IEEE STANDARDS ASSOCIATION

IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks

IEEE Computer Society

Sponsored by the
Microprocessor Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

6 May 2011

IEEE Std 1722™-2011

IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks

Microprocessor Standards Committee

of the

IEEE Computer Society

Approved 10 February 2011

IEEE-SA Standards Board

Abstract: The protocol, data encapsulations, and presentation time procedures used to ensure interoperability between audio- and video-based end stations that use standard networking services provided by all IEEE 802 networks meeting quality-of-service requirements for time-sensitive applications by leveraging concepts of IEC 61883 are specified in this standard.

Keywords: bridged LAN, IEC 61883, IEEE 802.1 AVB protocols, IEEE 802.1AS, IEEE 802.1BA, IEEE 802.1Qat, IEEE 802.1Qav, IEEE 1722, local area network (LAN), quality of service, time-sensitive media streaming, time synchronization

---

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6549-3 STD97076
Print: ISBN 978-0-7381-6550-9 STDPD97076

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 1722-2011, IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks.

Increasingly, entertainment media are digitally transported. Streaming audio/video and interactive applications over local area networks is becoming more common.

This standard builds on the work done by the IEEE 802.1 AVB task group by providing a common audio/video transport protocol capable of supporting the needs of both consumer and professional audio/video applications.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Audio/Visual Bridging Layer2 Transport Working Group had the following membership:

Robert Boatright, *Chair*
Matthew Xavier Mora, *Vice Chair*
Don Pannell, *Secretary*
David Olsen and Don Pannell, *Editors*
Alan K. Bartky, *Initial Editor*

Torrey Atcitty
Alex Beliaev
Alexander Busch
Ashley Butterworth
Debin Chen
George Claseman
William Dai
Guy Fedorkow
John Nels Fuller
Duncan Gray
Kevin Gross
Craig Gunther
Chuck Harrison
Joern Henneberg
Girault Jones
Tom Kennedy
Max Kicherer
Jeff Koftinoff
Raghu Kondapalli
Rick Kreifeldt
Thomas Mathey
Lee Minich
Suman Sharma
Rob Silfvast
Kevin B. Stanton
Michael Johas Teener
Fred Tuck
Ganesh Venkatesan
Niel Warren
Andy Yanowitz

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Thomas Alexander
Mark Anderson
Butch Anton
Danilo Antonelli
Lee Armstrong
Hugh Barrass
Georgios Bertos
Martin J. Bishop
Robert Boatright
Tomo Bogataj
Juan Carreon
Keith Chow
Charles Cook
Thomas Dineen
Sourav Dutta
Richard Eckard
Yukihiro Fujimoto
John Nels Fuller
Randall Groves
Craig Gunther
Joseph Gwinn
Stephen Haddock
David Hunter
C. Huntley
Akio Iso
Atsushi Ito
Anthony Jeffree
Girault Jones
Piotr Karocki
Stuart J. Kerry
Max Kicherer
Yongbum Kim
Bruce Kraemer
Geoff Ladwig
Paul Lambert
Richard Lancaster
Jeremy Landt
Greg Luri
Elvis Maculuba
Peter Martini
Matthew Xavier Mora
Michael S. Newman
David Olsen
Glenn Parsons
Subburajan Ponnuswamy
Robert Robinson
Benjamin Rolfe
Randall Safier
John Sauer
Peter Saunderson
Bartien Sayogo
Stephen Schwarm
Shusaku Shimada
Gil Shultz
Kapil Sood
Amjad Soomro
Kevin B. Stanton
Thomas Starai
Walter Struppler
Masahiro Takagi
William Taylor
Michael Johas Teener
David Thompson
Mark-Rene Uchida
Prabodh Varshney
Colin Whitby-Strevens
Kunpeng Wu
Oren Yuen
Janusz Zalewski
Zhen Zhou

When the IEEE-SA Standards Board approved this standard on 10 February 2011, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Malia Zaman
IEEE Standards Program Manager, Technical Program Development

Contents

Figures

Tables

IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

Increasingly, entertainment media are digitally transported. Streaming audio/video and interactive applications over bridged local area networks (LANs) need to have comparable real-time performance with legacy analog distribution. There is significant end-user and vendor interest in defining a simple yet common method for handling real-time audio/video suitable for consumer electronics, professional audio/video applications, etc. Technologies such as IEEE Std 1394™-2008, Bluetooth®, and USB exist today, but each has their own encapsulation, protocols, timing control, etc., such that building interworking functions is difficult. The use of a common audio/video transport over multiple IEEE 802® network types will realize operational and equipment cost benefits. By ensuring that all IEEE 802 wired and wireless devices share a common set of transport mechanisms for time-sensitive audio/video streams, we lessen the effort of producing interworking units between IEEE 802 and other digital networks.

1.1 Scope

This standard specifies the protocol, data encapsulations, and presentation time procedures used to ensure interoperability between audio- and video-based end stations that use standard networking services provided by all IEEE 802 networks meeting quality-of-service requirements for time-sensitive applications by leveraging the concepts of IEC 61883.

1.2 Purpose

This standard will facilitate interoperability between stations that stream time-sensitive audio and/or video across LANs providing time synchronization and latency/bandwidth services by defining the packet format protocols and synchronization mechanisms.

2. Normative references

The following referenced documents and URLs are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEC 61883-1:2003, Consumer Audio/Video Equipment— Digital Interface—Part 1: General.[1]

IEC 61883-2:2004, Consumer Audio/Video Equipment—Digital Interface—Part 2: SD-DVCR Data Transmission.

IEC 61883-4:2004, Consumer Audio/Video Equipment—Digital Interface—Part 4: MPEG2-TS Data Transmission.

IEC 61883-6:2005, Consumer Audio/Video Equipment—Digital Interface—Part 6: Audio and Music Data Transmission Protocol.

IEC 61883-7:2003, Consumer Audio/Video Equipment—Digital Interface—Part 7: Transmission of ITU-R BO.1294 System B.

IEC 61883-8:2008, Consumer Audio/Video Equipment—Digital Interface—Part 8: Transmission of ITU-R Bt.601 Style Digital Video Data.

IEEE P802.1AS™/D7.2 (Mar. 2010), Draft Standard for Local and Metropolitan Area Networks: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.[2,3,4]

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

IEEE Std 802.1Q™-2005, IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks.

IEEE Std 802.1Qat™, IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks—Amendment 14: Stream Reservation Protocol.

IEEE Std 802.1Qav™, IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks—Amendment 11: Forwarding and Queuing for Time-Sensitive Streams.

[1] IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iec.ch/). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.
[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[3] The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.
[4] This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

IEEE Std 802.3™-2008, IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

IEEE Std 802.11™-2007, IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

3. Terms, definitions, and notation

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions* should be consulted for terms not defined in this clause.[5]

3.1 Conformance levels

Several keywords are used to differentiate between different levels of requirements and optionally, as follows:

may: Indicates a course of action permissible within the limits of the standard with no implied preference ("may" means "is permitted to").

shall: Indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted ("shall" means "is required to").

should: An indication that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited ("should" means "is recommended to").

3.2 Definitions

Audio/Video Transport Protocol (AVTP): The protocol defined by IEEE Std 1722-2011. As an adjective, it indicates that the modified noun is specified in or interpreted in the context of this standard.

audio/video bridging (AVB) bridge: A bridge that implements IEEE Std 802.1Q™-2005 as amended by the Stream Reservation Protocol (SRP), Forwarding and Queuing Enhancements for Time Sensitive Streams (FQTSS), and implements the generalized Precision Time Protocol (gPTP).

Audio/Video Transport Protocol (AVTP) stream: An AVTP stream is between one Talker and one or more Listeners.

Audio/Video Transport Protocol (AVTP) gateway: A device that transports AVTP audio/video streams between an audio/video bridging (AVB) network and another type of media, for example, an IEEE 1394 bus.

doublet: Two octets of data.

frame: A frame is a complete unit of data transmission as transmitted on the network, including media access control (MAC) layer addressing and cyclic redundancy check (CRC).

---

[5] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

Listener: An end station that is the destination, receiver, or consumer of a stream.

octet: Eight bits of data.

packet: Refers to the IEC 61883 definition. Packet is used to refer to specific features of 61883 that use this term, such as source packet header (SPH) or common isochronous packet (CIP).

quadlet: Four octets of data.

Stream Reservation (SR) Class: As defined by IEEE Std 802.1Qav.

StreamID: A 64 bit field that uniquely identifies a stream, as defined in Stream Reservation Protocol (SRP).

Talker: An end station that is the source or producer of a stream.

3.3 Reserved fields

Any field within a data structure that is defined in this specification as reserved is reserved for future standardization. All reserved fields specified in this standard shall be set to a value of zero (0) on transmit. All reserved fields specified in this standard shall be ignored on receive.

3.4 Numerical values

Decimal, hexadecimal, and binary numbers are used within this document. For clarity, decimal numbers are generally used to represent counts, hexadecimal numbers are used to represent addresses, and binary numbers are used to describe bit patterns within binary fields.

Decimal numbers are represented in their usual 0, 1, 2, … format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0 to 9 and A to F) digits followed by the subscript 16. Binary numbers are represented by a string of one or more binary (0,1) digits in left to right order where the left-most bit is the most significant bit and the right-most bit is the least significant bit, followed by the subscript 2. Thus, the decimal number "26" may also be represented as "$1A_{16}$" or "11010_2."

These notational conventions have one exception: MAC addresses and IEEE Organizationally Unique Identifier (OUI)/IEEE Extended Unique Identifier (EUI) values are represented as strings of 8 bit hexadecimal numbers separated by hyphens and without a subscript as, for example, "01-80-C2-00-00-15" or "AA-55-11."

3.5 Notation of fields and values taken from other documents

This document uses fields and values defined in other documents with multiple methods of defining such things as usage of uppercase and lowercase, usage of underscore characters, italics, etc. As this document is intended to use these multiple protocols, its additional intent is also to make it easier for readers and implementers of those documents by not using different names and notation for those fields and values. So, the following conventions are used for field names from other documents to match the convention from those documents:

a) Fields from IEEE Std 802.1Q-2005: Fields are in all uppercase, no underscores (examples: destination MAC address [DA], source MAC address [SA], tagged protocol identifier [TPID], canonical format indicator [CFI], and VLAN identifier [VID]).
b) Fields from IEEE Std 1394-2008: Fields in lowercase except for acronyms within the field name with optional underscores (examples: tcode, data_length, and source_ID).

c) Fields from IEC 61883: Fields always starting in uppercase, acronyms in uppercase, abbreviations with uppercase first followed by lowercase, no underscores (examples: data block count [DBC], DBS, and Rsv).

3.5.1 Bit, octet, doublet, and quadlet ordering

All Audio Video Transport Protocol Data Units (AVTPDUs) consist of an integral number of octets, which are numbered starting from 1 and increasing in the order that they are put into a Data Link Service Data Unit. The bits in each octet are numbered from 0 to 7, where 7 is the low-order bit.

When consecutive octets are used to represent a binary number, the lower octet number has the most significant value. When the encoding of (an element of) a Multiple Registration Protocol Data Unit is represented using a diagram in this clause, the following representations are used:

a) Octet 1 is shown toward the top of the page, higher numbered octets being toward the bottom.
b) Where more than one octet appears on a given line, octets are shown with the lowest numbered octet to the left, higher numbered octets being to the right.
c) Within an octet, bits are shown with bit 0 to the left and bit 7 to the right.

Figure 3.1 represents a decimal value of 1, a hexadecimal value of 01_{16}, and a binary value of 00000001_2 with the only bit in the octet set to one of the least significant bit, bit 7.

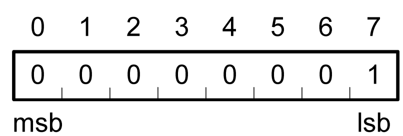


Figure 3.1—Bit ordering within an octet

A quadlet is a series of four octets. Within a quadlet, the most significant octet is shown as the left-most octet and the least significant octet is shown as the right-most, as shown in Figure 3.2.

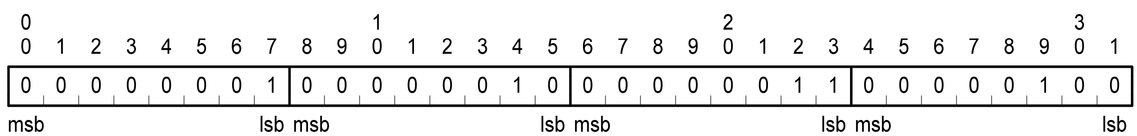


Figure 3.2—Octet ordering within a quadlet

Figure 3.2 represents a quadlet holding a four-octet array with values of 1, 2, 3, and 4 decimal for each octet.

A quadlet may contain bit fields of any length between 1 bit and 32 bits. When a field spans more than one octet, the point where it spans the octet is shown as a large tick mark as shown in Figure 3.3.

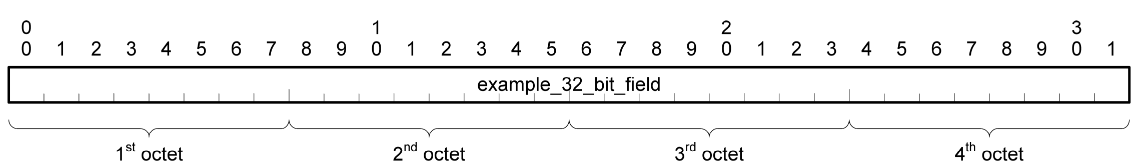


Figure 3.3—Example 32 bit field diagram

The 64 bit fields that need to be contained in more than one quadlet are shown as a series of eight octets in Figure 3.4.

