an AV/C target shall be addressed to the source IP address and port number obtained from the header of the IP datagram that contained the FC PDU.

An important difference between FCP/IP and FCP is that bus reset does not affect the status of any active or pending FCP/IP transactions. This is because the IP addresses used by FCP/IP are stable across bus reset, unlike the volatile IEEE 1394 node IDs used by FCP.

6.2.1 AV/C controller operations *via* FCP/IP

An AV/C controller initiates an FCP/IP transaction with a particular AV/C target by transmitting a unicast UDP/IP datagram whose payload is contains one or more command frame FC PDUs (see 5.1) intended for the same target. Upon the datagram's transmission, the controller shall start a response timeout interval of at least 200 ms, during which it shall await receipt of a response frame or Transport Status response from the target.

Once a particular *transaction_label* value has been used in an FCP/IP command frame, the controller shall not reuse that *transaction_label* value until a final response is received for the transaction.

> NOTE – Multiple FC PDUs intended for the same AV/C target may be transmitted within a single UDP/IP datagram. The execution order of command frame FC PDUs shall be determined by their order within the datagram. Because UDP/IP does not guarantee in-order delivery, aggregating FC PDUs within a single datagram is the only reliable method to preserve order if more than one FC PDU is to be queued at a target.

[8] The mDNS response is termed "gratuitous" because it is not a reply to an mDNS query, *i.e.*, its Question section is empty.

Copyright © 2008, 1394 Trade Association. All rights reserved.

An AV/C controller that receives a Transport Status response whose *rcode* field is equal to four, conflict error, shall not retransmit the pertinent command frame FC PDU until all pending FCP/IP transactions with that target have completed or timed out.

An AV/C controller that receives a Transport Status response whose *rcode* field is zero shall reset its response timeout to 200 ms and continue countdown of the timeout period.

When an AV/C controller receives a response frame FC PDU from a target, it shall stop the response timeout timer for the transaction. If a final response was received, the FCP/IP transaction identified by *transaction_label* is complete and the value of *transaction_label* may be reused in a subsequent transaction. Otherwise, when an interim response is received, the response timeout timer is canceled but the transaction remains pending. The *transaction_label* value associated with the transaction shall not be reused until a final response concludes the transaction.

NOTE – Because UDP does not provide guaranteed transport, a final response frame transmitted by a target might fail to be received by the controller. In order to prevent indefinite waits by controllers, [R1] recommends that unit specifications define a maximum time limit for completion of CONTROL commands, define an appropriate STATUS command usable to determine the execution state of the outstanding CONTROL command, define a CONTROL command that deterministically cancels execution of the outstanding CONTROL command or define all three. AV/C controllers should base their determination of command timeout and subsequent error recovery upon these methods.

If the response timeout for any pending FCP/IP transaction expires, the AV/C controller should take error recovery action as specified by [R1].

6.2.2 AV/C target operations *via* FCP/IP

When an idle AV/C target receives a command frame FC PDU, it becomes busy executing the command and might be unable to queue additional command frame FC PDUs. A legacy AV/C target would communicate its busy condition to the controller by returning *ack_busy_X*, *ack_busy_A* or *ack_busy_B* to any controller that attempted to queue an additional command. This L2 signaling method is unavailable to an AV/C target when it uses FCP/IP. Instead, the target must receive the command frame FC PDU in order to obtain the source address and port number of the controller in order to signal busy by transmitting a Transport Status response to the controller. Consequently, the target should be implemented with sufficient buffer space to have room always to receive an unexpected FC PDU; this nominally requires buffer space for two FC PDUs, one for the active command frame and one to hold an incoming FC PDU long enough to reject it.

If a command frame FC PDU arrives while the target is busy executing an AV/C command, it shall transmit a UDP/IP datagram that contains a Transport Status response whose *rcode* field is equal to four, conflict error. The Transport Status datagram shall be addressed to the source IP address and source port number obtained from the IP datagram that contains the rejected command frame FC PDU.

Otherwise, the target shall validate the format of the AV/C command just received. If there is an error in the command frame, the target shall promptly transmit a NOT IMPLEMENTED or REJECTED response frame to the AV/C controller. Otherwise, the target shall determine whether the command requires an immediate or deferred transaction. If the AV/C command requires a deferred transaction, the target shall promptly transmit an interim response frame to the controller before starting to execute the command. Otherwise, the target should promptly execute the command and promptly return a final response frame. If 80 ms elapse from receipt of the command without completion or error, the target shall transmit a Transport Status response to extend the controller's timeout period by 200 ms.

Although the time interval between an AV/C target's transmission of an interim response frame FC PDU to the AV/C controller and the target's subsequent transmission of a final response frame FC PDU to the controller is unspecified, a target that transmits an interim response shall eventually transmit a final response to complete the transaction.

Copyright © 2008, 1394 Trade Association. All rights reserved.

6.2.3 AV/C transaction sequences *via* FCP/IP

This clause illustrates the common FCP/IP transaction sequences that occur when AV/C devices follow the requirements of 6.2.1 and 6.2.2. The illustrations are simplified to show only the transmission of FC PDUs; link-layer acknowledgments are omitted.

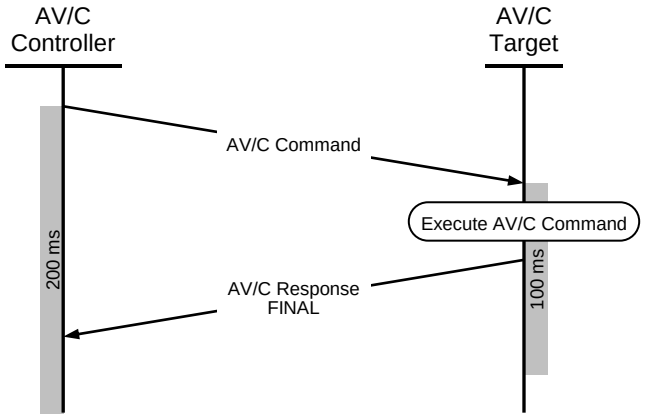

Figure 9 – AV/C immediate transaction

Figure 9 represents an immediate AV/C transaction, in which the target is able to complete command execution well within the 100 ms allowed and then to return a final response frame that reaches the controller well before its 200 ms response timeout expires.

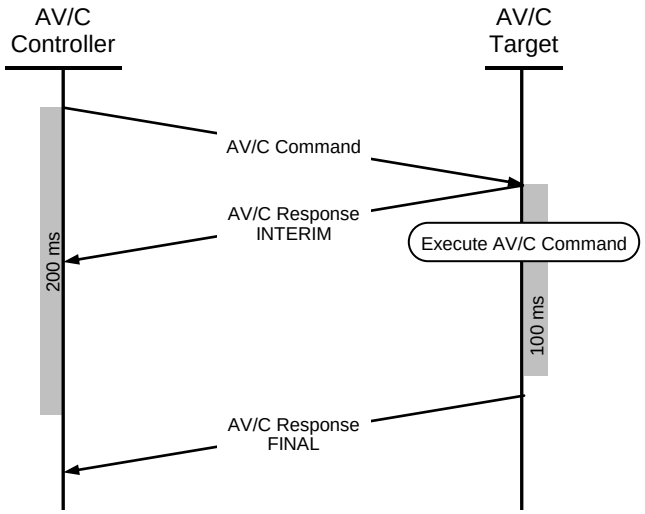

Figure 10 – AV/C deferred transaction

Figure 10 illustrates a deferred AV/C transaction. The AV/C target knows, by design, that the command it has received requires more than 100 ms to complete execution. Consequently, the target immediately transmits an interim response to the AV/C controller. Receipt of the interim response satisfies the controller's 200 ms response timeout, which is canceled and not restarted. Meanwhile, the target continues to execute the command; no time limit is placed upon its completion. Although the controller has stopped its response timeout timer, it shall not reuse the *transaction_label* that identifies the transaction because the transaction is not yet complete. Eventually, the target completes command execution and transmits a final response frame to the controller; this completes the transaction and *transaction_label* may be reused. Notice that it is possible, perhaps even probable, for the controller to receive target's final response frame long after the 200 ms response timeout would have expired.

