STANDARDS AND INFORMATION DOCUMENTS

Call for comment on DRAFT AES standard for Audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110-30 media transport

This document was developed by a writing group of the Audio Engineering Society Standards Committee (AESSC) and has been prepared for comment according to AES policies and procedures. It has been brought to the attention of International Electrotechnical Commission Technical Committee 100. Existing international standards relating to the subject of this document were used and referenced throughout its development.

Address comments by E-mail to standards@aes.org, or by mail to the AESSC Secretariat, Audio Engineering Society, 697 Third Ave., Suite 405, New York NY 10017. **Only comments so addressed will be considered**. E-mail is preferred. **Comments that suggest changes must include proposed wording.** Comments shall be restricted to this document only. Send comments to other documents separately. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

This document will be approved by the AES after any adverse comment received within **six weeks** of the publication of this call on http://www.aes.org/standards/comments/, **2024-08-25**, has been resolved. Any person receiving this call first through the *JAES* distribution may inform the Secretariat immediately of an intention to comment within a month of this distribution.

Because this document is a draft and is subject to change, no portion of it shall be quoted in any publication without the written permission of the AES, and all published references to it must include a prominent warning that the draft will be changed and must not be used as a standard.

AES standard for audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110-30 media transport

Published by
Audio Engineering Society, Inc.

Abstract

AES70 is a suite of standards for control and monitoring of devices in professional media networks. This standard, *AES standard for audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110 30 media transport* defines an application of the Core AES70 specification for managing AES67 and SMPTE ST 2110-30 media transport connections, and related synchronization and clocking mechanisms. Other standards in the AES70 suite specify concepts and mechanisms, control and monitoring repertoire, control protocols, and other media transport management applications.

AES70's intended range of use spans networks of all sizes. This includes mission-critical applications, high-security applications, IP and non-IP networks, and local and wide-area applications. AES70 can control real or virtual devices located on premises or hosted by cloud services. AES70 consumes little computing power and uses network bandwidth lightly.

AES70 is based on the Open Control Architecture (OCA), originally developed by the OCA Alliance.

Audio Engineering Society Inc. 697 Third Avenue, Suite 405, New York, NY, US.
www.aes.org/standards - standards@aes.org

Foreword

This foreword is not a part of this document, *AES standard for audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110 30 media transport.*

The role of AES standards. An AES standard implies a consensus of those directly and materially affected by its scope and provisions and is intended as a guide to aid the manufacturer, the consumer, and the general public. Prior to the publication of an AES standard, all parties, including the general public, are given opportunities to comment or object to any provision. Nevertheless, the existence of an AES standard shall not preclude anyone, whether or not he or she has approved the document, from manufacturing, marketing, purchasing, or using products, processes, or procedures not in agreement with the standard.

Patent rights. Attention is drawn to the possibility that some of the elements of this AES standard or information document may be the subject of patent rights. AES shall not be held responsible for identifying any or all such rights. Approval by the AES does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the document.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Review and revision. This document is subject to periodic review and possible revision. Users are cautioned to obtain the latest edition.

AES70 structure

The AES70 standard is a suite of standards, classified into two divisions. The ***Core Standards*** division, contains standards essential to all implementations of AES70; the ***Adaptation Standards*** division contains application-specific standards. This standard, *AES standard for audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110 30 media transport*, is an Adaptation Standard.

AES70-21 technical concept

A media network contains at least four sets of services:

- *media transport:* synchronized transportation of media data;
- *control:* controlling devices;
- *directory*: allowing devices and controllers to identify and discover each other;
- *time:* enabling media transport synchronization and control action timing.

A key function of the control set is *connection management*, i.e. control of the making, breaking, and operation of media transport connections and sessions. AES70 contains a rich set of connection management features. This Standard specifies a way of using those features to manage AES67 and SMPTE ST 2110-30 media transport connections and sessions.

Directory and time services are not part of this Standard.

AES70-21 version history

This version, AES70-21-2024 is the first version of the Standard.

The AES task group members who contributed to this Standard were F. Bergholtz, J. Berryman, T. de Brouwer, S. Jones, A. Kuzub, M. Lave, G. Linis, S. Price, A. Rosen, S. Scott, G. Shay, M. Smaak, P. Stevens, P. Treleaven, S. van Tienen, M. Versteeg, and P. Waddell.

The document was based on an advanced draft contributed by the OCA Alliance.

J. Berryman led the writing group.

Morten Lave
Chair, working group SC-02-12
2024-08-24

Note on normative language

In AES standards documents, sentences containing the word "shall" are requirements for compliance with the document. Sentences containing the verb "should" are strong suggestions (recommendations). Sentences giving permission use the verb "may". Sentences expressing a possibility use the verb "can".