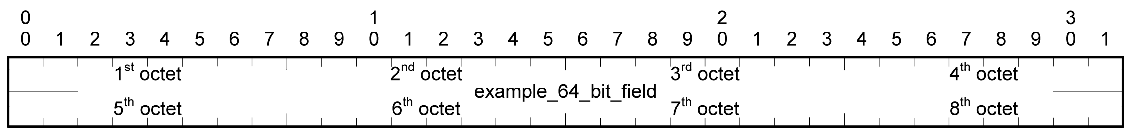


Figure 3.4—Example 64 bit field diagram

3.5.2 Field value conventions

This document describes values of fields. For clarity, names can be associated with each of these defined values, as illustrated in Table 3.1. A symbolic name, consisting of uppercase letters with underscore separators, allows other portions of this document to reference the value by its symbolic name, rather than by a numerical value.

Table 3.1—Wrap field values

Value	Name	Description
0	WRAP_AVOID	Frame is discarded at the wrap point
1	WRAP_ALLOW	Frame passes through wrap points
2 to 3	—	Reserved

A field value of TRUE shall always be interpreted as being equivalent to a numeric value of one (1) unless otherwise indicated. A field value of FALSE shall always be interpreted as being equivalent to a numeric value of zero (0) unless otherwise indicated.

4. Abbreviations and acronyms

This document contains the following abbreviations and acronyms:

AVB	audio/video bridging
AVTP	Audio/Video Transport Protocol
AVTPDU	Audio/Video Transport Protocol Data Unit
CFI	canonical format indicator
CIP	common isochronous packet
DA	Destination MAC Address
DBC	data block count
DBS	Data Block Size
DTCP	digital transmission content protection
DVCR	digital videocassette recorder
EUI	IEEE Extended Unique Identifier
FDF	format dependent field
FQTSS	Forwarding and Queuing Enhancements for Time Sensitive Streams (IEEE Std 802.1Qav)
FMT	stream format
FN	fraction number
gPTP	generalized Precision Time Protocol (IEEE P802.1AS/D7.2)
LAN	local area network
MAAP	MAC address acquisition protocol
MAC	media access control
MHz	megahertz (million cycles per second)
MIDI	Musical Instrument Digital Interface

MMA	Midi Manufacturers Association, Inc.[6]
MTU	maximum transmission unit, for this document refers to the maximum length of the MAC Client Data field defined in 3.2.7 of IEEE Std 802.3™-2008
OUI	IEEE organizationally unique identifier
PCP	priority code point
PDU	protocol data unit
QPC	quadlet padding count
Rsv	reserved
SID	source identifier
S/PDIF	Sony/Philips Digital Interconnect Format
SRP	Stream Reservation Protocol (IEEE Std 802.1Qat)
SPH	source packet header
SYT	synchronization timing
TPID	tagged protocol identifier ($81\text{-}00_{16}$ as defined in IEEE Std 802.1Q)
TSP	Transport Stream Packet
VDSPC	Video Data Source Packet Count
VID	VLAN identifier
VLAN	Virtual Local Area Network

5. Audio Video Transport Protocol (AVTP) base protocol

5.1 Overview

The AVTP specifies a protocol for audio and video data to be transported on an AVB network. AVTP is designed to take advantage of the unique advantages provided by networks that support gPTP, Stream Reservation Protocol (SRP), and FQTSS.

AVTP specifies methods to transport audio/video data and timing information so that audio/video content sent by a Talker can be reproduced on Listeners. AVTP streams are sent from a Talker and can be received by one or more Listeners.

AVTP stream data are carried directly on the underlying MAC layer (IEEE Std 802.3™-2008, IEEE Std 802.11™-2007, etc.).

This clause contains some of the basic assumptions and operation requirements of the AVTP.

5.1.1 AVTP network requirements

All devices that send, receive, or forward AVTP data are required to support the services provided by gPTP, SRP, and FQTSS. AVTP relies on these services being available to function properly. The behavior when a Talker, Listener, or bridge on the network does not support these AVB protocols is out of the scope of this standard.

AVTP makes use of gPTP to provide a network wide time base that can be used to convey timing information from a Talker to Listener(s).

AVTP makes use of SRP and FQTSS to provide reliable network delivery with bounded network latency for transporting audio/video data from Talker to Listener.

AVTP data can be sent from Talker to Listener either directly or forwarded by AVB bridges to the Listener.

[6] Midi Manufacturers Association, Inc., P.O. Box 3173, La Habra, CA 90632-3173 (http://www.midi.org email info@midi.org).

5.1.2 AVTP Ethertype

All AVTP frames shall use an Ethertype value as listed in Table 5.1.

Table 5.1—AVTP Ethertype

AVTP Ethertype
$22F0_{16}$

5.2 AVTPDU common header format

This subclause documents the fields contained in the first 12 octets of header that are common to all AVTP stream data and control AVTPDUs. These 12 octets consist of the following fields:

— **cd** (control/data) indicator: 1 bit
— **subtype**: 7 bits
— **sv** (StreamID valid) indicator: 1 bit
— **version** (AVTP version): 3 bits
— **type_specific_data**: 20 bits
— **stream_id**: 64 bits

Figure 5.1 shows these fields with offset zero (0) shown as the first octet of the AVTPDU.

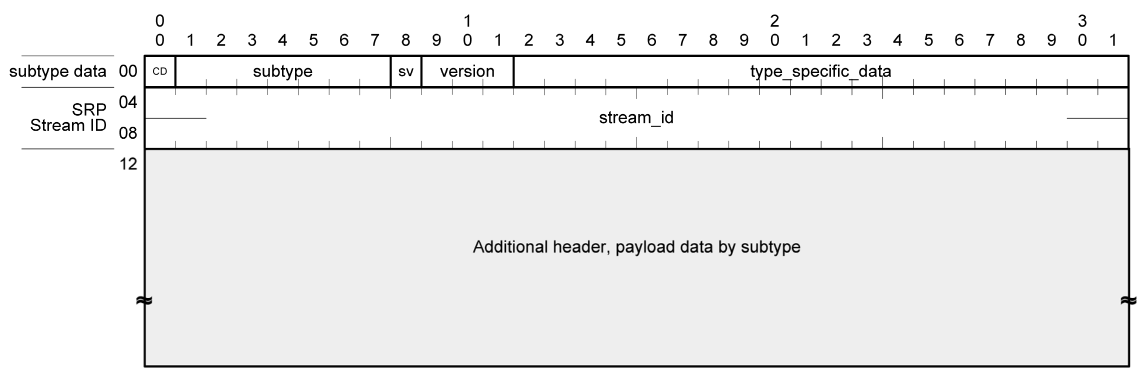


Figure 5.1—AVTPDU common header fields

NOTE—All protocol data unit (PDU) diagrams, such as Figure 5.1, include shading to indicate areas of the PDU that are discussed in the current clause. White areas are discussed in the current clause, whereas gray areas are discussed elsewhere. This shading is a feature of all PDU diagrams included in this document.[7]

5.2.1 cd (control/data indicator) field

The **cd** bit indicates whether this is a control or stream data AVTPDU.

If the **cd** bit is zero (0), then this AVTPDU is a stream data AVTPDU. In the case of stream data, the cd bit is set to zero (0) by the Talker and is not modified during the transmission of the AVTPDU through the network to the receiving listener(s).

If this field is set to one (1), then this AVTPDU is a control AVTPDU.

[7] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

See 5.3 and 5.4 for additional encapsulation and protocol rules related to this field.

5.2.2 subtype field

The 7 bit **subtype** field is used to identify the protocol being carried by AVTP. Each protocol defines its use of AVTP encapsulation within the rules established for common header formats for control and data AVTPDUs.

Currently defined subtype values are listed in Table 5.2.

Table 5.2—AVTP subtype values

Hexadecimal value	Function	Meaning
00_{16}	61883_IIDC_SUBTYPE	IEC 61883/IIDC over AVTP
01_{16}	MMA_SUBTYPE	MMA payload over AVTP
02_{16}-$7D_{16}$	—	Reserved for future protocols
$7E_{16}$	MAC address acquisition protocol (MAAP)	MAAP
$7F_{16}$	EXPERIMENTAL SUBTYPE	Experimental over AVTP

The experimental subtype is for testing purposes only. Its behavior is out of the scope of this document, and the experimental subtype shall not be used in conforming products.

Subsequent parsing of AVTPDUs shall be based on a combination of the values contained within the **subtype** and **cd** fields.

5.2.3 sv field

The **sv** field is used to indicate whether the 64 bit **stream_id** field contains a valid StreamID.

The bit is set to one (1) if the StreamID is a valid StreamID.

The bit is set to zero (0) if the StreamID is not valid.

For more details on valid combinations of the **stream_id** and **sv** fields, see 5.2.6.

5.2.4 version field

The **version** field is used to indicate what version of AVTP this AVTPDU is formatted for.

The **version** field shall be set to zero (000_2).

Values of one (001_2) through seven (111_2) are reserved for future versions of the AVTP.

If an AVTPDU is received with a reserved version number, the AVTPDU shall be ignored.

5.2.5 type_specific_data

The **type_specific_data** field is used to carry specific data based on the value of **cd**; see 5.3 and 5.4.

5.2.6 stream_id field

If the **sv** field is set to one (1), then the **stream_id** field shall contain the 64 bit StreamID associated with the AVTPDU. The **stream_id** field is used for stream identification.

All stream data AVTPDUs shall contain a valid StreamID in the **stream_id** field and shall set the **sv** (StreamID Valid) bit to one (1).

Control AVTPDUs relating to an individual stream shall contain a valid StreamID with the sv bit set to one (1).

Control AVTPDUs not related to an individual stream shall set the **stream_id** field to zero (0) and shall set the **sv** bit to zero (0).

AVTP end stations receiving control AVTPDUs with the **sv** bit set to zero (0) shall ignore the entire contents of the **stream_id** field regardless of its value.

5.3 AVTPDU common control header format

This subclause documents the common fields contained in a control AVTPDU header. An AVTPDU is designated a control AVTPDU when the **cd** bit is set to one (1). The header consists of the fields contained in the AVTP common header format and the following fields that replace the **type_specific_data**:

— **control_data**: 4 bits
— **status**: 5 bits
— **control_data_length**: 11 bits

Figure 5.2 shows these fields with offset zero (0) shown as the first octet of the control AVTPDU.

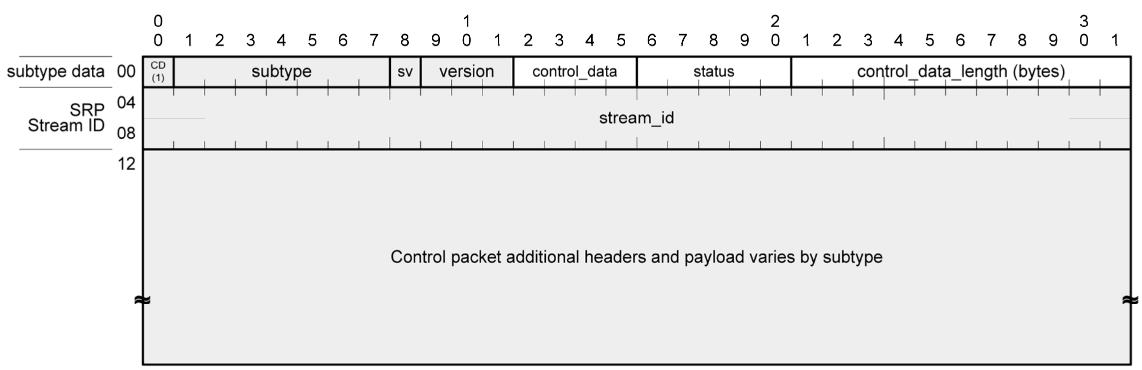


Figure 5.2—Control AVTPDU common fields

5.3.1 control_data field

The 4 bit **control_data** is available for use by the given control protocol as specified by the **subtype** field. This field may be used by the control protocol as it sees fit but is recommended for use for data such as command codes, events, etc. If not used by the given control AVTPDU, then this field shall be set to zero (0).

5.3.2 status field

The 5 bit **status** field is available for use by the given control protocol as specified by the **subtype** field. This field may be used by the control protocol as it sees fit but is recommended for use for data such as status values, flags, etc. If not used by the given control AVTPDU, then this field shall be set to zero (0).

5.3.3 control_data_length field

The 11 bit **control_data_length** field is used to contain the unsigned control AVTPDU payload length in octets of all valid data octets contained in the AVTPDU following the **stream_id** field in the control AVTPDU header.

The maximum value for this field is determined by the maximum transmission unit (MTU) and shall be set to allow all control AVTPDUs to fit in a single Ethernet frame.

5.4 AVTP common stream data AVTPDU header format

This subclause documents the common fields contained in a stream data AVTPDU header. An AVTPDU is designated a stream data AVTPDU when the cd bit is set to zero (0). The header consists of the fields contained in the AVTP common header format and the following fields that replace the type_specific_data and add additional fields needed for timestamping and other uses.