Copyright © 2008, 1394 Trade Association. All rights reserved.

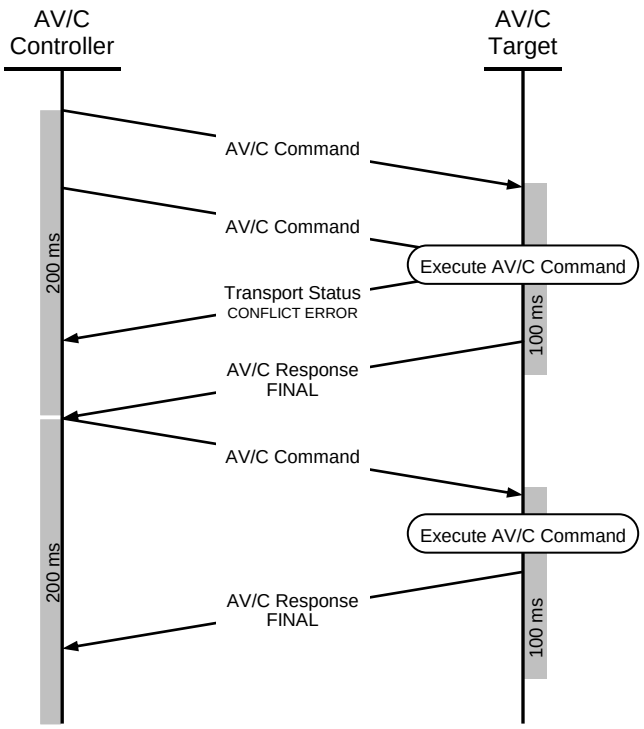

Figure 11 – AV/C target busy with AV/C controller retry

Figure 11 shows how the Transport Status response is used when an AV/C controller transmits a second command frame while the AV/C target is busy with the first command. Neither command uses a deferred transaction. The target executes the first command immediately, but before it can complete the command, another command frame is received. The target is busy and cannot queue another command frame, so it returns a Transport Status response with an *rcode* of CONFLICT ERROR. The controller will not retry the refused command until receipt of a final response frame indicates that the target is ready to accept a new command. The controller retransmits the formerly refused command frame, which completes normally.

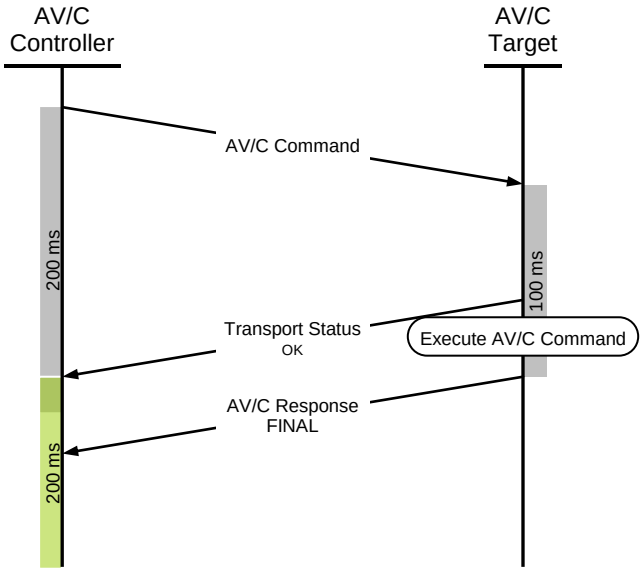

Figure 12 – Slow AV/C target with Transport Status OK

Copyright © 2008, 1394 Trade Association. All rights reserved.

The last case, illustrated by Figure 12, shows the use of Transport Status to prevent response timeout at the AV/C controller. The AV/C target knows, by design that the particular command takes nearly 100 ms to execute. Given that the delays on the communications path between controller and target could approach 50 ms each way, the transaction might timeout before the target's final response is received by the controller. The target guards against this possibility by transmitting a Transport Status response whose *rcode* is zero (OK). Upon receipt of the Transport Status response, the controller resets its response timeout timer to count down from 200 ms. In almost all cases, the target's final response frame should arrive at the controller before the extended response timeout expires.

6.3 Connection Management Procedures over IPv4 (CMP/IP)

Most of the CMP/IP connection setup tasks are performed by L3 IP bridges; this includes all of the following:

- Allocation of isochronous resources, channel number and bandwidth, within the sink device's IEEE 1394 cluster;
- Creation of an isochronous stream within the coaxial cable piconet;
- Allocation of isochronous resources, channel number and bandwidth, within the source device's IEEE 1394 cluster;
- Programming the source device's oPCR with the appropriate channel number and speed;
- Programming the sink device's iPCR with the appropriate channel number; and
- Establishing point-to-point connections between the source device's oPCR and the initial ingress bridge port's iPCR as well as between the terminal egress bridge port's oPCR and the sink device's iPCR.

L3 IP bridges adhere to CMP as specified by [R5] in the performance of the tasks above. All that is required are the parameters of the stream's bandwidth. An AV/C controller shall provide the necessary parameters in a PATH REQUEST message.

Initialization of most of the fields of the PATH REQUEST message is straightforward and described in [R3]. Additional explanation is appropriate for the *source_plug*, *sink_plug*, *max_payload*, *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields—which together specify the stream's endpoints and its bandwidth requirements.

6.3.1 Basic connection setup procedures

The method by which an AV/C controller determines which output plug index to store in the *source_plug* field is little changed from that used by legacy AV/C controllers. The principal difference is the manner in which a controller "reads" the source device's plug control registers. FCP/IP devices have no direct access to the registers but instead use the”tunneled” IEEE 1394 read facility of FCP/IP to obtain a copy of the registers. If the source device's oMPCR indicates that the device implements only one oPCR, the controller shall zero the *source_plug* field. Otherwise, the controller shall use criteria beyond the scope of this specification to select one of the oPCRs and store its index in the *source_plug* field.

Similar considerations apply to the *sink_plug* field. If the sink device implements only one iPCR, the controller shall zero the *sink_plug* field. Otherwise, the controller shall use criteria beyond the scope of this specification to select one of the iPCRs and store its index in the *sink_plug* field.

The *max_payload* field shall be set to four times the value of the *payload* field in the source device's oPCR identified by the *source_plug* field in the PATH REQUEST message.

Basic initialization of the PATH REQUEST message does not make use of the *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields, which should all be zeroed.

Copyright © 2008, 1394 Trade Association. All rights reserved.

Once the AV/C controller has initialized the PATH REQUEST message, it shall multicast it as a UDP/IP datagram addressed to all routers connected to the local subnet, 224.0.0.2, at well-known LDP port 646. All L3 IP bridges shall ignore the PATH REQUEST message except for the one L3 IP bridge connected to the same cluster as the AV/C target identified by *sink_EUI64*. See [R3] for details of L3 IP bridge processing of the PATH REQUEST message.