Contents

Tables

Figures

AES standard for audio applications of networks - Open Control Architecture - Part 21: Using AES70 to manage AES67 and SMPTE ST 2110-30 media transport

1. Introduction

This Standard specifies an application of the Core AES70 specification for managing AES67 and SMPTE ST 2110-30 media transport connections and related synchronization and clocking mechanisms. It is part of the AES70 standards family.

In the AES70 standards family, the capitalized term "Adaptation" means a formal specification of an AES70 application for a particular use.

2. Scope

This Standard applies to using AES70 to manage AES67 and SMPTE ST 2110-30 media stream connections between devices, where AES70 is used to control at least one of the devices, while other devices may be controlled using AES70, SIP, or out-of-the-scope means. The list of use cases is given in section 8.

3. Normative references

The following referenced documents are indispensable for the application of this Standard:

AES67. *AES67-2018: AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability,* Audio Engineering Society, New York, NY., US.

AES70-1. *AES70-1-2021: AES standard for audio applications of networks - Open Control Architecture - Part 1: Framework,* Audio Engineering Society, New York, NY., US.

AES70-2. *AES702-2-2021: AES standard for audio applications of networks - Open Control Architecture - Part 2: Class structure,* Audio Engineering Society, New York, NY., US.

AES70-2A. *AES70-2-2021, Annex A: AES standard for audio applications of networks - Open Control Architecture - Part 2: Class structure; Annex A,* Audio Engineering Society, New York, NY., US.

IEEE 1588-2008. *IEEE 1588-2008: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.* IEEE, New York, NY., US, July 2008.

ST 2022-7. *ST 2022-7:2019: SMPTE Standard: Seamless Protection Switching of RTP Datagrams.* Society of Motion Picture and Television Engineers, White Plains, NY., US.

ST 2110-30. *SMPTE Standard: Professional Media Over Managed IP Networks: PCM Digital Audio.* SMPTE ST 2110-30:2017, Society of Motion Picture and Television Engineers, White Plains, NY., US.

A document reference will be identified in the text by the document name, shown above in bold, in square brackets. A part of a document will be identified by a suffix of the form (<section heading>), attached to the document name. For example, [AES67] represents the AES67-2023 standard, and [AES67(Media clocks)] represents the Media clocks chapter of it.

The special notation [AES70] is used to denote the collection of documents that constitute the Core AES70 standard.

References to IETF RFC documents are given in the form [RFC n] where n is the RFC number.

4. Terms, definitions, and abbreviations

See [AES70-1] for most terms. In addition, this Standard defines or uses the following terms:

1. **Connection Negotiation**
   process by which controllers and/or devices agree on a compatible set of Stream Mode parameters for the purpose of establishing a media stream connection.

2. **Core AES70**
   members of the AES70 standards family that define the fundamentals of AES70: AES70-1, AES70-2, AES70-3, and possible future standards. In the AES70 suite, Core standards have document identifiers with numeric suffixes less than 20.

3. **Parameter Record**
   container for parameters defined outside AES70's scope, consisting of an instance of the OcaParameterRecord datatype and associated methods, constructed and used according to the **Parameter Record** design pattern defined in [AES70-2A(Support/Design Patterns)].

4. **Media Transport Session; (in context) Session**
   single media stream connection, or collection of media stream connections assembled together, serving a specific purpose. Session support is implementation-specific and may include management and persistence mechanisms, such as manual configuration interface, machine control protocols, built-in logic for session restoration after a device restart or restoration of failed connections, and so on.

5. **Media Transport Session management; (in context) Session Management**
   process of setting up, controlling, monitoring, and tearing down Media Transport Sessions.

6. **IP Network**
   OSI layer 3 network based on the Internet Protocol (IP) [RFC 791, RFC 2460].

7. **Packet Arbitrator**
   in a Redundancy Scheme (Definition 8), the element that selects media stream packets from the redundant stream connections.

8. **Redundancy Scheme**
   set of IP Networks and AES70 mechanisms assembled and configured to implement Seamless Reconstruction (Definition 9) of media streams.

9. **Seamless Reconstruction**
   in SMPTE ST 2022-7, creation of a reconstructed output stream from a set of redundant, potentially network-impaired, input streams. In most cases the reconstructed stream will be either fully intact or with significantly reduced packet loss ratio.

10. **Stream Mode**
    payload properties of an individual media stream connection: frame format, , sampling rate, sample encoding, channel count, and packet time.