- **mr** (media clock restart): 1 bit
- **r** (reserved): 1 bit
- **gv** (gateway_info valid): 1 bit
- **tv** (avtp_timestamp_valid): 1 bit
- **sequence_num** (sequence number): 8 bits
- **reserved**: 7 bits
- **tu** (timestamp_uncertain): 1 bit
- **avtp_timestamp**: 32 bits
- **gateway_info**: 32 bits
- **stream_data_length**: 16 bits
- **protocol_specific_header**: 16 bits
- **stream_payload_data**: 0 to *n* octets (where *n* does not cause the packet to exceed the MTU)

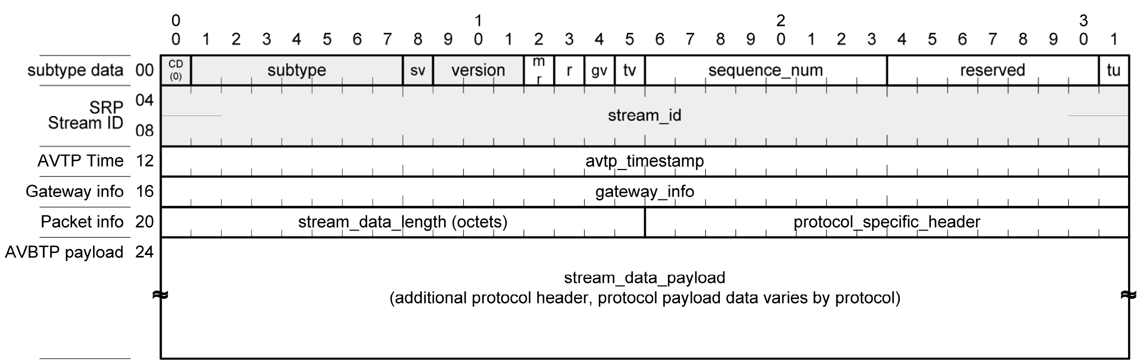


Figure 5.3—AVTP common stream header format (cd field set to zero (0))

5.4.1 mr: (media clock restart) field

The media clock restart field (**mr**) is a 1 bit field used by the Talker to indicate that there has been a change in the source of the media clock. An example of this would be a change from one audio/video input to a different audio/video input where the media clock is supplied from the input (i.e., switching from one

Sony/Philips Digital Interconnect Format (S/PDIF) input to a different S/PDIF input). This bit may be used by the Listener to adjust quickly to the new media clock. This kind of change need not be seamless but allows the Listener to take any appropriate action to minimize the disruption to the stream.

This bit is toggled by the Talker each time a media clock restart is needed, and it stays at its new value until a new media clock restart is needed. This toggle ensures the media clock restart will not be missed if it were to appear in a single AVTPDU that just happens to be dropped. Once this bit toggles, it must remain in its new state for a minimum of eight AVTPDUs for a given stream.

5.4.2 r: (reserved) field

The reserved bit field (**r**) is a 1 bit field reserved for future use in stream data AVTPDU headers.

This field is set to zero (0) on transmit and ignored on receive.

5.4.3 gv: (gateway_info field valid) field

The **gateway_info** field valid (**gv**) field is reserved for use by AVTP gateways that are not defined by this standard.

AVTP end stations that do not have an AVTP gateway function shall set this field to zero (0) on transmit and ignore this field on receive.

5.4.4 tv: (avtp_timestamp_valid) field

The avtp_timestamp_valid (**tv**) field is a 1 bit field used to indicate the validity of the **avtp_timestamp** field time value.

If the avtp_timestamp_valid bit is set to zero (0) by the AVTP Talker, then this field shall indicate that the **avtp_timestamp** field contains no valid data and therefore shall be ignored by an AVTP Listener.

If the avtp_timestamp_valid bit is set to one (1) by the AVTP Talker, then this field shall indicate that the **avtp_timestamp** field is valid.

For how the **avtp_timestamp** field is interpreted and processed, see 5.4.8.

5.4.5 sequence_num: (sequence number) field

The sequence number bit field (**sequence_num**) is an 8 bit field used to indicate the sequence of AVTPDUs in a stream by the Talker. This field may start at any value for the first AVTPDU of a stream and shall be incremented by one (1), with wrapping, on each subsequent AVTPDU of a stream. This field is to wrap from 11111111_2 to 00000000_2 (FF_{16} to 00_{16}).

The value of this field is determined by the Talker. Listeners can use the sequence number to detect AVTPDUs lost in transmission.

A Listener may join a stream at any time after the stream has started, and therefore, a Listener must be able to tolerate a starting **sequence_num** of any value.

5.4.6 reserved: (reserved) field

The **reserved** field is a 7 bit field reserved for future use in stream data AVTPDU headers. This field is set to zero (0) on transmit and ignored on receive.

5.4.7 tu (timestamp_uncertain) field

The 1 bit **tu** field shall be set by the Talker when the attached gPTP system reports any significant problem with its synchronization system. Examples might be the invocation of the best master clock algorithm because of a Sync Timeout or detection of an unacceptable phase transient by circuitry independent of the gPTP state machine. Thus, it is used by Talkers to indicate that the timestamps may not be globally synchronized with network time. The bit may be used by Listeners, possibly in conjunction with their knowledge of the status of the gPTP clock, to prevent unacceptable disturbances in the recovered media streams.

5.4.8 avtp_timestamp field

The 32 bit **avtp_timestamp** field contains the AVTP presentation time (5.5.2) if the avtp_**timestamp_valid** bit is set to one (1). The **avtp_timestamp** field shall be in units of nanoseconds and has a maximum value of 2^{32}-1 ns. The **avtp_timestamp** field rolls over approximately every 4 s.

NOTE—The **avtp_timestamp** field is not simply the gPTP **nanoseconds** field. The **avtp_timestamp** field is given by the following formula:

$$\textbf{avtp\_timestamp} = (\textbf{AS\_sec} \times 10^9 + \textbf{AS\_ns}) \text{ modulo } 2^{32}$$

where

AS_sec is the gPTP **seconds** field
AS_ns value is the gPTP **nanoseconds** field

If the avtp_timestamp_valid bit is zero (0), then the content of the **avtp_timestamp** field is undefined and shall be ignored.

5.4.9 gateway_info field

The 32 bit **gateway_info** field is reserved for use by AVTP gateways that are not defined by this standard.

AVTP end stations that do not have an AVTP gateway function shall set this field to zero (0) on transmit and ignore this field on receive.

5.4.10 stream_data_length field

The 16 bit **stream_data_length** field indicates the unsigned count of the stream's AVTPDU payload length in octets of all valid data octets contained in the AVTPDU following the **protocol_specific_header** field in the stream data AVTPDU header.

The maximum value for this field is determined by the MTU and shall be set to allow all stream data AVTPDUs to fit within a single standard frame.

NOTE—This field is sized at a full 16 bits to allow for easier frame size handling for AVTP gateway functions for other networks such as IEEE Std 1394-2008, which allows larger frame sizes than is allowed on IEEE 802.3 networks and also matches the position and function relative to other fields of the IEEE 1394 equivalent field for isochronous frames.

5.4.11 protocol_specific_header field

The 16 bit **protocol_specific_header** field shall carry 16 bits of protocol specific data as specified by the protocol subtype.

See 6.2 for an example of how IEC 61883 makes use of this field.

5.4.12 stream_data_payload field

The **stream_data_payload** field consists of additional protocol specific data. The valid data length is indicated by the **stream_data_length** field.

5.5 Timing and synchronization

5.5.1 General

AVTP defines a presentation time to achieve timing synchronization between a Talker and Listener(s). The presentation time represents in nanoseconds the gPTP time when the data contained in the AVTPDU are to be available to the AVTP Listener(s).

AVTP presentation time is used as a reference to synchronize any necessary media clocks and to determine when the first sample of a stream is to be presented to the client. Because media clocks vary with audio/video types, the exact usage of the AVTP presentation time is media format dependent.

5.5.2 AVTP presentation time

The AVTP presentation time is contained in the **avtp_timestamp** field of stream data AVTPDUs.

The AVTP presentation time may not be valid in every AVTPDU. If an AVTPDU contains a valid timestamp, then the **tv** (avtp_timestamp_valid) bit shall be set to one (1).

The AVTP presentation time represents the timestamp of when the media sample was presented to AVTP at the Talker plus a constant, Max Transit Time, to compensate for network latency. Network latency is dependent on the network configuration and speed. Max Transit Time represents the worst-case network latency assumed for a given configuration. It is possible for a Talker and a Listener to determine the Max Transit Time value to use in a given stream. The mechanism for a Talker and a Listener to negotiate Max Transit Time is outside the scope of this standard. Unless otherwise negotiated between the Talker and the Listener, the Max Transit Time to calculate the AVTP presentation time is defined by the Default Max Transit Time in Table 5.3.

Table 5.3—AVTP default Max Transit Time

Class	Max Transit Time (ms)
A	2.0
B	50.0

NOTE—The Max Transit Time's numbers are a function of the speed, size (hops), and configuration of the IEEE 802.1 AVB Cloud. The Max Transit Time can be determined by the maximum latency per hop calculations as defined in SRP (SRP, subclause 35.2.2.8.6) and is reported by SRP to the Listener as Accumulated Latency. The Default Max Transit Time is set to support up to seven Ethernet (IEEE Std 802.3-2008) hops for SR class A, and up to two WiFi (IEEE Std 802.11-2007) hops plus six Ethernet (IEEE Std 802.3-2008) hops for SR class B.

The AVTP presentation time as received by the Listener(s) in the **avtp_timestamp** field is utilized to synchronize the media clock of the Listener to the Talker. Because the AVTP presentation time is directly related to the gPTP global time, it can also be used to synchronize multiple Listeners.

5.5.3 timestamp_uncertain

Although the gPTP time is intended to be stable, it is possible for there to be discontinuities in the gPTP time. These could be caused by events such as changes in the identity of the gPTP Grandmaster clock, changes in the timing source of the Grandmaster clock, or other events encountered by a Talker.

When a Talker detects a discontinuity, either from an gPTP indication or simple observation, it should set the timestamp_uncertain field to one (1). This indicates to the Listener(s) of the stream that the AVTP presentation times contained in the **avtp_timestamp** field may for a limited period of time not correspond to the gPTP time and the Listener needs to take appropriate action. Once the Talker has determined that the gPTP clock has returned to a normal state, the timestamp_uncertain field should be reset to zero (0).

When a Listener detects that the timestamp_uncertain field has been set to 1 or detects a discontinuity, either from an gPTP indication or simple observation, it should stop attempting to correlate AVTP presentation time to gPTP time until the source of the discontinuity has ceased and the Listener has allowed the appropriate time for the gPTP clock to stabilize.

The time required for the gPTP clock to stabilize is implementation dependent and therefore outside the scope of this standard.

5.5.4 Presentation time measurement points

This subclause defines the measurement points to be used when calculating the AVTP presentation time. In general, the presentation time marks the time that the IEEE 1722 layer transfers the timestamped data to the next processing layer in the stack.

The AVTP presentation time is the gPTP time by which the stream data will be available in the Listener to be read and/or processed by the next function in the Listener. This is shown as the Presentation Time Reference Plane in Figure 5.4. The presentation time in a Listener is used to know when to start processing (e.g., playing) a new stream and is used to recover the stream's media clock.

A Talker sets the presentation time in all AVTPDUs such that a Listener can know when to start processing (e.g., playing) a new stream and the Listener can use the presentation timestamps to recover the stream's media clock. This requires the Talker to set the presentation time "in the future" by the Max Transit Time amount defined in 5.5.2. The starting point in the Talker for this Max Transit Time is the Ingress Time Reference Plane (Figure 5.4). It is the point in a Talker where the "ownership" of the IEEE 1722 frame is transferred to the Talker's MAC such that the MAC is now allowed to and is able to transmit the frame (assuming the MAC is not in the middle of transmitting the previous Ethernet frame).

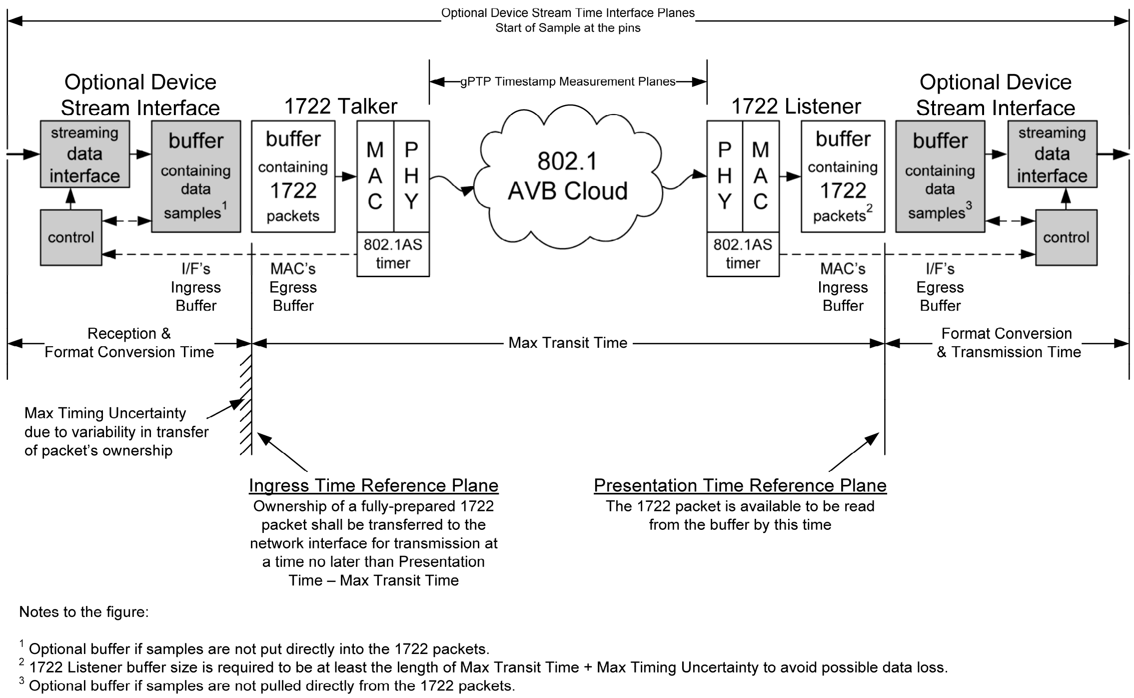


Figure 5.4—Presentation time measurement point

The Talker shall transfer control of AVTP stream frames to the MAC layer no later than the Presentation Time–Max Transit Time. If the Talker fails to transfer an AVTP stream frame before (Presentation Time – Max Transit Time), then the AVTP frame may not arrive at the Listener before it is required for audio/video playback. This may result in loss of portions of the media data stream. In any design, there is always some uncertainly about exactly when this Ingress Time Reference Plane crossing will occur. It is the Talker's responsibility to know the Timing Uncertainty of its own design and to begin the transfer of ownership such that, in the worst case, the crossing of the Ingress Time Reference Plane will occur no later than (Presentation Time – Max Transit Time).