NOTE – An error message might be returned by the residential network's gateway router, which is not expected to implement LDP. If the AV/C controller receives such an error message, it should be ignored.

When path setup completes, successfully or in error, a PATH NOTIFICATION message will be returned to the AV/C controller. If the Status field indicates success, the controller may activate the source device's output. Otherwise, in the case of failure, the L3 IP bridges have already torn down the path and no additional action is required of the controller.

6.3.2 Recommended connection setup procedures

Although initialization of the *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields in the PATH REQUEST message is optional, AV/C controllers that have additional information about the bandwidth requirements of the source device's output stream should initialize these fields. The inclusion of this additional information in the PATH REQUEST message is of critical importance to efficient utilization of a scarce resource, data transmission time over the coaxial cable backbone. Given this information, L3 IP bridges might be able to allocate less time for a particular stream and thus leave more time available for other, concurrently active streams.

For example, consider an L3 IP bridge designed to accumulate roughly 10 ms of data before transmitting the aggregate over the coaxial cable. If the advanced parameters described in this clause are not provided, the bridge can only assume that *max_payload* bytes of isochronous data will be received from the source device every 125 µs. However, it is not necessarily true that *N* * *max_payload* bytes of isochronous data will be received from the source device every *N* * 125 µs. A constant bit rate application might generate *P* bytes of isochronous data every 100 µs; *max_payload* would have to be set to 2*P* even though only one out of five isochronous intervals would contain 2*P* bytes of isochronous data. If the bridge were unaware of the slower average bit rate, it would allocate, of necessity, sufficient channel time for the transmission of 160*P* bytes every 10 ms when all that would be required is channel time adequate to transmit 100*P* bytes during the same 10 ms interval. The channel timesavings can be significant, which is an especially important consideration since coaxial cable channel time is a critical, scarce resource.

In order for the AV/C controller to provide useful information in the *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields, it requires information beyond what is provided in the oPCR. Two of these fields, *source_quantum* and *source_bit_rate*, can be initialized with data obtainable from the IEC 61883 data format specifications. The other two, *window* and *aggregate_payload*, are derived by the AV/C controller, as explained below.

Given the values of *source_quantum* and *source_bit_rate*, the controller calculates the inter-arrival interval, $T_{ARRIVAL}$, of source quanta generated by the AV/C target's constant bit rate application:

$$T_{ARRIVAL} = \text{source\_bit\_rate} / (8 * \text{source\_quantum})$$

The next steps involve controller heuristics; it should try different *window* durations to discover the one that yields most efficient result for an *aggregate_payload* in the approximate vicinity of 4,000 bytes. Given a *window* size expressed as an integer number of 125 µs intervals, *aggregate_payload* may be derived from the following formula:

$$\text{aggregate\_payload} = (125 * \text{window} / T_{ARRIVAL} + 1) * (\text{source\_quantum} + K_{OVERHEAD})$$

The division operator represents integer division, hence the addition of one to round up to the next integral number of source quanta generated in a single isochronous interval. The constant, $K_{OVERHEAD}$ represents encoding overhead associated with each source quantum. For example, in the case of IEC 61883 MPEG transport stream encoding there are eight bytes of the CIP header overhead for each isochronous packet and four bytes of source packet header (SPH) overhead for each 188-byte source quantum. Note that it is important to calculate the inter-arrival interval from the

Copyright © 2008, 1394 Trade Association. All rights reserved.

constant bit rate application's intrinsic speed and only later to add the encoding overhead. The assignment of CIP header overhead to every source quantum is an over-allocation. In reality 8 * *window* quadlets of overhead would be attributable to CIP header's.

The efficiency of a particular *window* size is expressed relative to the maximum aggregate payload possible within *window* * 125 µs if the only basis for an estimate is *max_payload*, That is:

```
max_aggregate_payload = window * max_payload
```

Efficiency is expressed as a percentage:

```
efficiency = 1 - (aggregate_payload / max_aggregate_payload)
```

Absent other considerations, an AV/C controller should select the *window* interval with the largest efficiency.

After these advanced stream characteristics have been selected and calculated, the AV/C controller shall launch the PATH REQUEST message as described in 6.3.1 and then await a PATH NOTIFICATION message upon completion of path setup.

6.3.3 Connection teardown procedures

Just as in the case for CMP/IP connection setup, most of the connection teardown work is performed by L3 IP bridges; this includes all of the following:

- Deletion of a point-to-point connection between the terminal egress bridge port's oPCR and the sink device's iPCR;
- Deallocation of isochronous resources, channel number and bandwidth, within the sink device's IEEE 1394 cluster;
- Deletion of the pertinent stream within the coaxial cable piconet;
- Deletion of a point-to-point connection between a source device's oPCR and the initial ingress bridge port's iPCR; and
- Deallocation of isochronous resources, channel number and bandwidth, within the source device's IEEE 1394 cluster;

Wherever applicable, L3 IP bridges conform to [R5] when tearing down a connection.

Once the AV/C controller has initialized the PATH TEARDOWN message as specified by [R3], it shall multicast it as a UDP/IP datagram addressed to all routers connected to the local subnet, 224.0.0.2, at well-known LDP port 646. All L3 IP bridges shall ignore the PATH TEARDOWN message except for the one L3 IP bridge connected to the same cluster as the AV/C target identified by *sink_EUI64*. See [R3] for details of L3 IP bridge processing of the PATH TEARDOWN message.

NOTE – An error message might be returned by the residential network's gateway router, which is not expected to implement LDP. If the AV/C controller receives such an error message, it should be ignored.

When path teardown completes, successfully or in error, a PATH NOTIFICATION message will be returned to the AV/C controller.

Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex A
(normative)

AV/C commands unsupported across L3 IP bridges

A small subset of the AV/C command set is not supported for transport over L3 IP bridges *via* FCP/IP. Attempts to use these unsupported commands will produce unspecified results for one or both of the following reasons:

– The AV/C command contains an embedded IEEE 1394 node ID, which is meaningful only within the context of the node's IEEE 1394 cluster. If the referenced device is remote with respect to the controller, the controller could not have obtained its node ID. If the referenced device is local with respect to the controller, the node ID either will be invalid in the context of the target's IEEE 1394 cluster or will erroneously reference some unintended device in that cluster; or

– The AV/C command relies upon some IEEE 1394 facility that cannot be accessed across L3 IP bridges, *e.g.*, asynchronous connections use read, write and lock requests to access a set of CSRs, which, in turn, generate block write requests to transfer data asynchronously from one AV/C device to another. None of those IEEE 1394 transactions operates across L3 IP bridges.

The table below enumerates the AV/C commands unsupported for transport across L3 IP bridges *via* FCP/IP:

Command	Reference	Opcode	Implementation Requirements		
			CONTROL	STATUS	NOTIFY
OPEN INFO BLOCK	[B3]	5	Optional [a]	Optional	
OPEN DESCRIPTOR		8	Mandatory [a]	Optional	Optional
CHANNEL USAGE	[R1]	12_{16}		Recommended	Recommended
INPUT SELECT	[B2]	$1B_{16}$	Mandatory	Mandatory [b]	
OUTPUT PRESET		$1C_{16}$	Optional	Optional	
ASYNCHRONOUS CONNECTION	[B1]	26_{16}	Mandatory	Mandatory	
CAPTURE REF	[B4]	44_{16}	Mandatory	Mandatory	

[a] The CONTROL variant of this command is supported for transport across L3 IP bridges *via* FCP/IP.