11. Time To Live (TTL)
maximum number of router-to-router hops an IP packet can make before being discarded by the IP routing mechanism - see [RFC 791] for full details.

5. Prerequisite

The Core AES70 mechanisms upon which this Standard is based are defined in:

- [AES70-1(Networking model)]
- [AES70-1(Connection management)]
- The related class and datatype definitions in [AES70-2A].

Familiarity with these references is essential for understanding this Standard.

6. Document conventions

In this Standard:

1. All document conventions defined in [AES70-1(Document conventions)] apply.

2. "ONo" denotes an AES70 object number

3. The notation **AES67 / ST 2110-30** indicates equal applicability to both AES67 and ST 2110-30.

4. Full definitions of variables (class properties, method parameters, etc.) are given in the form:

   <datatype> <variablename>

   e.g.

   OcaString HostName .

5. Classes and datatypes defined or referenced in this Standard shall be named according to the following rules:

   - Every standard shall define a class name prefix, shown in red in what follows. The Core AES70 prefix is Oca; this Standard's prefix shall be Aes67.
   - The name of a class or datatype shall begin with the prefix of the standard defining it.
   - When a class is subclassed by another standard, the new standard's prefix shall be **prepended** to the original class name, for example, Aes67Oca.
   - Examples:

Original class defined by Core AES70:	OcaMediaTransportApplication
Subclass defined by this Standard:	Aes67OcaMediaTransportApplication
Original datatype defined by this Standard:	Aes67EndpointAdaptationData

7. Adaptation elements

[AES70-1] defines ***Adaptation*** **as a** " formal specification of a set of augmentations and/or constraints applied to AES70 for a particular use."

This Standard specifies the following Adaptation elements:

1 Values and value constraints

Samples of value constraints are in Annex A and Annex B, in which this Standard specifies an extensive set of media Stream Mode descriptions and related details.

2 Operating rules

Samples of Adaptation-specific operating rules are in Clause 11.4, which specifies configuring rules required for support of the network Redundancy Scheme standardized in [ST 2022-7].

3 Adaptation-specific data elements

In various places throughout the AES70 class structure, classes and datatypes have properties named AdaptationData of datatype OcaAdaptationData. The purpose of these properties is to store Adaptation-specific information. The formats of such information elements for AES67 connection management are set by data structures specified by this Standard.

4 Control class extensions

In AES70, a control class extension is created by defining a new control class that is a subclass of the Core AES70 class. This Standard defines the following Adaptation-specific subclasses:

- Aes67OcaMediaTransportApplication. See chapter 10.2.
- Aes67OcaMediaTransportSessionAgent. See clause 11.3.1.

6. Adaptation-specific Control Classes

As specified by [AES70-1], an Adaptation-specific Control Class is defined by defining a new control class that is a subclass of a Core AES70 class. This Standard defines two Adaptation-specific subclasses as follows:

Aes67OcaMediaTransportApplication, Class ID **1.2.20.A.2100**

Aes67OcaMediaTransportSessionAgent, Class ID **1.2.20.A.2101**

Where **A** is the Authority ID that identifies the Authority responsible for defining the nonstandard classes used in this Standard. **A** shall be as follows:

0xFFFF00. 0x000B5E

This Authority ID contains the 24-bit Company ID (CID) or Organization Unique Identifier (OUI) issued by the IEEE Registration Authority for proprietary classes defined by the AES, and follows the format defined in [AES70-1(Authority ID format)].

7. Adaptation identifier

AES70 requires each Adaptation to have a unique identifier. The identifier of this Adaptation shall be "OcaAes67".

8. Use cases

This Standard shall support the following use cases:

1 **Controller uses AES70 to control all devices directly.**

Figure 1 – Case 1: A controller sends AES70 commands to each device to set up the endpoints for the required connection(s).

2 **Controller uses AES70 to control one of the devices directly, and that device uses AES70 to control the other device(s).**

Figure 1 - Case 2: A controller sends AES70 commands to one of the devices, and that device assumes the role of a controller and sends AES70 commands to the other device to complete the connection(s).

3 **Controller uses AES70 to control one of the devices directly, and out-of-scope means are used to control the other device(s).**

Figure 1 - Case 3: A controller sends AES70 commands to the AES70-controlled device to set up the endpoint(s) for the required connection(s). The endpoint(s) on the other device is(are) set up by means outside the scope of this Standard. For example, a controller may use non-AES70 machine control methods to set up the other endpoint(s); or the other endpoint(s) may be set up manually.