A Talker's Timing Uncertainty is a combination of many factors, including its own local timer resolution and its worst-case response time to a timer event. A Talker shall be designed such that its Timing Uncertainty is no more than the Max Timing Uncertainty amount defined in Table 5.4.

The buffering required in a Listener is used to absorb a stream for the amount of time specified as the sum of (Max Transit Time + Max Timing Uncertainty + Max gPTP) Skew Time (defined in B.3 of IEEE P802.1AS/D7.2) for the class of the stream. If the Listener does not have sufficient buffering for a given stream, a buffer overflow could occur, resulting in data loss.

The Max Transit Time is a function of the speed, size (hops), and configuration of the AVB Network (5.5.2).

Timing Uncertainty is the potential delay introduced by the transfer of ownership of the frame from the IEEE 1722 layer to the media interface. The Max Timing Uncertainty is the maximum amount of transfer delay allowed.

Figure 5.4 shows Optional Device Stream Interfaces on both the Talker and the Listener. As shown, these interfaces could be media interfaces that connect to physical devices. These media interfaces are optional and outside the scope of this standard. Any stream conversion time required for these interfaces (e.g., buffering time, audio/video codec delay times, etc.) are outside of and in addition to the AVTP Max Transit Time + Max Timing Uncertainty.

Table 5.4—AVTP Max Timing Uncertainty

Class	Max Timing Uncertainty (µs)
A	125
B	1000

5.5.5 Protocol layering and other required protocols

This clause documents other required protocols and standards required, but not specified, in this transport level specification.

All AVTP end stations shall support the following standards:

— Generalized Precision Timing Protocol
— Stream Reservation Protocol
— Forwarding and Queuing Enhancements for Time Sensitive Streams

6. IEC 61883/IIDC over AVTP

6.1 Overview

This clause describes the AVTPDU formats to adapt the IEC 61883 protocols to an IEEE 802 network, as well as to adapt the protocols to work with timing and synchronization based on IEEE P802.1AS/D7.2. This clause does not attempt to document the IEC 61883 protocols, only the changes needed to work in an IEEE 802.1 AVB environment.

AVTP adapts the following IEEE 1394/IEC 61883 type protocols to run in an IEEE 802 environment:

— IEC 61883-2:2004: SD-DVCR data transmission
— IEC 61883-4:2004: MPEG2-TS data transmission
— IEC 61883-6:2005: Audio and music data transmission protocol
— IEC 61883-7:2003: Transmission of ITU-R BO.1294 System B
— IEC 61883-8:2008: Transmission of ITU-R BT.601 style Digital Video Data
— IIDC 1394-based Digital Camera Specification [B8]

6.2 Common IEC 61883/IIDC Stream data encapsulation

The IEC 61883/IIDC stream data encapsulation is used for carrying IEC 61883 and IIDC stream data traffic over AVTP networks.

This encapsulation uses a **cd** field of zero (0) and a **subtype** field of zero (0).

This encapsulation also uses the **protocol_specific_header** to contain four fields that are common for both IIDC and IEC 61883 packets. These fields are modeled after IEEE Std 1394-2008 and consist of the following:

— **tag** field: 2 bits

- **channel** field: 6 bits
- **tcode** field: 4 bits
- **sy** field: 4 bits

These fields are shown in Figure 6.1.

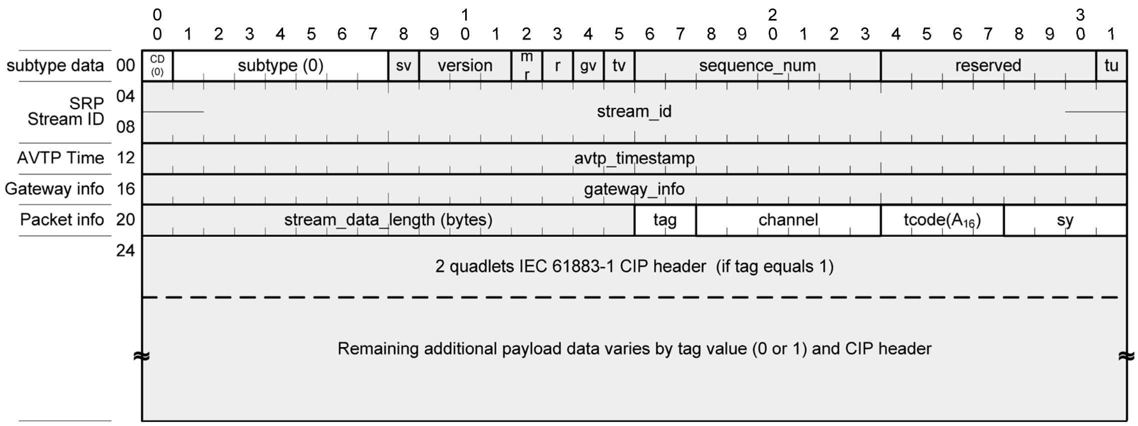


Figure 6.1—IEC 61883/IIDC common header fields

6.2.1 tag field

The 2 bit **tag** field follows the same meaning, format, and rules as specified by IEEE Std 1394-2008. Of the four possible combinations for this field, the following values are supported or not supported as specified below.

Supported by AVTP:

- 00_2: No CIP header included (used by Instrumentation & Industrial Digital Camera (IIDC) 1394 trade association specification)
- 01_2: CIP header included

Not supported by AVTP:

- 10_2: Reserved by IEEE 1394.1 clock adjustment [B7]
- 11_2: Global asynchronous stream packet format (used in IEEE Std 1394-2008 for Serial Bus to Serial Bus bridges)

6.2.2 channel field

The 6 bit channel field follows the same meaning, format, and rules as specified by IEEE Std 1394-2008. The following values are supported as specified:

- 0 to 30 and 32 to 63: Originating channel ID from an IEEE 1394 serial bus via IEEE 1394/IEC 61883 to IEEE 1722/IEC 61883 gateway
- 31: Originating source is on AVB network (native AVB)

6.2.3 tcode (type code)

The 4 bit **tcode** field follows the same meaning, format, and rules as specified by IEEE Std 1394-2008. For AVTP, the only value supported shall be a fixed value of 1010_2 binary (A_{16} hexadecimal, same as IEEE 1394 isochronous packet format) with the following rules for Talkers and Listeners:

- AVTP Talkers shall always set this field to A_{16} hexadecimal on transmit
- AVTP Listeners shall always ignore this field on receive

6.2.4 sy field

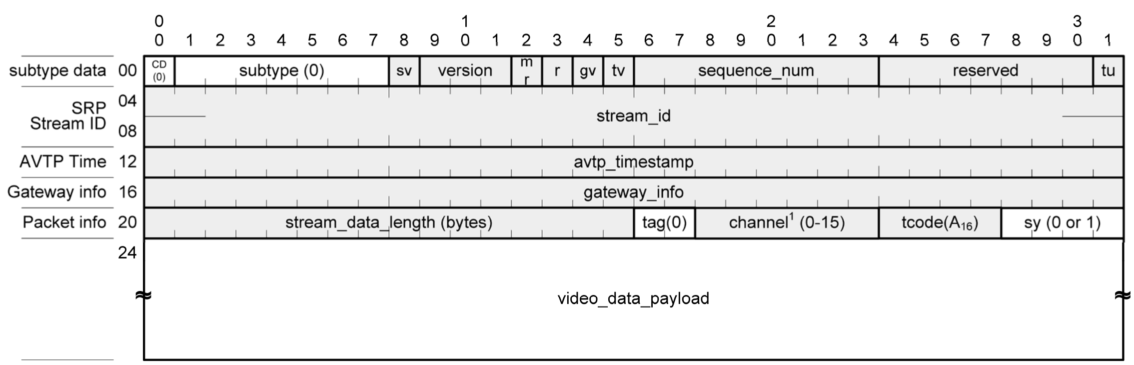
Use of the 4 bit sy field is application specific and therefore beyond the scope of this standard. The known industry standards that currently use this field are as follows:

- IIDC 1394-based Digital Camera Specification [B8] (used for video start of frame indicator)
- Digital Transmission Content Protection (DTCP)[8] (DTCP Specification [B2],[9] DTCP Volume 1 Supplement [B3])

6.2.5 IIDC encapsulation

This subclause describes the AVTPDU format to be used when there is no CIP header included in the AVTPDU (Figure 6.2). This option is selected by the tag value being set to zero (0). This adds the following additional field to the IEC 61883/IIDC common header:

- **video_data_payload**: 0 to *n* octets



Notes to figure:

[1] Current standard for IIDC restricts channel ID to 0 - 15

Figure 6.2—IIDC AVTPDU header fields

6.2.5.1 video_data_payload field

The **video_data_payload** field consists of 0 to *n* octets of additional protocol-specific data. The valid data length is indicated by the **stream_data_length** field.

The **video_data_payload** field is defined in IIDC 1394-based Digital Camera Specification [B8].

[8] Available at http://www.dtcp.com.
[9] The numbers in brackets correspond to those of the bibliography in Annex A.

6.2.6 IEC 61883 encapsulation

This subclause describes the AVTPDU format to be used when a CIP header is included in the AVTPDU (Figure 6.3 and Figure 6.4). This option is selected by the tag value being set to one (1).

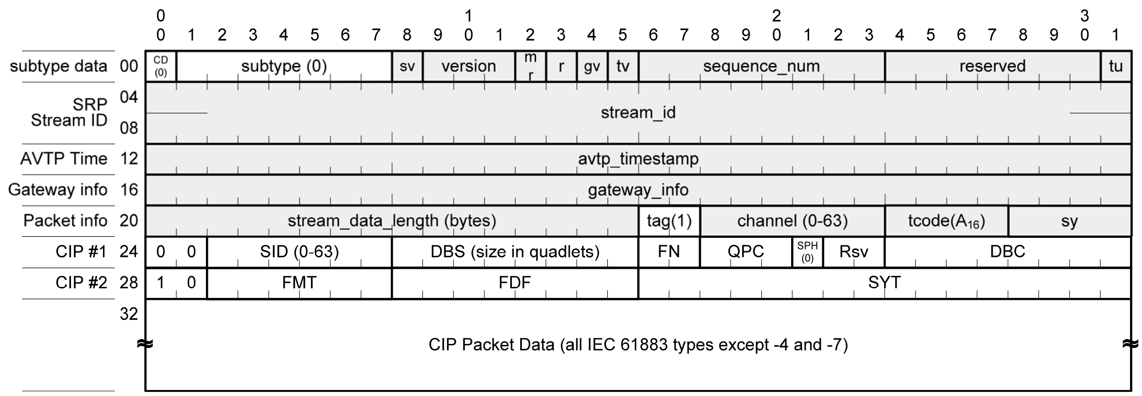


Figure 6.3—CIP header and data fields, tag = 1, SPH = 0

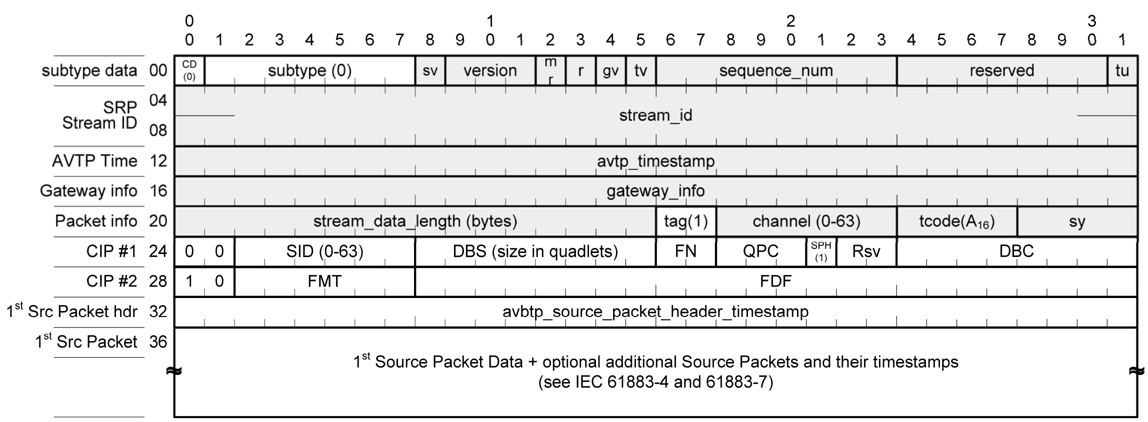


Figure 6.4—CIP header and data fields, tag = 1, SPH = 1

6.2.6.1 CIP header 1st quadlet indicator

The 2 bit CIP header 1st quadlet indicator field has the same definition as defined in IEC 61883-1. The AVTP shall only support a fixed value of 00_2 set by Talkers and ignored by Listeners.