[b] Optional for some profiles

AV/C controllers should not originate any of the above commands if they would be transported *via* FCP/IP. AV/C targets should return an immediate response of NOT IMPLEMENTED for any of the above commands originated by a remote AV/C controller. An AV/C target that does not distinguish between local and remote devices should return an immediate response of NOT IMPLEMENTED for any of the above commands transported *via* FCP/IP. More information about individual commands is provided in the clauses that follow.

A.1 AV/C Descriptor Mechanism commands

The optional STATUS command type of OPEN INFO BLOCK and OPEN DESCRIPTOR requests the target to return the identity (node ID) of the AV/C target that is holding the specified resource, information block or descriptor, open and thereby preventing other devices from accessing the resource. In theory, the controller that desires access to the resource could then request the identified target to relinquish its access—but AV/C specifies no protocol either to request or to compel the current owner to relinquish access.

1394 TRADE ASSOCIATION

Copyright © 2008, 1394 Trade Association. All rights reserved.

The optional NOTIFY command type of OPEN DESCRIPTOR requests the target to notify the controller of any change in ownership of the specified resource, information block or descriptor. If the change in ownership is from owned to un-owned (node ID equals $FFFF_{16}$), the command operates successfully across L3 IP bridges and might have marginal utility.

As these commands have debatable utility[9], no problems are anticipated if they remain unsupported for transport across L3 IP bridges *via* FCP/IP.

A.2 CHANNEL USAGE command

This optional AV/C command, when transported *via* FCP, is either unicast to a single AV/C target or broadcast to all AV/C devices within the local cluster. In the unicast case, the AV/C target is expected to respond that it is or is not using the specified channel, while in the broadcast case, a response is expected from only the target using the channel.

A.3 Connection and Compatibility Management (CCM) commands

CMP/IP (see 6.3), which has been designed to work across L3 IP bridges, provides an alternative to CCM and thus obviates the need to use the unsupported INPUT SELECT and OUTPUT PRESET commands.

NOTE – Although [B5] acknowledges that IP-based alternatives to CCM exist, it nevertheless mandates CCM implementation by AV/C devices. The CEA should be advised to delete the requirement in a future corrigendum or revision.

A.4 ASYNCHRONOUS CONNECTION command

AV/C asynchronous connections provide a confirmed, flow-controlled method for data transfer from a producer device (source) to the consumer device (sink) so long as both devices are connected to the same IEEE 1394 cluster. The numerous subfunctions of the ASYNCHRONOUS CONNECTION command are used to setup and tear down asynchronous connections mediated by asynchronous plug registers. In addition to the ASYNCHRONOUS CONNECTION command, the protocol uses IEEE 1394 read, write and lock requests—none of which are transported across L3 IP bridges.

The AV/C asynchronous connections protocol is utilized by the following:

- EIA/CEA-775 (for the transmission of EIA-799 OSD graphics);
- AV/C Camera/Storage subunit (for the transmission of files);
- AV/C Panel Subunit (for the direct mode transmission of GUI elements); and
- AV/C Printer subunit (for the transmission of documents).

NOTE – AV/C asynchronous connections could be adapted to IPv4 transport, should it be necessary.

A.5 CAPTURE REF command

In addition to an embedded node ID within its format, the CAPTURE REF command makes use of AV/C asynchronous connections.

NOTE – If AV/C asynchronous connections are in future adapted to IPv4 transport, CAPTURE REF could be revised to use the IP-based asynchronous connections.

[9] The use of OPEN INFO BLOCK is deprecated by AV/C.

Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex B
(normative)

Service announcement and discovery with SSDP

In addition to the service announcement and discovery methods mandated by 6.1, an FCP/IP-capable device might also implement support for Simple Service Discovery Protocol (SSDP [B26]). Although such implementation is optional, if implemented it shall conform to this annex.

B.1 Service announcement

Subsequent to power reset or when newly connected to an IEEE 1394 cluster, an FCP/IP-capable target device shall announce its presence by transmitting a multicast HTTP message to 239.255.255.250:1900; the method shall be `NOTIFY` and the notification subtype shall be `ssdp:alive`. The text below in italics indicates values to be provided by the device manufacturer.

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age = advertisement lifespan, in seconds
LOCATION: URL of XML file that describes this instance of FCP/IP
NT: urn:1394ta-org:service:fcpIp:1
NTS: ssdp:alive
SERVER: firmware/version UPnP/1.0 product/version
USN: uuid:eui64::urn:1394ta-org:service:fcpIp:1
```

The value of `max-age` should be 1800 (30 minutes).

Since AV/C controllers are not obligated to implement mDNS, the host component of the `LOCATION`: URL should not require DNS resolution—it should be a numeric IPv4 address.

The `SERVER`: parameters `firmware`, `product` and their respective `version` are all vendor-dependent and should be descriptive.

The `uuid` parameter shall represent the value of the device's EUI-64 and shall be encoded as a 16-character string of hexadecimal digits 0 – 9 and A – F, inclusive, each of which represents, in network nibble order, four bits of the 64-bit number.

B.2 Planned disconnection or shutdown

When the disconnection or shutdown of a FCP/IP-capable target device and its associated FCP/IP service are planned, the service should multicast an `ssdp:byebye` to alert potential users of the FCP/IP service. Receipt of this message cancels all the outstanding FCP/IP service announcements from the device. The format of the message is as follows:

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
NT: urn:1394ta-org:service:fcpIp:1
NTS: ssdp:byebye
USN: uuid:eui64::urn:1394ta-org:service:fcpIp:1
```

The `uuid` parameter shall be encoded as described in B.1.

Copyright © 2008, 1394 Trade Association. All rights reserved.

B.3 Service discovery with M-SEARCH

When an FCP/IP-capable controller device is newly connected to an IEEE 1394 cluster, it should transmit a multicast HTTP message to 239.255.255.250:1900; the method shall be M-SEARCH.

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: “ssdp:discover”
MX: 1
ST: urn:1394ta-org:service:fcpIp:1
```

B.4 Service discovery response

An FCP/IP-capable target device should respond to an M-SEARCH discovery request with a unicast HTTP message transmitted to the source IPv4 address and source port number. The structure of the service discovery response, shown below, is essentially similar to that of the announcement.

```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age = advertisement lifespan, in seconds
DATE: datestamp response generation
EXT:
LOCATION: URL of XML file that describe this instance of FCP/IP
SERVER: firmware/version UPnP/1.0 product/version
ST: urn:1394ta-org:service:fcpIp:1
USN: uuid:eui64::urn:1394ta-org:service:fcpIp:1
```

The comments made in B.1 continue to apply to the max-age parameter, the host component of the LOCATION: URL, the SERVER: parameters firmware, product and their respective version and the uuid parameter.

The DATE: header, if present, shall record the full time (as specified by [B21] [B18]) the response was generated.

The presence of the EXT: header confirms that the MAN: header was understood by the target device.

B.5 FCP/IP service XML file

For an AV/C target, the LOCATION: URL shall point to an XML file with the following contents:

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <service>
    <name> fcpIp </name>
    <descriptorList>
      <descriptor>
        <name> fcpPort </name>
        <dataType> ui2 </dataType>
        <value> N </value>
      </descriptor>
      <descriptor>
        <name> unitSpecifierID </name>
        <dataType> bin.hex </dataType>
        <value> 00A02D </value>
      </descriptor>
      <descriptor>
        <name> unitVersion </name>
        <dataType> bin.hex </dataType>
        <value> 010001 </value>
      </descriptor>
    </descriptorList>
  </service>
</scpd>
```


Copyright © 2008, 1394 Trade Association. All rights reserved.