4 **Controller uses AES70 to control one of the devices directly, and that device uses Session Initiation Protocol (SIP) to complete the connection(s).**

Figure 1 - Cases 4a and 4b: A controller sends AES70 commands to the AES70-controlled device, and the device sets up the connection(s) using the SIP feature of its AES67 implementation. SIP may operate in serverless (i.e. peer to peer) (Case 4a) or server-based (Case 4b) mode.

NOTE: SIP functionality is required by [AES67], but not by [ST 2110-30].

Figure 1 illustrates the listed use cases, including both SIP modes. For simplicity, all use cases are depicted with exactly two devices; real-life applications may involve multiple devices.

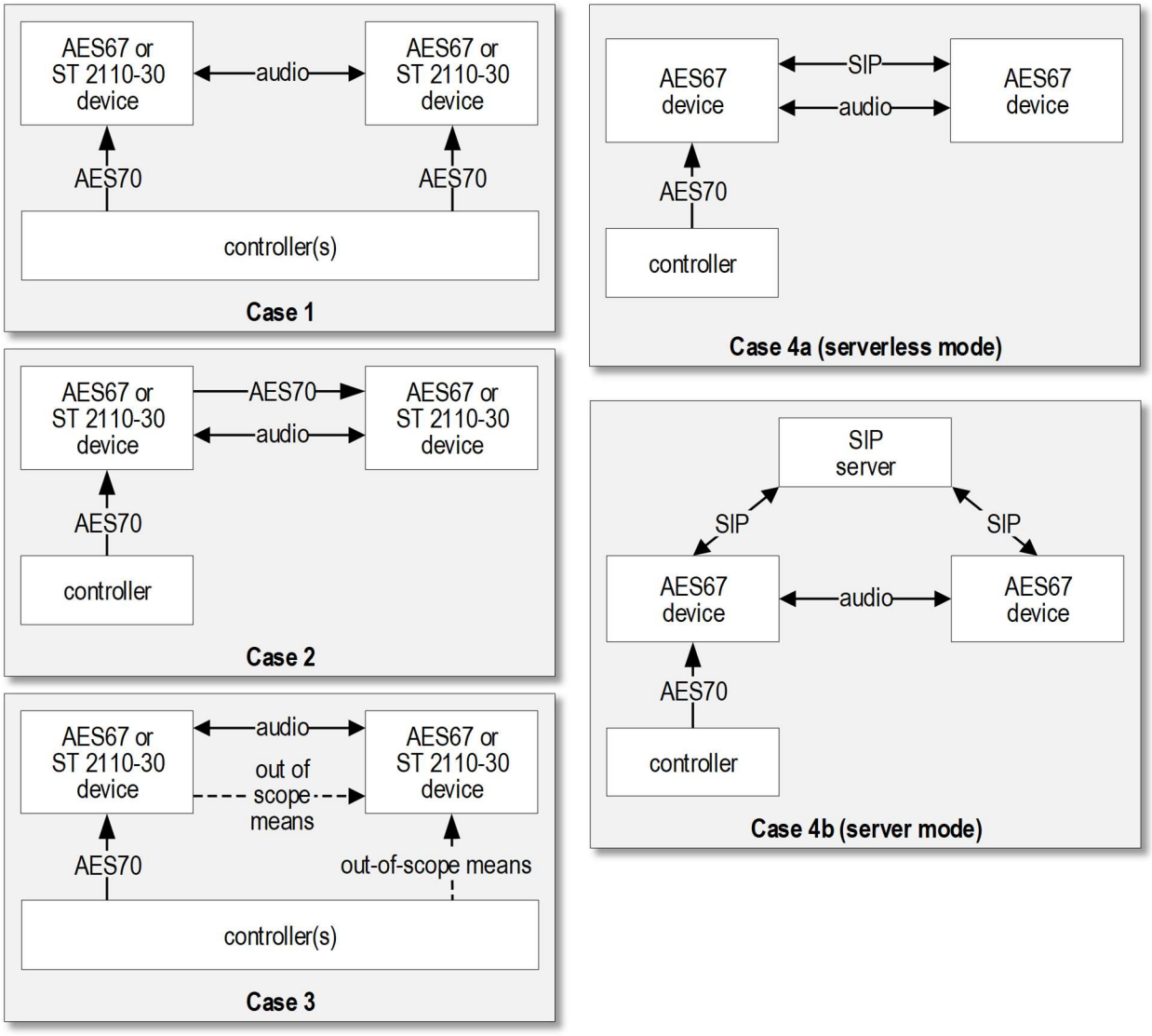


Figure 1. Use Cases

NOTE: The mechanisms specified by this Standard may be effective for supporting additional use cases, even though such uses are out of scope.