6.2.6.2 Source identifier (SID) field

The 6 bit **SID** field has the same definition as defined in IEC 61883-1. For the AVTP, it shall use the following values:

— 0 to 62: Originating Source ID from an IEEE 1394 serial bus (frame originated from an IEEE 1394/IEC 61883 to IEEE 1722/IEC 61883 interworking unit)

— 63: Originating source is on the AVB network

6.2.6.3 Data Block Size (DBS)

The 8 bit DBS field has the same definition as currently in IEC 61883-1:2003, which is the size of Data Blocks in quadlets, as follows:

- 0: 256 quadlets
- 1 to 255: 1 to 255 quadlets

For the AVTP, the DBS field shall not be set to a value that would cause the AVTPDU length to exceed the MTU of the network. All AVTPDUs of this format shall contain an integral number of data blocks as defined in IEC 61883-1:2003.

6.2.6.4 Quadlet padding count (QPC)

The 3 bit **QPC** field has the same definition as currently defined in IEC 61883-1:2003. For all types of IEC 61883 as defined today, this field is always zero.

6.2.6.5 Fraction number (FN) field

The 2 bit **FN** field has the same definition as currently defined in IEC 61883-1. This is currently only used in IEC 61883-4:2004 and IEC 61883-7:2003 (where also the **SPH** field is always set to one (1)).

6.2.6.6 Source packet header (SPH) field

The **SPH** bit has the same definition as currently defined in IEC 61883-1:2003.

If set to one (1):

- The AVTPDU contains IEC 61883-4:2004 or IEC 61883-7:2003 (or future) source packets with source packet headers.

If set to zero (0):

- Then AVTPDU does not contain source packets (contains integer number of Data Blocks).

6.2.6.7 Reserved (Rsv) field

The 2 bit Rsv field has the same definition as currently defined in IEC 61883-1:2003. It is reserved (currently not used by IEEE Std 1394-2008/IEC 61883), set to zero (0) on transmit and set to ignore on receive.

6.2.6.8 DBC field

The 8 bit DBC field has the same definition as currently defined in IEC 61883-1:2003. It contains the sequence number of the first data block in the AVTPDU.

6.2.6.9 CIP header second quadlet indicator

The 2 bit CIP header second quadlet indicator field has the same definition as defined in IEC 61883-1:2003. For the AVTP, it shall be fixed at 10_2 binary.

6.2.6.10 Stream format (FMT) field

The 6 bit **FMT** field has the same definition as currently defined in IEC 61883-1:2003.

6.2.6.11 Format dependent field (FDF)

The **FDF** field has the same definition as defined in IEC 61883-1:2003. If the **SPH** field is set to 0, then this field is 8 bits in length. If the **SPH** field is set to 1, then this field is 24 bits in length.

6.2.6.12 Synchronization timing (SYT) field

The 16 bit **SYT** field is only present if the **SPH** field is set to a value of zero (0). This IEC 61883-1-defined field is present but not used by AVTP end stations; it is only used by IEEE 1394/IEC 61883 to IEEE 1722/IEC 61883 AVTP gateways. AVTP Talker end stations shall set this field if present to $FFFF_{16}$ (the IEC 61883 no data timestamp value) on transmit, and AVTP Listener end stations shall ignore this field on receive.

6.2.6.13 avtp_source_packet_header_timestamp field

If the SPH field is set to a value of one (1), then the stream data AVTPDU contains one or more CIP source packets. Each CIP source packet contains a 32 bit timestamp as the first quadlet of the source packet followed by an integral number of quadlets as defined by the standard that defines the source packet format.

Currently defined formats that are supported by AVTP are IEC 61883-4:2004 and IEC 61883-7:2003.

IEC 61883-4:2004 and IEC 61883-7:2003 as written define the source packet header quadlet to contain the presentation time of the packet based on IEEE 1394 cycle time.

AVTP uses the same formats as defined in IEC 61883-4:2004 and IEC 61883-7:2003, but it uses a gPTP-based presentation time in the same format as defined for the **avtp_timestamp** field (5.4.8). For these AVTPDUs, the Talker shall set the **tv** (avtp_timestamp_valid) bit to zero (0) for all AVTPDUs (i.e., the AVTP presentation time is set in the source packet header instead of in the **avtp_timestamp** field of the AVTPDU common header).

6.3 IEC 61883 control AVTPDU format

Use of AVTPDUs with a subtype of 61883_IIDC_SUBTYPE (Table 5.2) and a **cd** bit of one (1) is reserved for possible future versions of this specification.

6.4 Timing and synchronization

6.4.1 General

The processing of presentation time, timing, and synchronization for IEC 61883 over AVTP is generally accomplished in the same manner as specified in IEC 61883-1:2003 through IEC 61883-7:2008 over IEEE Std 1394-2008 for processing of presentation time. The main difference between running IEC 61883 protocols over AVTP versus IEEE Std 1394-2008 is the use of gPTP-based time expressed in 32 bits of nanoseconds versus the use of IEEE Std 1394-2008 time expressed in seconds, cycles, and 24.576 MHz cycle offset values.

The usage of the timing and synchronization information included in the CIP header or Source Packet Header is generally consistent with the definition in the IEC 61883 series of standards where the main differences are as follows:

- The functionality of the **SYT** field is replaced by the **avtp_timestamp** field.
- The Source Packet Header is expressed in gPTP-based nanoseconds.

— For protocols of IEC 61883-4:2004 and IEC 61883-7:2003 where the SPH field is set to one (1), the 32 bit source packet header field for each source packet contains the associated presentation time of the source packet expressed in gPTP-based nanoseconds instead of IEEE 1394-based seconds, cycle, and cycle offsets.
— In the IEC 61883 series of specifications the term “receiver” is used, whereas in this specification, the term “Listener” is used.
— In the IEC 61883 series of specifications, the term “transmitter” is used, whereas in this specification, the term “Talker” is used.
— IEEE Std 1394-2008 on networks faster than 100 megabits per second allows CIP packets larger than can fit in a single standard Ethernet frame. This standard does not. If an application needs to send more data than can fit in a single Ethernet frame, it shall generate multiple correctly formatted CIP packets, each of which can fit in a standard Ethernet frame.
— Some of the IEC 61883 series of specifications sometimes have options that differentiate “professional,” “consumer-use,” or “cost-sensitive” equipmentThis standard does not, and if a device supports a given format, it shall support all mandatory requirements for that format specified in this document.

  Example: This specification documents the **avtp_timestamp** field function. All AVTP Talkers and Listeners must support the field and the processing thereof. As a contrast, the SYT field from IEEE Std 1394-2008 in IEC 61883 allows different processing based on the class of equipment.

NOTE—This standard does not require a Talker to synchronize frame transmission to an interval of 125 µs for SR class A streams nor to 250 µs for SR class B streams. However, using a period other than integer subdivisions of these class measurement intervals (i.e., /1, /2, /3, /4, etc.) will result in extra bandwidth being reserved, which is therefore unavailable for use by other streams (extra bandwidth is usable by best effort traffic). For example, an SR class A Talker can send a single frame every 125 µs if it sets MaxIntervalFrames to 1 in the MSRP reservation, or it can send a packet every 159 µs with MaxIntervalFrames still equal to 1, or every 12.5 µs by setting MaxIntervalFrames to 10, or every 100 µs by setting MaxIntervalFrames to 2. An IEEE 1722 Talker can choose to send a single audio sample at a time if it ensures that extra overhead is accounted for in the MSRP reservation. The bandwidth, in octets per second, reserved by a Talker, can be calculated a (((MaxFrameSize + media dependent frame overhead) × MaxIntervalFrames) / class measurement interval).

6.4.2 IEC 61883-2 timing and synchronization

IEC 61883-2-formatted AVTPDUs shall follow the same timing and synchronization rules as defined in Clause 6 of IEC 61883-2:2004 but using gPTP time in the **avtp_timestamp** field instead of IEEE 1394-based cycle time in the SYT field. AVTP Talkers shall set the CIP header **SYT** field to all ones for all transmitted CIP packets and AVTP Listeners shall ignore the **SYT** field.

Replace Clause 6 of IEC 61883-2 with:

> The transmitter shall transmit a valid timestamp value in the **avtp_timestamp** field once every video frame period. The timestamp shall be transmitted in an AVTPDU that meets the following conditions:
>
> — packet_arrival_time_L ≤ timestamp value
> — timestamp value – transmission_delay_limit ≤ packet_arrival_time_F
>
> where
>
> packet_arrival_time_F is the gPTP time when the first bit of the packet that has the timestamp has arrived at the receiver
>
> packet_arrival_time_L is the gPTP time when the last bit of the packet that has the timestamp has arrived at the receiver

transmission_delay_limit is equal to AVTP Max Transit Time (5.5.2)

In case of *Hx* (H = 1,2,4) transmission, KH data blocks are transmitted in a video frame period M using K isochronous packets. Isochronous packet n contains H data blocks of nH, $nH + 1$, … and $(n + 1)H – 1$.

The isochronous packet n of a video frame period M shall be transmitted on the following conditions ($n = 0, \ldots, K – 1$):

- Packet_arrival_time_L ≤ nominal timing for isochronous packet n
- Nominal timing for isochronous packet n – transmission_delay_limit ≤ packet_arrival_time_F

  where

  packet_arrival_time_F is the gPTP time when the first bit of the isochronous packet n has arrived at the receiver

  packet_arrival_time_L is the gPTP time when the last bit of the isochronous packet n has arrived at the receiver

K is the number of isochronous packets without empty packets in a video frame period.

K = 250 (525-60 system)
K = 300 (625-50 system)

Nominal timing for isochronous packet $n = T_M + (T_M + 1 – T_M) \times n/K$

T_M is the timestamp for video frame period M transmitted in the **avtp_timestamp** field.

6.4.3 IEC 61883-4 timing and synchronization

The timing and synchronization of IEC 61883-4:2004 over AVTP uses the same method as used in IEC 61883-4:2004 over IEEE 1394 networks in that the source packet header quadlet always contains a valid timestamp. In IEC 61883-4:2004 over AVTP, this quadlet contains nanosecond gPTP-based time instead of IEEE 1394 seconds, cycle, and cycle offset time.

For IEC 61883-4-formatted AVTPDUs, replace Clause 4.3 of IEC 61883-4:2004 with the following:

> The timestamp in the source packet header is used by receivers for reconstructing a correct timing of the Transport Stream Packets (TSPs) at their output. The timestamp indicates the intended delivery time of the first bit/octet of the TSP from the receiver output to the transport stream target decoder. The timestamp represents the 32 least significant bits of the gPTP time at the transmitter at the moment the first bit/octet of the TSP arrives from the application, plus a constant offset. The offset is equal to the overall delay of the TSP between the moment of arriving (of the first bit) and the moment the TSP (first bit) is delivered by the receiver to the application. The value of this offset is Max Transit Time defined in 5.5.2.

6.4.4 IEC 61883-6 timing and synchronization

Replace 7.2 of IEC 61883-6:2005 with the following:

> In the case where a CIP packet contains multiple data blocks, it is necessary to specify which data block of the CIP corresponds to the gPTP-based AVTP timestamp.

The transmitter prepares the timestamp for the data block that meets this condition:

mod(data block count, SYT_INTERVAL) = 0

where

data block count is the running count of transmitted data blocks.

SYT_INTERVAL denotes the number of data blocks between two successive valid **avtp_timestamp** fields, which includes one of the data blocks with a valid **avtp_timestamp** field. For example, if there are three data blocks between two valid **avtp_timestamp** fields, then the SYT_INTERVAL would be 4.

The receiver can derive the index value from the DBC field of a CIP with a valid **avtp_timestamp** field using the following formula:

index = mod((SYT_INTERVAL – mod(DBC, SYT_INTERVAL)), SYT_INTERVAL)

where

index is the sequence number.

SYT_INTERVAL denotes the number of data blocks between two successive valid **avtp_timestamp** fields, which includes one of the data blocks with a valid **avtp_timestamp** field.

DBC is the data block count field of a CIP.

The receiver is responsible for estimating the timing of data blocks between valid timestamps.

The method of timing estimation is implementation dependent.

Replace 7.3 of IEC 61883-6:2005 with the following:

A data block contains all data arriving at the transmitter within an audio sample period. The data block contains all the data that make up an "event."

The transmitter shall specify the presentation time of the event at the receiver. A receiver shall have the capability of presenting events at the time specified by the transmitter.

The transmitter shall add TRANSFER_DELAY to the quantized timing of an event to construct the presentation time. The TRANSFER_DELAY value is initialized with the DEFAULT_TRANSFER_DELAY value. Note that for all transmitters, the TRANSFER_DELAY may be changed to achieve a shorter TRANSFER_DELAY value to allow for a shorter time if the end-to-end delay in the AVB network can allow it.

The DEFAULT_TRANSFER_DELAY value is Max Transit Time defined in 5.5.2.

6.4.5 IEC 61883-6 event types

Clause 8.2 of IEC 61883-6:2005 describes the 24 bit * 4 audio pack. 24 bit * 4 audio pack formatted audio data is not supported by AVTP.

6.4.6 IEC 61883-7 timing and synchronization

Replace 5.1.3 of IEC 61883-7:2003 with

> The source packet header field is a one quadlet field (4 octets) that represents a gPTP-based timestamp.
>
> The timestamp is used by IEC 61883-7-capable AVTP receivers for reconstructing a correct timing of the transport stream packets at their output. The timestamp indicates the intended delivery time of the first bit/octet of the transport stream packets from the receiver output to the Transport Stream Target Decoder. The timestamp represents the least significant 32 bit binary time of the gPTP-based clock at the moment the first bit/octet of the transport stream packet arrives from the application, plus an offset that is equal to the overall delay of the transport stream packet between the moment of arriving (of the first bit) and the moment the transport stream packet (first bit) is delivered by the receiver to the application.
>
> The default value of this offset Max Transit Time is defined in 5.5.2.