The `fcpPort` variable shall specify the port number, 1024 through 65 535, inclusive, to which UDP/IP datagrams intended for the FCP/IP service shall be addressed. There is no expectation that different instantiations of the FCP/IP service by different target devices would use the same port number. However, devices that implement more than one instance of the FCP/IP service (*e.g.*, multiple AV/C units within a single IEEE 1394 node) shall use a different port number for each instance of the service.

The value of the `unitSpecifierID` variable, $00A02D_{16}$, indicates that the 1394 Trade Association is responsible for the definition of the target device.

The value of the `unitVersion` variable, 010001_{16} indicates that the device implements the 1394 Trade Association AV/C command set.

B.6 Integration with CEA 2027-B

Devices conformant to [B5] use SSDP to advertise their presence, but do so by using a UPnP "basic" device as the search target for device discovery queries. In turn, the basic device points to a 2027-B XML definition file that fully characterizes the device and its subdevices—but does so using XML syntax and semantics standardized by 2027-B.

2027-B devices that use FCP/IP for the delivery of AV/C command and response frames should include the following XML code in the device's XML definition file:

```
<service>
  <name> fcpIp </name>
  <descriptorList>
    <descriptor>
      <name> fcpPort </name>
      <dataType> ui2 </dataType>
      <value> N </value>
    </descriptor>
    <descriptor>
      <name> unitSpecifierID </name>
      <dataType> bin.hex </dataType>
      <value> 00A02D </value>
    </descriptor>
    <descriptor>
      <name> unitVersion </name>
      <dataType> bin.hex </dataType>
      <value> 010001 </value>
    </descriptor>
  </descriptorList>
</service>
```

The variable FcpPort defined by [B5] and the `fcpPort` variable defined by this specification should be set to the same value. If they are mistakenly set to different values, the value of `fcpPort` shall take precedence.

1394 TRADE ASSOCIATION

Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex C
(normative)

Minimum node capabilities for IEEE 1394 interfaces on FCP/IP-capable devices

FCP/IP-capable devices that implement an IEEE 1394 interface shall conform to the requirements of this annex, which are in addition to the minimum capabilities defined by IEEE 1394 ([R6]) All IEEE 1394 interfaces implemented by an FCP/IP-capable device are subject to these requirements.

The device shall be able to receive and transmit primary packets whose data payload is less than or equal to 512 bytes; the *max_rec* field in the device's bus information block shall be greater than or equal to eight. This ability applies to requests and responses addressed to or originated by the device.

The device shall be able to initiate write requests and shall report this by setting the *drq* bit in the Node_Capabilities entry in configuration ROM to one. Consequently, FCP/IP-capable devices shall implement the *dreq* bit in the STATE_CLEAR and STATE_SET registers. The value of STATE_CLEAR.*dreq* shall be unaffected by a bus reset. FCP/IP-capable devices may autonomously set *dreq* to zero (request initiation enabled) upon a power reset or a command reset.

During initialization subsequent to a power reset and if its link layer is active, the device shall either respond to quadlet read requests addressed to FFFF F000 0400_{16} with a response data value of zero or acknowledge the request subaction with *ack_tardy*. This indicates that the remainder of configuration ROM, as well as other FCP/IP-capable device CSRs, is not accessible.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex D
(normative)

Conformance requirements

This annex is intended to assist designers, implementers and conformance test developers; it provides a concise summary of mandatory and optional features and, for each feature, reference to the governing normative clauses or external documents.

The conformance requirements for FCP/IP-capable devices are specified by the tables below. Related features are grouped together in separate tables; each feature is identified by an item designator. References within brackets are normative or bibliographic citations while numeric references without brackets denote sections or clauses within this document. Subsidiary clauses inherit the implementation requirements of their parent clause. Entries in the implementation column are decipherable according to the following table.

Entry	Interpretation
Mandatory	The device shall implement the feature.
Optional	The device may implement the feature.
Not applicable	The feature is outside the scope of this document.
Optional [*label*]	The device shall implement at least one of the options belonging to the option group designated by *label*.
Item: *status*	The implementation *status* (mandatory, optional or not applicable) of the feature is conditional upon the implementation of another feature identified by *item*.

AV/C target and controller function modes are entirely orthogonal to each other; both may be concurrently active within the same device without interference. A device may implement target functionality but not controller, controller functionality but not target—or both target and controller functionality.

Table D-1 – AV/C function modes implementation requirements

Item	Feature	Reference	Implementation
MC	AV/C controller	[R1]	Optional [M]
MT	AV/C target	[R1]	Optional [M]

FCP/IP can be implemented on any IP-capable network interface—but not all interfaces are suitable to all AV/C function modes.

Table D-2 – Network interface implementation requirements

Item	Feature	Reference	Implementation
N1	IEEE 802		MC: Optional [N] MT: Not applicable
N2	IEEE 1394	[R6]	MC: Optional [N] MT: Mandatory
N3	Minimum node capabilities	Annex C	N2: Mandatory

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

NOTE – Because CMP/IP requires plug control registers on the source and sink devices, IEEE 1394 is the only network interface suitable for AV/C targets.

All FCP/IP-capable devices shall implement Internet protocol on all of their network interfaces. Different interface technologies may have different normative references for their implementation, as is noted.

Table D-3 – Internet protocol implementation requirements

Item	Feature	Reference	Implementation
IP1	Internet Protocol version 4 (IPv4)	N1: [B13] N2: [R10]	Mandatory
IP2	Address resolution protocol (ARP)	N1: [B14] N2: [R10]	Mandatory
IP3	Distributed host control protocol (DHCP)	N1: [B19] N2: [R11]	Mandatory
IP4	Link-local address assignment	[R12]	Mandatory
IP5	Multicast DNS (mDNS)	[R8]	MC: Optional MT: Mandatory
IP6	DNS-based service discovery (DNS-SD)	[R7]	MC: Optional MT: Mandatory

All FCP/IP features—both data structures and operations—are mandated for all interfaces of all devices conformant to this specification.

Table D-4 – FCP/IP data structure format implementation requirements

Item	Feature	Reference	Implementation
D1	FCP data unit (FC PDU)	5.1	Mandatory
D2	Path management messages	5.3	MC: Mandatory
D3	TIME OFFSET message	5.3	Optional

Table D-5 – FCP/IP operations implementation requirements

Item	Feature	Reference	Implementation
FCP1	AV/C transactions over FCP/IP	6.2	Mandatory
FCP2	CMP/IP	6.3	MC: Mandatory

As can be seen in the table below, all FCP/IP-capable AV/C devices shall implement legacy AV/C features, *e.g.*, an AV/C unit directory, the ability to utilize legacy FCP for command and response frame transport, and plug control registers, in addition to the FCP/IP capabilities. This is important to preserve interoperability with legacy AV/C devices that might be connected to the same IEEE 1394 cluster as the FCP/IP-capable AV/C device.

Table D-6 – AV/C implementation requirements

Item	Feature	Reference	Implementation
AVC1	Configuration ROM AV/C unit directory	[R5]	MC: Not applicable MT: Mandatory

Copyright © 2008, 1394 Trade Association. All rights reserved.

Item	Feature	Reference	Implementation
AVC2	Function control protocol (FCP)	[R5]	MC: Optional MT: Mandatory
AVC3	Connection management procedures (CMP)	[R5]	MC & AVC2: Mandatory MT: Optional
AVC4	Plug control registers (PCRs)	[R5]	MC: Not applicable MT: Mandatory
AVC5	Multiple AV/C units		MC: Not applicable MT: Optional [a]

[a] An AV/C target that implements more than one AV/C unit shall have a configuration ROM AV/C unit directory for only one of the units.

A key architectural requirement for this specification is the ability to support HANA devices; if this optional feature is implemented, mandates the implementation of additional features.

Table D-7 – HANA implementation requirements

Item	Feature	Reference	Implementation
H1	CEA 2027-B	[B5]	Optional
H2	Simple Service Discovery Protocol (SSDP)	[B26]	H1: Mandatory

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex E
(informative)

Message sequence charts (MSCs) for CMP/IP operations

Message sequence charts are frequently used to represent ordered relationships in the exchange of messages between two or more agents. [B25] formally specifies MSC, which consists of two parts: graphic symbols and an accompanying systems definition language (SDL). This annex loosely adopts—and adapts—MSC graphic notation to illustrate the nominal sequence of events for common CMP/IP operations.

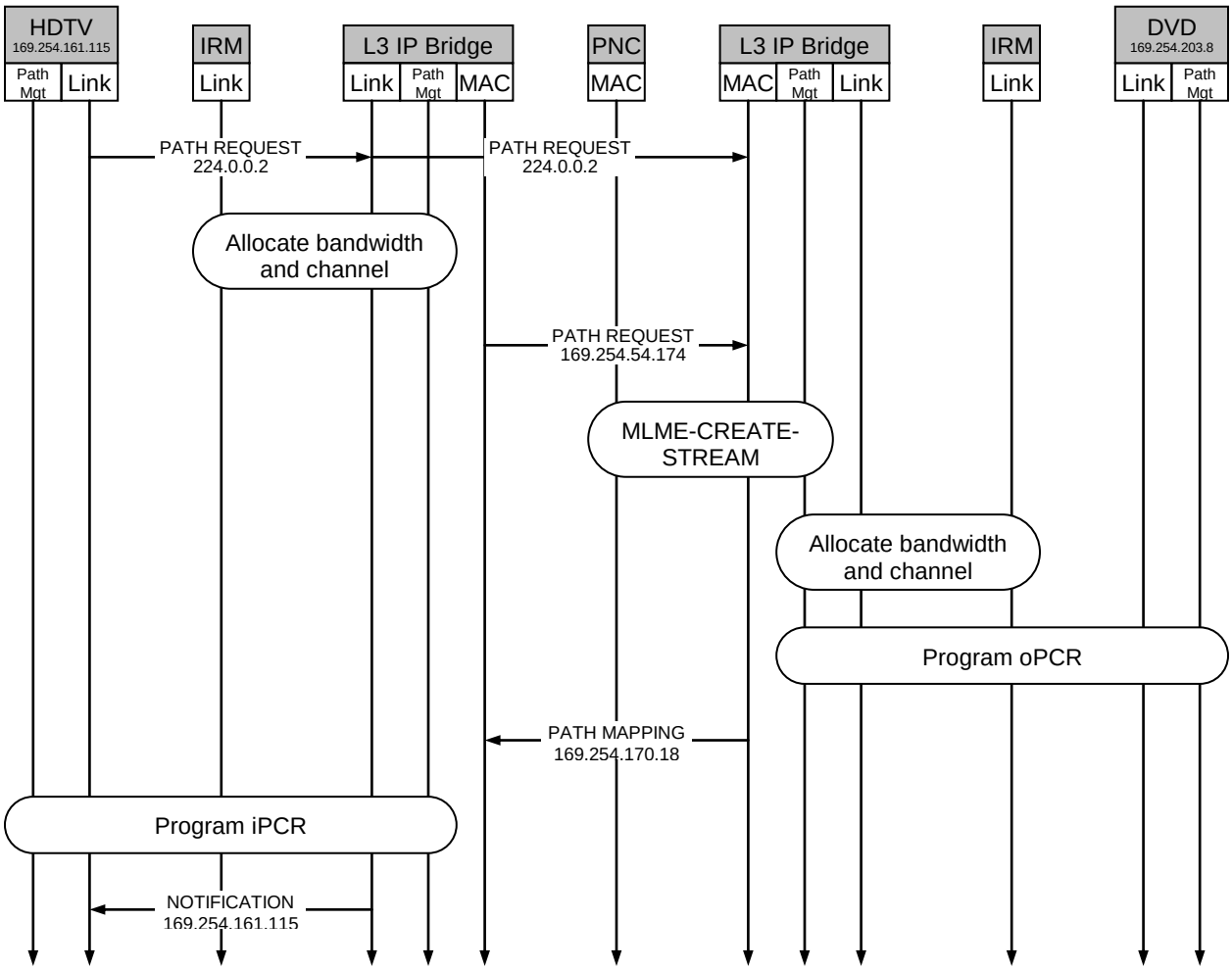

Figure E-1 – Path setup operations

NOTE – Both the two-hop topology and the device's IPv4 address assignments were taken from Figure 4. Other topologies that have more bridge hops might be feasible but are beyond the scope of this specification.

Isochronous connection set up begins when a controller (*e.g.*, an HDTV) initializes and transmits a PATH REQUEST message that specifies a source device (a DVD player) and a sink device (the HDTV itself).[10] The message is initialized as specified by [R3], encapsulated as an IP datagram and multicast to all subnet routers' well-known LDP

[10] Because sink devices often possess a display—hence a relatively rich user interface capability—controller functionality is often co-located within a sink device.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

port 646. All bridges receive the PATH REQUEST message, but only the bridge with a port connected to the sink device's cluster processes it. First, the bridge allocates a channel number and bandwidth from the isochronous resource manager. If the allocations succeed, the bridge enters the information into its LIB, prepends an MSDU header to the IP datagram that contains the PATH REQUEST and transmits the resultant MSDU to the next upstream bridge *en route* to the source device. When that bridge receives the message, it requests the piconet controller (PNC) to create an isochronous stream whose MLME-CREATE-STREAM parameters are derived from the bandwidth information in the PATH REQUEST message. If the PNC creates the stream, the bridge records the Stream Index in its LIB and checks whether isochronous resources are already allocated for the source device. If the source's output plug control register (oPCR) connection count is zero, the bridge allocates a channel number and bandwidth. If successful, the bridge programs the source's oPCR and its own iPCR with the channel number and increments the point-to-point connection counters of both plug control registers. At this point, resources have been allocated for all segments of the stream's path.

Next, the bridges distribute the “labels” that identify the stream in each segment: channel number for IEEE 1394 and Stream Index for coaxial cable. The bridge adjacent to the source device transmits a PATH MAPPING message (see [R3]), which contains the Stream Index that identifies the stream within the piconet, to the bridge adjacent to the sink device. When the bridge receives the message, it updates its LIB with the association between unique stream identifier and Stream Index. Finally, the bridge programs its own oPCR and the sink device's iPCR with the channel number used by the stream within the cluster and increments the point-to-point connection counters of both PCRs.

Now the isochronous path is set up and ready for use. The last downstream bridge transmits a PATH NOTIFICATION message to the controller as confirmation that the stream path has been set up successfully.