6.4.7 IEC 61883-8 timing and synchronization

Replace 4.3 of IEC 61883-8:2008 with the stream data AVTPDU format for IEC 61883 as shown in Figure 6.1.

For 4.4 of IEC 61883-8:2008,

Change:

> The SYT field is encoded as defined in IEC 61883-1:2003.

To:

> The SYT field is set to all ones on transmit and ignored on receive. The function of the SYT field is replaced by the **avtp_timestamp** field.

For 4.6.1 of IEC 61883-8:2008,

Change:

> For a stream that conforms to this specification, each IEEE 1394 isochronous packet consists of the CIP header followed by zero or more source packets.

To:

> For a stream that conforms to this specification, each AVTPDU consists of an AVTPDU header, followed by the CIP header followed by zero or more source packets.

For 4.2.6.2 of IEC 61883-8:2008,

Change:

> An IEEE 1394 isochronous channel that is used to transmit data according to this specification shall only transmit a single stream of video per IEEE 1394 isochronous channel.

To:

An AVTP Stream (identified by a unique 64 bit StreamID) that is used to transmit data according to this specification shall only transmit a single stream of video per that stream.

For IEC 61883-8:2008, replace 4.7.1.1 with:

When a nonempty AVTPDU is ready to be transmitted, the transmitter shall transmit it as soon as possible within the bounds of the stream transmission as defined by FQTSS. The behavior of AVTPDU transmission depends on the definition of the condition in which “a nonempty AVTPDU is ready to be transmitted.” The two situations in which this condition is defined are as follows:

a) A nonempty AVTPDU being ready for transmission is defined to be true if one or more Video Data source packets have arrived within the transmission bounds as defined by FQTSS. This transmission method is called Nonblocking Transmission and is described in 4.7.1.2 of IEC 61883-8:2008.
b) The condition of “a nonempty AVTPDU is ready to be transmitted” can also be defined as true when a fixed number of data blocks have arrived. This transmission method is called Blocking Transmission and is described in 4.7.1.3 of IEC 61883-8:2008.

   Because there is no SPH, there is only one timestamp, and this is in the **avtp_timestamp** field of the AVTPDU header. If a CIP contains multiple Video Data source packets, it is necessary to specify which source packet corresponds to the timestamp. Because the stream contains a SIM Source Packet at the frequency of once per frame, a mechanism is required to ensure that the AVTP timestamp is generated at a regular interval of Video Data Source Packets. The Video Data Source Packet Count (VDSPC) field in the Video Data Source Packet is used for this purpose. The transmitter prepares the timestamp for the Video Data Source Packet, which meets this condition:

   mod(VDSPC, SYT_INTERVAL) = 0

   where

   VDSPC is the running count of transmitted video data source packets.

   SYT_INTERVAL denotes the number of video data source packets between two successive valid SYT timestamps, which includes one of the video data source packets with a valid SYT. For example, if there are three video data source packets between two valid avtp_timestamp fields, then the SYT_INTERVAL would be 4. The SYT_INTERVAL is dependent on the video mode and color space used. The values of SYT_INTERVAL are given in Table 1 of IEC 61883-8:2008.

The receiver knows the video data source packet for which the **avtp_timestamp** field is valid because it is the source packet whose VDSPC satisfies the following equation:

mod(VDSPC, SYT_INTERVAL) = 0

The receiver is responsible for estimating the timing of data blocks between valid timestamps. The method of timing estimation is implementation dependent.

The **avtp_timestamp** field specifies the presentation time of the video data source packet at the receiver. A receiver shall have the capability of presenting events at the time specified by the transmitter.

The TRANSFER_DELAY value is derived by replacing the IEEE 1394 transmission worst-case delay with the AVTP Max Transit Time (5.5.2). On an AVTP network, it is also possible to replace the values for Allowance for Encryption, Allowance for Decryption, and Decision point to transmit the AVTPDU just missed with zero (0) microseconds in the calculation of TRANSFER_DELAY. The derivation of the TRANSFER_DELAY value is given in Annex F of IEC 61883-8:2008.

The transmitter quantizes the timing of the "synchronization clock," for instance, the rising edge of the video clock, by referring to its own gPTP-based time. It transmits the sum of this time and TRANSFER_DELAY in the **avtp_timestamp** field of the AVTP header. If the timing information is not required for a CIP, the AVTP timestamp valid (**tv**) field shall be set to zero (0).

For IEC 61883-8:2008, replace 4.7.1.2 with:

The transmitter shall construct one or more stream data AVTPDU at the rate specified by the corresponding FQTSS stream class measurement interval. Each AVTPDU shall comply with the format as specified in Figure 6.4. When more than one source packet is transmitted that results in multiple AVTPDUs (each with its own CIP header), the transmitter shall divide those source packets as evenly as possible across the multiple AVTPDUs.

For IEC 61883-8:2008, replace 4.7.1.3 Blocking transmission method with:

The blocking method may be used by a transmitter when it is dealing with fixed counts of source packets per its own transmission interval where it then transmits them to the AVB network using the queuing mechanisms as specified in FQTSS for per stream shaping.

In the case where there is no data to transmit for a given FQTSS transmission cycle, the transmitter may either not transmit any AVTPDUs at all or it may transmit a stream data AVTPDU containing just a CIP header or a special nonempty AVTPDU that has the ND (NO DATA) flag set to 1_2 in its FDF field and has the same size of dummy data as a nonempty AVTPDU.

For blocking, the duration of the successive Video Data source packets in a CIP shall be added to the TRANSFER_DELAY.

If a CIP contains N Video Data source packets, then:

ACTUAL_TRANSFER_DELAY >= TRANSFER_DELAY + (N * VDSP_DURATION)

where

The TRANSFER_DELAY value is Max Transit Time defined in 5.5.2.

VDSP_DURATION is the duration of a Video Data source packet; it is dependent on video mode and color space. The VDSP_DURATION for each video mode is given in Annex B of IEC 61883-8:2008. The total delay for MAX_VDSP video source packets is also given in Annex B of IEC 61883-8:2008.

It is recommended that the receiver have sufficient extra buffer to compensate for any delay in receiving data because of the blocking transmission's characteristics. The actual value of extra delay required, and hence the additional buffer size required, depends on the video modes and color spaces supported by the receiving node.

7. MMA Payload Format over AVTP

7.1 Overview

AVTPDUs with a subtype value of MMA_SUBTYPE (Table 5.2) contain the MMA Payload Format. MMA stands for the Musical Instrument Digital Interface (MIDI) Manufacturers Association, which is the publisher and authoritative source of MIDI specifications. The MMA publishes the "MMA Payload Format Specification for AVTP" [B9], which defines the transport of MMA defined data over AVTP.

7.2 AVTP stream data AVTPDU format for MMA_SUBTYPE

This encapsulation uses a **cd** field of zero (0) and a **subtype** field of MMA_SUBTYPE (Figure 7.1). This encapsulation does not define subtype-specific control AVTPDUs.

This encapsulation also uses the common header field **protocol_specific_header**. For the MMA_SUBTYPE, the **protocol_specific_header** common header field (16 bits) is called the **MMA_payload_format_version** field and indicates the MMA Payload Format Version used by the **stream_data_payload** field. The external document "MMA Payload Format Specification for AVTP" [B9] defines all valid values for the **MMA_payload_format_version** field and provides a detailed specification for the format and usage of the **stream_data_payload** field corresponding to every valid value of **MMA_payload_format_version**.

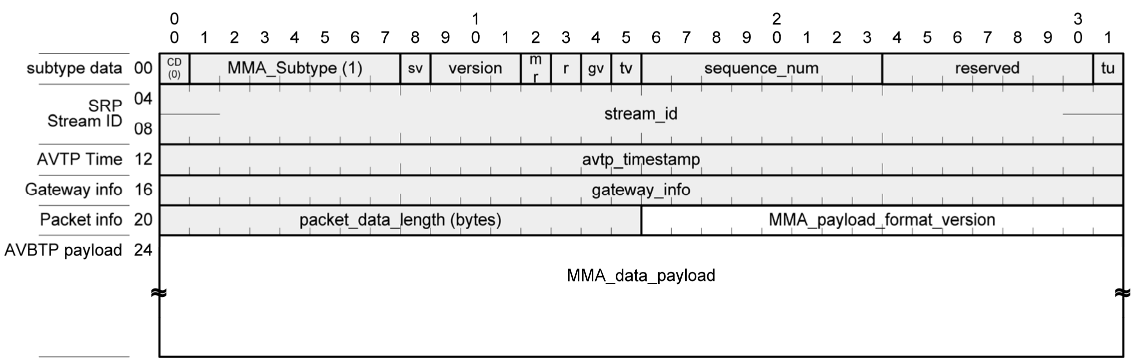


Figure 7.1—AVTP stream data AVTPDU format for MMA_SUBTYPE

Only the MMA may define the values of the **MMA_payload_format_version** and the corresponding payload format and usage for each such value. When the subtype common header field contains MMA_SUBTYPE, any other usage of the **protocol_specific_header** field or of the **stream_data_payload** field is invalid.

The MMA may define any number of MMA Payload Format Versions in the future by revising the "MMA Payload Format Specification for AVTP" [B9].

Annex A

(informative)

Bibliography

[B1] 1394 Trade Association 2004006, AV/C Digital Interface Command Set General Specification Version 4.2.[10]

[B2] Digital Transmission Content Protection (DTCP) Specification Volume 1 (Informational Version).[11]

[B3] DTCP Volume 1 Supplement D DTCP Use of IEEE 1394 Similar Transports (Informational Version), June 15, 2007.[12]

[B4] IEC 61784-2:2007, Industrial Communication Networks—Profiles—Part 2: Additional Fieldbus Profiles for Real-Time Networks Based on ISO/IEC 8802-3.[13]

[B5] IEEE P802.1BA™/D1.1 (Apr. 2010), Draft Standard Local and Metropolitan Area Networks: Audio Video Bridging (AVB) Systems.[14,15,16]

[B6] IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.

[B7] IEEE Std 1394.1™-2004, IEEE Standard for High Performance Serial Bus Bridges.

[B8] IIDC 1394-based Digital Camera Specification Ver.1.31, Feb. 2004.[17]

[B9] MMA Payload Format Specification for AVTP.[18]

[10] Available at http://www.1394ta.org/developers/Specifications.html.
[11] Available at http://www.dtcp.com/documents/dtcp/Info_20100319_DTCP_V1_1p6.pdf.
[12] Available at http://www.dtcp.com/documents/dtcp/Info20070615DtcpV1Sd1p1.pdf.
[13] IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iec.ch/). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.
[14] This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.
[15] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[16] The IEEE standards or products referred to in this clause are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.
[17] Available at http://damien.douxchamps.net/ieee1394/libdc1394/iidc/IIDC_1.31.pdf.
[18] The MIDI Manufacturers Association Inc., La Habra, CA, USA http://www.midi.org/avbtp.

Annex B

(normative)

MAC Address Acquisition Protocol

B.1 Overview

Multicast MAC addresses or locally administered unicast addresses are required by AVTP for the transmission of media streams. Because AVTP runs directly on a layer 2 transport, there is no existing protocol to allocate multicast MAC addresses dynamically. MAAP is designed to provide a way to allocate dynamically the multicast MAC addresses needed by AVTP. MAAP is not designed to allocate locally administered unicast addresses.

A block of Multicast MAC addresses has been reserved for the use of AVTP. These addresses are listed in Table B.4.

The MAAP specifies a mechanism to allocate Multicast MAC addresses dynamically in a specified address range.

Any application that uses addresses from the MAAP Dynamic Allocation Pool shall implement the MAAP.

MAAP can be used to request a single address or a range of consecutive addresses.

The MAAP uses a probe, announce, and defend mechanism in a peer-to-peer configuration to acquire addresses to be used by AVTP. An overview of the MAAP is described in this clause, and the detailed MAAP is described in B.3.

To acquire an address range for use, an implementation of MAAP executes the following steps:

a) Select an address range from the MAAP Dynamic Allocation Pool.
b) Send a series of MAAP_PROBE PDUs to determine whether the address range is already in use.
c) Listen for MAAP_DEFEND PDUs indicating the address range is in use.
d) Repeat the above steps until an unused address range has been found.

Once an address range has been acquired, MAAP announces and defends the address range as follows:

e) Send MAAP_ANNOUNCE PDUs periodically to inform the network of address ranges that are currently in use. This allows networks that are joined after the MAAP has been run to identify conflicts.
f) Listen for MAAP_PROBE PDUs and send MAAP_DEFEND PDUs in response to any address ranges that conflict with previously acquired address range(s).
g) Listen for MAAP_ANNOUNCE PDUs that conflict with previously acquired address range(s). If a conflicting MAAP_ANNOUNCE PDU is received, then MAAP discontinues use of the conflicting address range(s) and acquires new address range(s).

B.2 Protocol message format

MAAP uses a single PDU format for all messages. The MAAP uses the AVTPDU common control header format followed by the MAAP data.

A diagram of the MAAP PDU format is shown in Figure B.1.