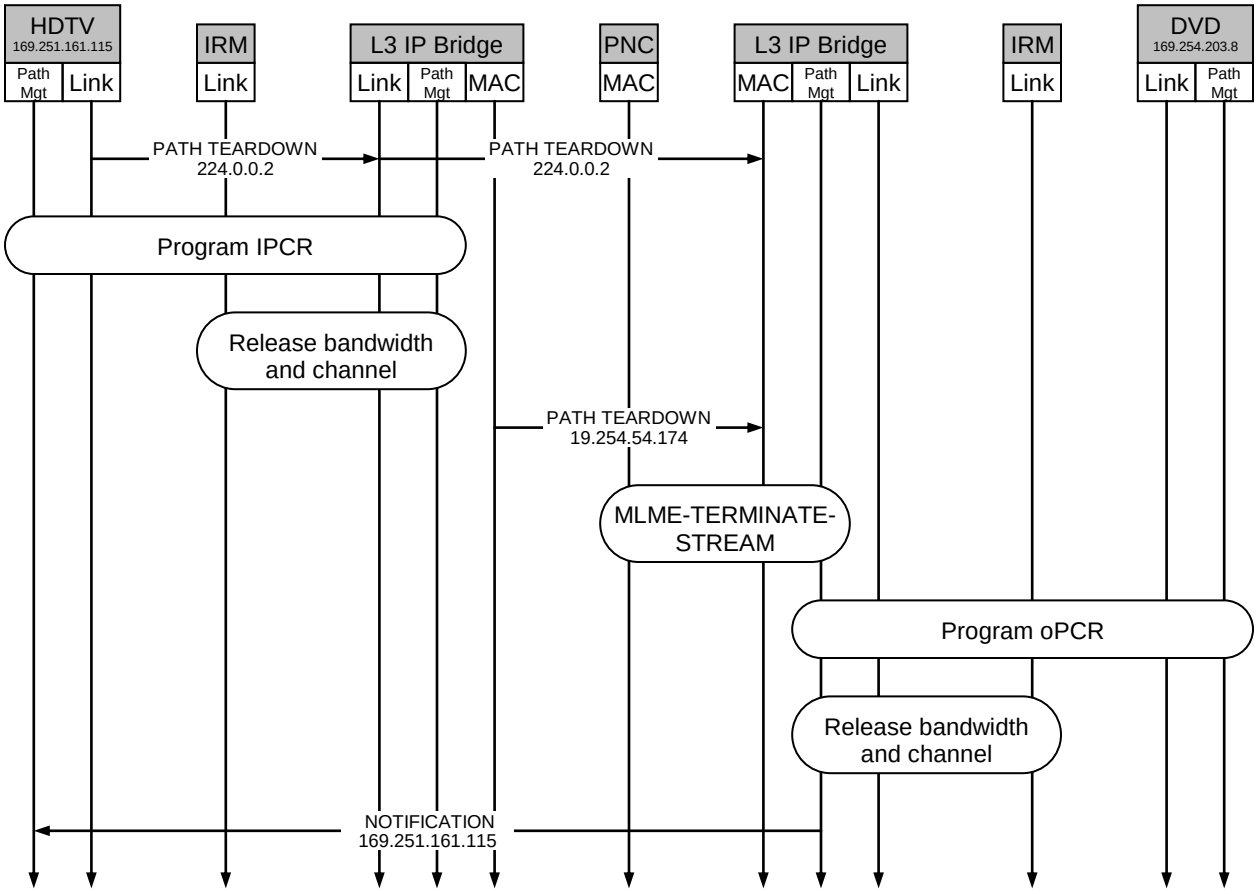

Figure E-2 – Path teardown operations

Copyright © 2008, 1394 Trade Association. All rights reserved.

Isochronous connection teardown begins when a controller (*e.g.*, an HDTV) initializes and transmits a PATH TEARDOWN message that identifies a sink device (*e.g.*, the HDTV itself) and a source device (*e.g.*, a DVD player) for which the controller previously establish the connection. The PATH TEARDOWN message is initialized as specified by [R3], encapsulated as an IP datagram and multicast to all subnet routers, well-known LDP port 646. L3 IP bridges ignore all multicast PATH TEARDOWN messages unless their IEEE 1394 port connects to the cluster that contains the sink device. Upon receipt of the PATH TEARDOWN message, the bridge decrements the point-to-point connection counters of the sink device's iPCR and the bridge's oPCR. If the bridge’s point-to-point connection counter is nonzero, other nodes are still utilizing the stream and it cannot be terminated for the cluster. In this example, however, the HDTV is assumed the only listener for the stream. The bridge releases the channel number and bandwidth allocated during path setup. The appropriate entries in the bridge's LIB are cleared and the PATH TEARDOWN is transmitted to the next upstream bridge. Receipt of the PATH TEARDOWN message causes the bridge to delete the path to the downstream bridge. If there are no other downstream coaxial cable bridges listening to the stream, the bridge uses MLME-TERMINATE-STREAM to release the piconet resources reserved for the stream and updates its LIB accordingly. Since (in this example) no downstream listeners remain, the bridge programs the source device's oPCR and the bridge's own iPCR to render them inactive before it releases the channel number and bandwidth that had been in use. At this point, path teardown is complete and the final upstream bridge transmits a NOTIFICATION message with a Status TLV to the controller that originated the PATH TEARDOWN MESSAGE.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex F
(informative)

Minimizing isochronous stream channel time

As was already stated in 6.3.2, coaxial cable backbone efficiency can be significantly improved if the optional *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields in the PATH REQUEST message are initialized by the AV/C controller. This annex examines the actual efficiency improvements attainable by varying *window* duration for a constant bit-rate 19.4 Mb/s ATSC isochronous stream. When output by an AV/C device, the stream takes the form of an SD MPEG-2 transport stream composed of fixed-length, 188-byte transport stream packets (TSP) at the approximate inter-arrival interval of 77.5 µs. At nominal 125 µs intervals, whatever TSPs have accumulated are encapsulated into a single isochronous subaction for transmission. Figure F-1 below illustrates the IEC 61883-4 (see [B7]) format of a sample isochronous subaction's data payload.

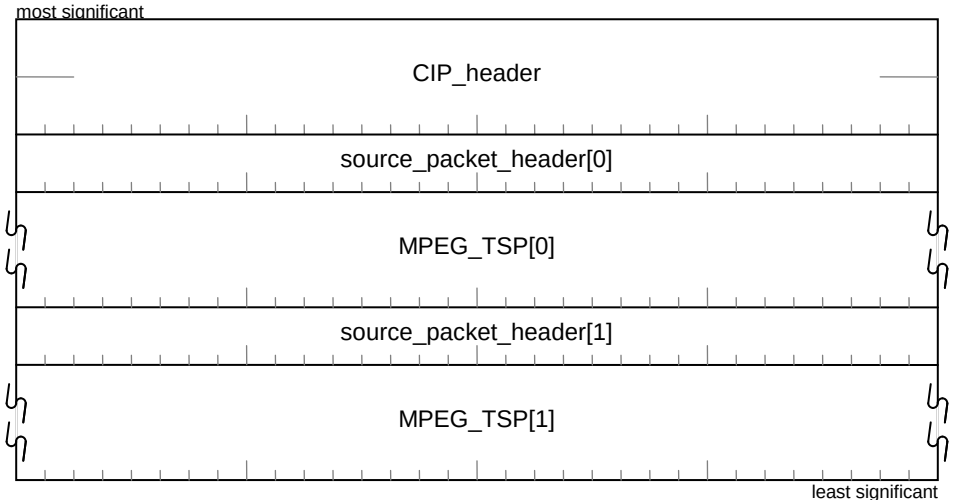

Figure F-1 – Common isochronous packet (CIP) format

NOTE – Consult IEC 61883-4 for detailed information on the *CIP_header* and *source_packet_header* fields.