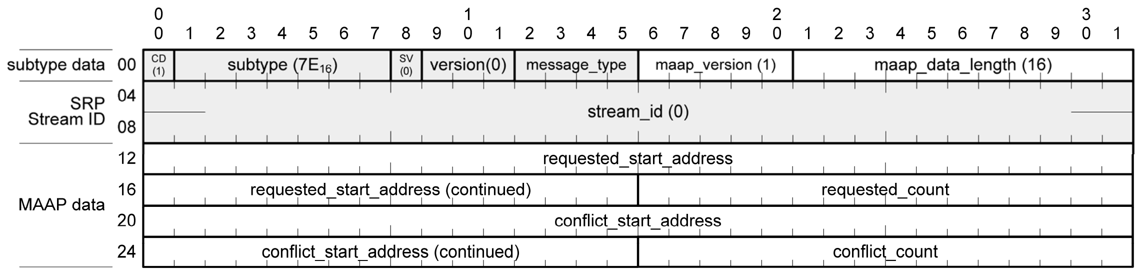


Figure B.1—IEEE 802.3 MAAP PDU format

The MAAP control PDU consists of the fields contained in the AVTPDU common control header and the following fields:

- **message_type**: 4 bits
- **maap_version**: 5 bits
- **maap_data_length**: 11 bits
- **stream_id**: 8 octets
- **requested_start_address**: 6 octets
- **requested_count**: 2 octets
- **conflict_start_address**: 6 octets
- **conflict_count**: 2 octets

All MAAP_PROBE and MAAP_ANNOUNCE frames are sent with a multicast destination MAC address set to the reserved MAAP multicast address defined in Table B.5.

All MAAP_DEFEND frames are sent with a destination MAC address set to the source MAC address received in the MAAP_PROBE frame that caused the sending of the MAAP_DEFEND frame.

The Source MAC Address shall be set to the MAC address of the sender.

The Ethertype field shall be the AVTP ethertype defined in Table 5.1.

B.2.1 CD

The **CD** field shall be set to 1.

B.2.2 Subtype

The **subtype** field shall be set to $7E_{16}$.

B.2.3 SV

The **SV** (StreamID valid) field shall be set to zero (0).

B.2.4 Version

The **version** field identifies the version of AVTP; see 5.2.4.

B.2.5 Message type

The **message_type** field contains one of the defined MAAP message types as defined in Table B.1. If a MAAP PDU is received with a reserved **message_type**, the PDU shall be ignored.

Table B.1—MAC Address Acquisition Protocol message types

Value (decimal)	FUNCTION	Meaning
0	—	Reserved
1	MAAP_PROBE	Probe MAC address(es) PDU
2	MAAP_DEFEND	Defend address(es) response PDU
3	MAAP_ANNOUNCE	Announce MAC address(es) acquired PDU
4 to 5	—	Reserved

B.2.6 maap_version

The **maap_version** field identifies the version of MAAP being used. The current version of MAAP is one (1).

To ensure compatibility with current protocol versions, while allowing the protocol to be extended in the future, the requirements listed in B.2.6.1 through B.2.6.3 shall be met by the implementation.

B.2.6.1 MAAP PDUs that carry a version higher than the receiver version

MAAP PDUs that carry a protocol version higher than the protocol version implemented by the receiver shall be interpreted according to the protocol definition of the receiver’s implemented version.

All MAAP PDUs received that contain a higher version number and a message type that is defined in the implemented version of MAAP shall be interpreted using the implemented version of MAAP, ignoring all unknown fields. This requires that future versions of MAAP maintain compatibility with the message types implemented in all previous versions of MAAP.

All MAAP PDUs received that contain a higher version number and a message type that is not defined in the implemented version of MAAP shall be ignored.

B.2.6.2 MAAP PDUs that carry a version equal to the receiver version

MAAP PDUs that carry a protocol version equal to the protocol version implemented by the receiver shall be interpreted according to the protocol definition of the receiver’s implemented version.

B.2.6.3 MAAP PDUs that carry a version less than the receiver version

MAAP PDUs that carry a protocol version less than the protocol version implemented by the receiver shall be interpreted according to the protocol definition corresponding to the protocol version received in the MAAP PDU. This requires that future versions of MAAP maintain the ability to interpret MAAP PDUs from all previous versions of MAAP.

B.2.7 maap_data_length

The **maap_data_length** is the length in octets of the MAAP data fields and shall be set to 16.

B.2.8 stream_id

The **stream_id** field is not used in MAAP and shall be set to zero (0).

B.2.9 requested_start_address

The **requested_start_address** in a MAAP_PROBE or MAAP_ANNOUNCE message is an address from the MAAP assigned address range that is the first address of a consecutive range of addresses being requested.

In a MAAP_DEFEND message, this field is set to the **requested_start_address** value received in the MAAP_PROBE or MAAP_ANNOUNCE PDU that initiated the transaction.

B.2.10 requested_count

The **requested_count** field in a MAAP_PROBE or MAAP_ANNOUNCE message is the number of addresses being requested. If only a single address is being requested, this field is set to one (1).

In a MAAP_DEFEND message, this field is set to the requested count value received in the MAAP_PROBE or MAAP_ANNOUNCE PDU that initiated the transaction.

B.2.11 conflict_start_address

The **conflict_start_address** field in a MAAP_DEFEND message is set to the first address that conflicts with a requested address range from a MAAP_PROBE PDU.

In all other MAAP messages, this field is set to zero (0).

B.2.12 conflict_count

The **conflict_count** field in a MAAP_DEFEND message is set to the number of addresses in the allocated address range, beginning with **conflict_start_address**, that conflicts with a requested address range from a MAAP PROBE PDU.

In all other MAAP messages, this field is set to zero (0).

B.3 Protocol specification

B.3.1 Notational conventions and abbreviations

The following conventions are used in the abbreviations used in this subclause:

rXXX	receive PDU XXX
sXXX	send PDU XXX
XXX!	state machine event

The MAAP is defined by the MAAP State machine (Table B.2). The following abbreviations are used in the state machine descriptions. For their meaning, see B.3.3, B.3.5, and B.3.6.

Protocol events:

Begin!	Initialize state machine (B.3.5.1)

Release!	Release an address range (B.3.5.2)
Restart!	Restart the state machine (B.3.5.3)
ReserveAddress!	Reserve an address (B.3.5.4)
rProbe!	Receive a MAAP PROBE PDU (B.3.5.5)
rDefend!	Receive a MAAP DEFEND PDU (B.3.5.6)
rAnnounce!	Receive a MAAP ANNOUNCE PDU (B.3.5.7)
probeCount!	maap_probe_count has decremented to zero (0) (B.3.5.8)
announceTimer!	announce_timer has expired (B.3.4.1)
probeTimer!	probe_timer has expired (B.3.4.2)

Protocol counters:

maap_probe_count	Counter that decrements from MAAP_PROBE_RETRANSMITS to zero (0)

Protocol actions:

generate_address	Generate a random address range (B.3.6.1)
init_maap_probe_count	Initialize maap_probe_count (B.3.6.2)
dec_maap_probe_count	Decrement maap_probe_count (B.3.6.3)
compare_MAC	Compare the received and local MAC address (B.3.6.4)
probe_timer	Probe period timer (B.3.4.2)
announce_timer	Announce period timer (B.3.4.1)
sProbe	Send a MAAP_PROBE PDU (B.3.6.5)
sDefend	Send a MAAP_DEFEND PDU (B.3.6.6)
sAnnounce	Send a MAAP_ANNOUNCE PDU (B.3.6.7)
-x-	Inapplicable event/state combination. No action or state transition occurs in this case.

Timers are used in the state machine descriptions to cause actions to be taken after defined time periods have elapsed. The following terminology is used in the state machine descriptions to define timer states and actions that can be performed upon them:

a) A timer is said to be running if the most recent action performed upon it was a start.
b) A running timer is said to have expired when the timer period associated with the timer has elapsed since the most recent start action took place.

c) A timer is said to be stopped if it has expired or if the most recent action performed upon it was a stop action.
d) A start action sets a stopped timer to the running state and associates a time period with the timer. This time period supersedes any periods that might have been associated with the timer by previous state events.
e) A stop action sets a timer to the stopped state.

Protocol states:

INITIAL	Initialize the state machine
PROBE	Probe the availability of an address range
DEFEND	Defend an acquired address range

B.3.2 State machine

A participant in MAAP maintains a single instance of this state machine for each address range currently allocated, or that is in the process of being allocated. The state machine defines events, actions, and state changes to be executed in response to a received packet or event. Functions, events, and state changes in each state table entry shall be executed sequentially. State changes are indicated by the new state name listed in the state table. If no state name is listed in an entry, then no state change occurs and the machine remains in its current state. State table entries marked with an “-x-“ indicate that no functions, events, or state changes occur and that the state machine remains in its current state. An entry that contains a state change/event function indicates that the state change is to occur first and then is immediately followed by the execution of the event in the destination state.

The detailed operation of this state machine is described in Table B.2.

Table B.2—MAAP State machine

		STATE		
		INITIAL	**PROBE**	**DEFEND**
EVENT	**Begin!**	generate_address[a] ReserveAddress!	-x-	-x-
	Release![c]	-x-	Stop probe_timer INITIAL	Stop announce_timer INITIAL
	Restart!	generate_address ReserveAddress!	-x-	-x-
	ReserveAddress!	init_maap_probe_count Start probe_timer sProbe PROBE	-x-	-x-
	rProbe![b]	-x-	compare_MAC[d] Stop probe_timer INITIAL/Restart!	sDefend
	rDefend![b]	-x-	Stop probe_timer INITIAL/Restart!	compare_MAC[d] Stop announce_timer INITIAL/Restart!
	rAnnounce![b]	-x-	Stop probe_timer INITIAL/Restart!	compare_MAC[d] Stop announce_timer INITIAL/Restart!
	probeCount!	-x-	Stop probe_timer Start announce_timer sAnnounce DEFEND	-x-
	announcetimer!	-x-	-x-	Start announce_timer sAnnounce
	probetimer!	-x-	Start probe_timer sProbe dec_maap_probe_count	-x-

[a] A Begin! event can be initiated with an assigned address range or MAAP can select an address range with the generate_address function. If an address range is supplied with the Begin! event, generate_address will not be called, and the supplied address range will be used. If the application has previously obtained an address range and has access to persistent storage, the application may record the previous address range and attempt to reuse the saved address range.

[b] Only received MAAP_PROBE, MAAP_DEFEND, and MAAP_ANNOUNCE PDUs that conflict with the address range associated with this state machine generate rProbe!, rDefend!, and rAnnounce! events. All MAAP_PROBE, MAAP_DEFEND, and MAAP_ANNOUNCE PDUs that do not conflict with the address range associated with this state machine are ignored.

[c] After a Release! event is received and the state machine has returned to INITIAL state, the address range associated with this state machine is considered to be free and the state machine can be destroyed.

[d] If the compare_MAC function returns TRUE, then no further processing or protocol action is taken and the protocol state does not change.

B.3.3 MAAP Constants

Constant values used by the MAAP are listed in Table B.3

Table B.3—MAAP Probe constant values

Constant	Value
MAAP_PROBE_RETRANSMITS	3
MAAP_PROBE_INTERVAL_BASE	500 ms
MAAP_PROBE_INTERVAL_VARIATION	100 ms
MAAP_ANNOUNCE_INTERVAL_BASE	30 s
MAAP_ANNOUNCE_INTERVAL_VARIATION	2 s

B.3.4 Protocol timers

B.3.4.1 announce_timer

The announce period timer, announce_timer, controls the period of time that separates the transmission of MAAP_ANNOUNCE PDUs. The announce period timer is set to a random value *T*, in the range MAAP_ANNOUNCE_INTERVAL_BASE < *T* < MAAP_ANNOUNCE_INTERVAL_BASE + MAAP_ANNOUNCE_INTERVAL_VARIATION.

NOTE—A random variation is added to the Announce and Probe period times to avoid synchronized bursts of frames flooding the network.

B.3.4.2 probe_timer

The probe period timer, probe_timer, controls the period of time that separates the transmission of MAAP_PROBE PDUs. The probe period timer is set to a random value *T*, in the range MAAP_PROBE_INTERVAL_BASE < *T* < MAAP_PROBE_INTERVAL_BASE + MAAP_PROBE_INTERVAL_VARIATION.

B.3.5 Protocol events

B.3.5.1 Begin!

The state machine is initialized or reinitialized.

B.3.5.2 Release!

A Release! event signals that the address range associated with this instance of the state machine is no longer in use and is no longer to be defended.

B.3.5.3 Restart!

A Restart! event signals that an address range conflict has been detected and that the state machine will be restarted with a new address range.

B.3.5.4 ReserveAddress!

A ReserveAddress! event signals that an address range is available to start the process of sending MAAP_PROBE PDUs to acquire the address range for use.

B.3.5.5 rProbe!

An rProbe! event signals that a MAAP_PROBE PDU has been received that contains an address range that conflicts with the address range assigned to this state machine.

B.3.5.6 rDefend!

An rDefend! event signals that a MAAP_DEFEND PDU has been received that contains an address range that conflicts with the address range assigned to this state machine.

B.3.5.7 rAnnounce!

An rAnnounce! event signals that a MAAP_ANNOUNCE PDU has been received that contains an address range that conflicts with the address range assigned to this state machine.

B.3.5.8 probeCount!

A probeCount! event signals that the dec_maap_probe_count has decremented the maap_probe_count to a value less than or equal to zero (0).

B.3.6 Protocol actions

B.3.6.1 generate_address

The generate_address function randomly selects an address range based on the number of addresses requested by the application. The address range is selected from the MAAP Dynamic Allocation Pool (Table B.4). The generate_address function selects the address range using a pseudo-random number generator with a uniform distribution across the MAAP Dynamic Allocation Pool range.

The pseudo-random number generation algorithm should be chosen so that different hosts do not generate the same sequence of numbers for subsequent calls to generate_address. The pseudo-random number generation algorithm should have a maximum length sequence of no less than $2^{32} - 1$. The pseudo-random number generator shall be seeded using the least significant (most rapidly varying) octets of the sum of the requestor's IEEE 802 MAC address and the local real time clock. The use of a ones-complement (end-around carry) sum is suggested but not required.

NOTE—Information about previously allocated address ranges gleaned from received MAAP_ANNOUNCE messages can be used to improve the performance of the generate_address function, to avoid selection of addresses that are already in use. The specific algorithms to do this are out of scope of this standard.

B.3.6.2 init_maap_probe_count

The init_maap_probe_count function sets the value of maap_probe_count to MAAP_PROBE_RETRANSMITS.

B.3.6.3 dec_maap_probe_count

The dec_maap_probe_count function decrements the value of maap_probe_count by one (1) and sends a probeCount! event if the resulting value of maap_probe_count is less than or equal to zero (0).

B.3.6.4 compare_MAC

The compare_MAC function compares the MAC address from the received MAAP PDU and the receiving station's MAC address. The MAC address comparison shall be performed in octet-wise reverse order; i.e., for the purpose of the unsigned comparison, the least significant byte is treated as if it were most significant, the most significant byte is treated as if it were least significant, and the interior bytes are compared in reverse order of their normal significance. compare_MAC shall return TRUE if the MAC address of the receiving station is numerically lower than the MAC address from the received MAAP PDU.

When the compare_MAC function returns TRUE, then no further processing or protocol action is taken and the protocol state does not change.

NOTE—An octet-wise reverse order comparison is used in the compare_MAC function to make the company assigned 24 bit OUI be the least significant component of the comparison.

B.3.6.5 sProbe

The sProbe function sends a MAAP_PROBE PDU where:

- **message_type** field is set to MAAP_PROBE.
- **requested_start_address** field is set to the start of the address range being requested.
- **requested_count** field is set to the number of contiguous MAC addresses being requested.
- **conflict_start_address** field is set to 00:00:00:00:00:00.
- **conflict_count** field is set to zero (0).

B.3.6.6 sDefend

The sDefend function sends a MAAP_DEFEND PDU where:

- **message_type** field is set to MAAP_DEFEND.
- **requested_start_address** field is set to the value of the **requested_start_address** field from the MAAP_PROBE that caused the sDefend function to be called.
- **requested_count** field is set to the value of the **requested_count** field from the MAAP_PROBE that caused the sDefend function to be called.
- **conflict_start_address** field is set to the first allocated address that conflicts with the requested address range.
- **conflict_count** field is set to the count of allocated addresses beginning with **conflict_start_address** that conflict with the requested address range.

NOTE—A MAAP_PROBE PDU can contain an address range that conflicts with multiple contiguous or noncontiguous address ranges that are being defended. Because an instance of the state machine is maintained for each acquired address range, a separate MAAP_DEFEND PDU will be sent for each address range that conflicts with a received MAAP_PROBE.

B.3.6.7 sAnnounce

The sAnnounce function sends a MAAP_ANNOUNCE PDU where:

- **message_type** field is set to MAAP_ANNOUNCE.
- **requested_start_address** field is set to the start of the address range acquired.
- **requested_count** field is set to the number of addresses acquired.
- **conflict_start_address** field is set to 00:00:00:00:00:00.
- **conflict_count** field is set to zero (0).

B.4 Reserved MAAP MAC addresses

There are 256 MAAP addresses that are reserved for the purposes of this standard. The MAAP reserved addresses are defined in Table B.5. MAAP reserved multicast addresses are divided into three categories, including a dynamic allocation pool, a locally administered static allocation pool, and a pool reserved for use by this standard. The multicast address pools are defined in Table B.4. Usage of the MAAP reserved address pool is defined in Table B.5.

Table B.4—MAAP Multicast Addresses

Address Range	Function	Meaning
91:E0:F0:00:00:00–91:E0:F0:00:FD:FF	MAAP Dynamic Allocation Pool	These addresses are available for dynamic allocation by the MAAP.
91:E0:F0:00:FE:00 – 91:E0:F0:00:FE:FF	MAAP locally administered Pool	These addresses are reserved to be statically allocated.
91:E0:F0:00:FF:00 – 91:E0:F0:00:FF:FF	MAAP Reserved Pool	Use of these addresses is defined in Table B.5.

Table B.5—Reserved Multicast MAAP MAC Addresses

Address	FUNCTION	Meaning
91:E0:F0:00:FF:00	MAAP	Multicast address used for MAAP Frames
91:E0:F0:00:FF:01-91:E0:F0:00:FF:FF	—	Reserved

Annex C

(informative)

IEEE 802.3 media-specific encapsulation

C.1 Introduction

This annex for end stations describes the IEEE 802.3 encapsulation of stream data and control AVTPDUs. The purpose of this annex is to aid in the understanding of the relationship of AVTP to the overall Ethernet protocol stack. Whereas details of the encapsulation for IEEE Std 802.3-2008 are included here, other encapsulations are not excluded (e.g., IEEE Std 802.11-2007, MOCA).

The IEEE 802.1 AVB standards require that stream data frames (as opposed to command or non-AVB data frames) be marked with a 3 bit priority code point (PCP) so bridges can detect these frames as SR class A or SR class B and treat these frames differently from non-AVB stream frames. Without this marking of the stream frames, the guarantees and performance of AVB networks cannot be met. This marking of stream frames is not the only requirement for a successful AVB network (see the full set of IEEE 802.1 AVB standards), but it is a necessary one, so its proper usage from an end station point of view is discussed here.

This annex covers the following fields:

a) Destination MAC address: 48 bits
b) Source MAC address: 48 bits
c) IEEE 802.1Q protocol header: 4 octets consisting of:
    — Tagged Protocol Identifier: 16 bits
    — Canonical Format Indicator (CFI): 1 bit
    — PCP: 3 bits
    — Virtual Local Area Network (VLAN) Identifier: 12 bits

For IEEE 802.1Q operation (VLAN tagged frames), the Ethertype field immediately following the source MAC address is known as the TPID field and is set to 8100_{16}. For this case, the AVTP Ethertype is at an offset 4 octets past the start of this field.

Figure C.1 shows an AVTP frame encapsulated within an IEEE 802.3 frame with an IEEE 802.1Q header (also known as an IEEE 802.1Q VLAN Tag field). For IEEE 802.3 frames, all AVTP stream data frames use this format and AVTP control frames may use this format.

NOTE—The AVTP frame's common header and data, as discussed in 5.2 (shown in gray in the figure), immediately follows the AVTP frame's Ethertype field.

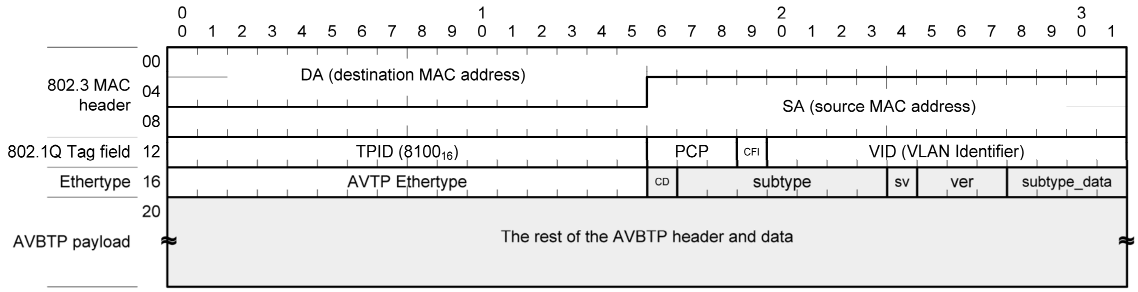


Figure C.1—AVTP control or stream data frame in an IEEE 802.3 frame with an IEEE 802.1Q Tag field

Figure C.2 shows an AVTP frame encapsulated within an IEEE 802.3 frame without an IEEE 802.1Q header. This format is optional for AVTP control frames. AVTP stream data frames do not use this format as all stream data frames will be priority encoded for SR class A or SR class B traffic, thus making the VLAN tag header mandatory.

NOTE—The AVTP frame's common header and data, as discussed in 5.2 (shown in gray in the figure), immediately follows the AVTP frame's Ethertype field.

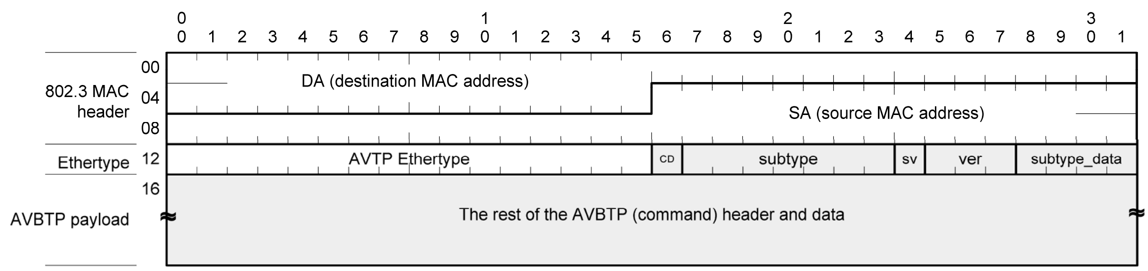


Figure C.2—AVTP control frame in an IEEE 802.3 frame without an IEEE 802.1Q Tag field

C.2 IEEE 802.3 Fields

C.2.1 IEEE 802.3 destination MAC address field

For AVTP stream data frames, the MAC Destination Address of each stream will be unique for the layer 2 network and may either be a locally administered unicast address or a multicast address (as defined in SRP). Multicast addresses may be assigned by use of MAAP defined in Annex B. The use of MAAP can guarantee that the Multicast addresses will be unique per AVTP stream.

For AVTP control frames, the MAC Destination Address may be unicast, multicast, or broadcast depending on the specification of the usage of each AVTP control frame.

C.2.2 IEEE 802.3 source MAC address field

For AVTP stream data and control frames, the MAC Source Addresses contains the sender's unicast MAC address.

C.2.3 IEEE 802.1Q header field

Some AVTP frames require an IEEE 802.1Q header as described by the following rules:

a) All AVTP Talkers send AVTP stream data frames with an IEEE 802.1Q header present and the PCP field set to indicate either SR class A or SR class B.

b) All AVTP compliant devices must be able to receive and process AVTP stream data with an IEEE 802.1Q header present.

c) All AVTP compliant devices must be able to receive and process AVTP control frames with or without an IEEE 802.1Q header present.

The IEEE 802.1Q header portion of an AVTP frame is filled out as discussed in C.2.3.1 through C.2.3.4 (when it appears in a frame).

C.2.3.1 IEEE 802.1Q tagged protocol identifier field

All frames with an IEEE 802.1Q header field set the TPID field to a value of 8100_{16} hexadecimal as required by IEEE Std 802.1Q-2005.

C.2.3.2 VID field

The VLAN identifier field is used to indicate the IEEE 802.1Q VLAN that an AVTP frame is to be processed on. It is not to be used for any other purpose (i.e., it is not to be used as a stream identifier in any way). VLANs are generally set up and managed in bridges. End stations (Talkers and Listeners in this standard) may or may not be VLAN aware. A VLAN aware station is one that can process information by management that tells the device what VID or VIDs it is a member of and what VID each stream is to use. Special care is needed when using VLAN aware stations, and all VLAN aware stations default to a mode where they act as if they are not VLAN aware until configured otherwise by management.

If an AVTP Talker is not VLAN aware, it transmits all stream data frames with a VID of SRclassVID (SRP, 35.2.2.9.4), which it learns via the MSRP Domain attribute. If a Talker is directly connected to a Listener, it may use a VID of SR_PVID (SRP, 35.2.1.4).

If an AVTP Talker is VLAN aware, it transmits stream data frames with the VID VLAN management has assigned to them.

To receive stream data frames, all AVTP Listeners must request membership in the Talker's VLAN. A Listener does not need to be VLAN aware to do this; it simply needs to issue an MVRP membership request for the VID contained in the Talker Advertise.

Conceptually, all Talker functions and all Listener functions are not VLAN aware. If a Talker device needs to be VLAN aware, it can be thought of as a non-VLAN aware Talker function that always transmits streams with a VID of SR_PVID followed by a logical two port bridge function that assigns a VID to streams from the particular Talker function. A device with two Talker functions that needs to transmit separate streams out with separate VIDs can be thought of as two non–VLAN-aware Talker functions connected with a three port bridge function that assigns separate VIDs for each Talker. Likewise, if a Listener device really needs to perform VLAN filtering, it can be seen as a non–VLAN-aware Listener that ignores all VID processing with a two port bridge function in front of it that performs VLAN filtering.

C.2.3.3 CFI field

As of the time of the writing of this standard, IEEE Std 802.3-2008 requires this bit to be zero (0).

C.2.3.4 Priority Code Point (PCP) field

For all stream data frames, AVTP Talkers set the PCP value to the SRclassPriority (SRP, 35.2.2.9.3), which it learns via the MSRP Domain attribute for either stream SR class A traffic or stream SR class B traffic.

For all nonstream data frames (i.e., for all other frame types), AVTP Talkers set the PCP value to something other than the FQTSS-specified default or management-specified value for either SR class A traffic or SR class B traffic.

All AVTP Listeners must be able to receive and process stream data frames with any PCP value (the SR class A and SR class B defaults are used unless modified by management).

C.2.3.5 Other considerations

AVB networks require that some IEEE 802.3 compliant device modes of operation cannot be used. These requirements are documented in IEEE P802.1BA™ [B5]. For informative purposes, some of the rules that must be followed in AVB Networks are listed as follows:

a) Link speed must be 100 Mbit/s or faster.
b) Link duplex must be full duplex.
c) MAC Pause must not be used on AVB links.
d) Priority Pause must not be used on AVB priorities.
e) The maximum size of the MAC Client data portion of a frame must never exceed 1500 octets.