Although Figure F-1 shows two MPEG TSPs, an isochronous packet for a 19.4 Mb/s ATSC stream may contain zero, one or at most two TSPs. The CIP header is included even if no TSP is present; consequently the data payload for an isochronous subaction with zero, one or two TSPs is eight, 200 or 392 octets, respectively.[11] Because IEEE 1394 requires allocation for the maximum payload that could occur in any isochronous interval, bandwidth will be allocated for 25.1 Mb/s. This over-allocation is unavoidable for IEEE 1394 link segment but can be improved upon significantly for transit across the coaxial cable backbone.

Consider first the channel time allocation required if the MAC accumulates isochronous data over some number of 125 µs intervals (the window duration). If the MAC has no better information upon which to base its estimate than the value of *max_payload* from the PATH REQUEST message, it would estimate the maximum aggregated payload as follows:

$$msdu\_alloc_X = K_{MSDU\ HEADER} + window * (K_{SSDU\ HEADER} + max\_payload)$$

[11] IEC 61883-1 is not definitive as to whether an isochronous subaction may be omitted if it contains no TSP. Implementers are encouraged to investigate this possibility, on a case-by-case basis, in order to conserve bandwidth.

Copyright © 2008, 1394 Trade Association. All rights reserved.

In the formula above, $K_{MSDU\ HEADER}$ and $K_{SSDU\ HEADER}$, both equal to four, represent overhead for the MSDU header and for each of the SSDU headers, respectively.

Next, consider the channel time allocation required if the AV/C controller initializes the *window*, *aggregate_payload*, *source_quantum* and *source_bit_rate* fields in the PATH REQUEST message. For the example 19.4 Mb/s ATSC stream, *source_quantum* would be set to 200, *source_bit_rate* would be set to 19 400 000 and *aggregate_payload* would be calculated according to the value of *window*, as shown below:

$$TSP = 188$$

$$K_{ARRIVAL} = source\_bit\_rate\ /\ TSP = 77.5\ \mu s$$

$$aggregate\_payload = K_{CIP\ HEADER} * window + CEILING(window * 125/K_{ARRIVAL}, 1) * (K_{SPH}+TSP)$$

In the formulae above, *TSP* is the size, in bytes, of a transport stream packet, $K_{ARRIVAL}$ is the inter-arrival period between the generation of transport stream packets by the isochronous source, $K_{CIP\ HEADER}$ is the 8-byte CIP header overhead, CEILING() is a function that rounds the result up to the next largest integer and K_{SPH} is the 4-byte source packet header overhead. Given this information, the MAC calculates largest MSDU to be transmitted:

$$msdu\_alloc_A = K_{MSDU\ HEADER} + window * K_{SSDU\ HEADER} + aggregate\_payload$$

The results for window durations ranging from 125 µs to 2 ms are summarized by the table below:

Window duration (µs)	MSDU allocation (bytes) based on		Payload reduction
	max_payload	*aggregate_payload*	
125	400	400	0%
250	796	796	0%
375	1192	1000	16%
500	1588	1396	12%
625	1984	1792	10%
750	2380	1996	16%
875	2776	2392	14%
1000	3172	2596	18%
1125	3568	2992	16%
1250	3964	3388	15%
1375	4360	3592	18%
1500	4756	3988	16%
1625	5152	4192	19%
1750	5548	4588	17%
1875	5944	4984	16%
2000	6340	5188	18%

The last column of the table shows the payload reduction achieved as a function of window duration; it is calculated as $1 - msdu\_alloc_A / msdu\_alloc_X$. As is evidenced by the table, some benefits of the accurate characterization of an isochronous stream accrue with window durations as short as 375 µs—but greater payload reductions are obtainable with longer aggregation windows.

Copyright © 2008, 1394 Trade Association. All rights reserved.

For window sizes greater than 2 ms, payload reduction asymptotically approaches 19% (the calculations are left as an exercise for the reader). This suggests that the MAC can obtain the greatest efficiencies if, subject to latency constraints, it utilizes a PPDU size between roughly 4 KiB and the maximum allowed.

1394 TRADE ASSOCIATION
Copyright © 2008, 1394 Trade Association. All rights reserved.

Annex G
(informative)

Bibliography

[B1] 1394 Trade Association, TA Document 2000006, AV/C Commands for Management of Asynchronous Serial Bus Connections 1.1, October 24, 2000

[B2] 1394 Trade Association, TA Document 2002010, AV/C Connection and Compatibility Management Specification 1.1, March 19, 2003

[B3] 1394 Trade Association, TA Document 2002013, AV/C Descriptor Mechanism 1.2, April 12, 2004

[B4] 1394 Trade Association, TA Document 2003004, AV/C Printer Subunit Specification 2.0, February 4, 2004

[B5] Consumer Electronics Association, CEA 2027-B, A User Interface for Home Networks Using Web-based Protocols

[B6] Digital Transmission Licensing Administrator, Digital Transmission Content Protection Specification—Volume 1 (Informational Version), Revision 1.2a, February 25, 2002

[B7] IEC 61883-4 (2004-08), Consumer audio/video equipment—Digital interface—Part 4: MPEG2-TS data transmission

[B8] IEEE Std 802.2-1998 Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical Link Control

[B9] IEEE Std 1212-2001, Standard for a Control and Status Registers (CSR) Architecture for microcomputer buses

[B10] IEEE Std 1394.1- 2004, Standard for High Performance Serial Bus Bridges

[B11] IETF Draft-Sekar-DNS-LLQ-01, DNS Long-Lived Queries, August 10, 2006

[B12] IETF Draft-Sekar-DNS-UL-01, Dynamic DNS Update Leases, August 10, 2006

[B13] IETF RFC 791, Internet Protocol, September 1981

[B14] IETF RFC 826, An Ethernet Address Resolution Protocol

[B15] IETF RFC 951, Bootstrap Protocol (BOOTP), September 1985

[B16] IETF RFC 1034, Domain Names – Concepts and Facilities, November 1987

[B17] IETF RFC 1035, Domain Names – Implementation and Specification, November 1987

[B18] IETF RFC 1123, Requirements for Internet Hosts – Application and Support, October 1989

[B19] IETF RFC 1533, DHCP Options and BOOTP Vendor Extensions, October 1993

[B20] IETF RFC 1542, Clarifications and Extensions for the Bootstrap Protocol, October 1993

Copyright © 2008, 1394 Trade Association. All rights reserved.

[B21] IETF RFC 2616, Hypertext Transfer Protocol – HTTP/1.1, June 1999

[B22] IETF RFC 2782, A DNS RR for Specifying the Location of Services (DNS SRV), February 2000

[B23] IETF RFC 3031, Multiprotocol Label Switching Architecture, January 2001

[B24] IETF RFC 4436, Detecting Network Attachment in IPv4 (DNAv4), March 2006

[B25] ITU-T Recommendation Z.120 (2004-04), Message sequence chart (MSC)

[B26] UPnP Forum, UPnP Device Architecture 1.0, Document Version 1.0.1, 20 July 2006

Copyright © 2008, 1394 Trade Association. All rights reserved.