9. The AES67/ST 2110-30 media transport application control model

The AES67 media transport application control model is a specialization of the Core AES70 connection management model, which is known as ***CM4*** ("connection management revision 4"). - CM4 is specified normatively in [AES70-1(Media transport control model)] and in the relevant classes and datatypes in [AES70-2A].

CM4 is a specialization of the Core AES70 networking model named ***NAC*** ("network application control"). NAC is specified normatively in [AES70-1(Networking model)] and in the relevant classes and datatypes in [AES70-2A].

9.1. AES67/ST 2110-30 class subtree

Figure 2 illustrates the AES67/ST 2110-30 class subtree, including the two Adaptation-specific subclasses Aes67OcaMediaTransportApplication and Aes67OcaMediaTransportSessionAgent.

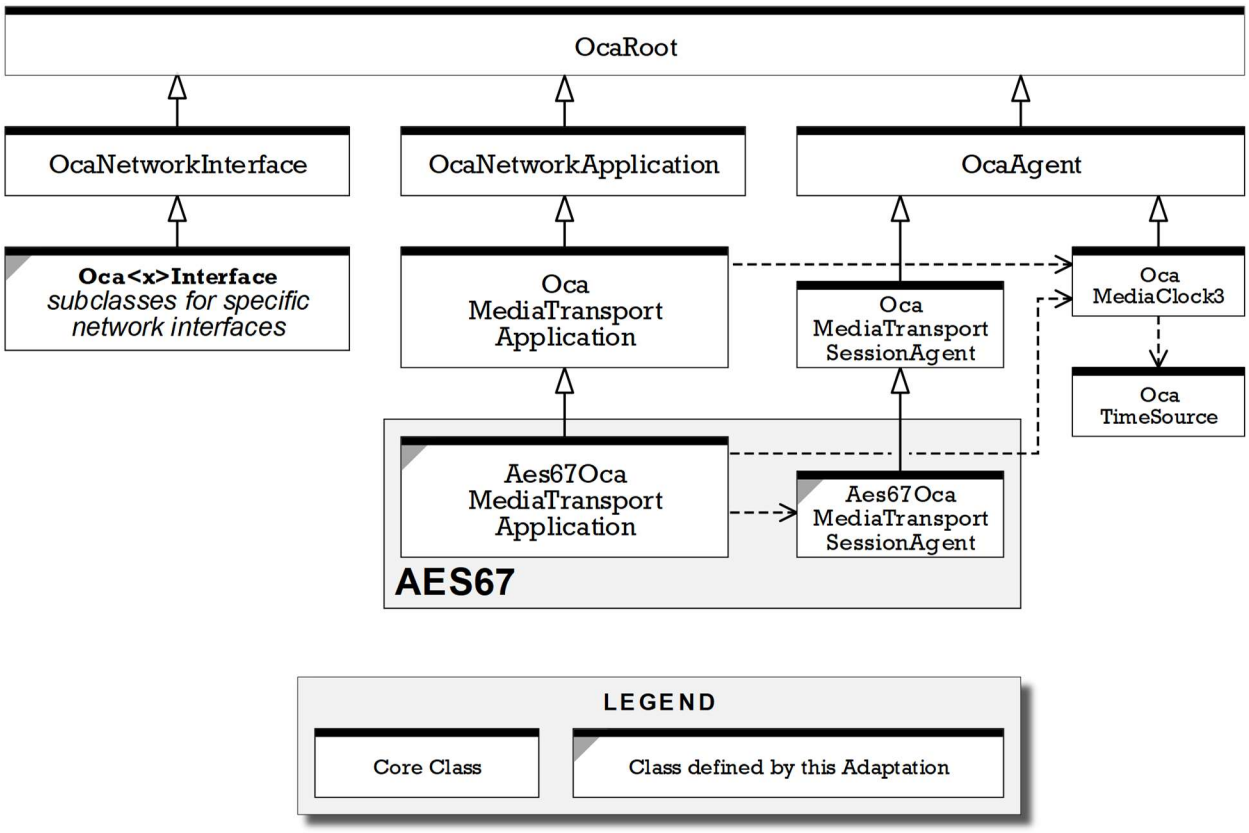


Figure 2. AES67/ST 2110-30 classes

9.2. AES67/ST 2110-30 NAC Stack

CM4 defines a number of significant datatypes. This Standard defines specialized versions of some of them. Normative specifications of these elements are in the following clauses and are illustrated in Figure 3.

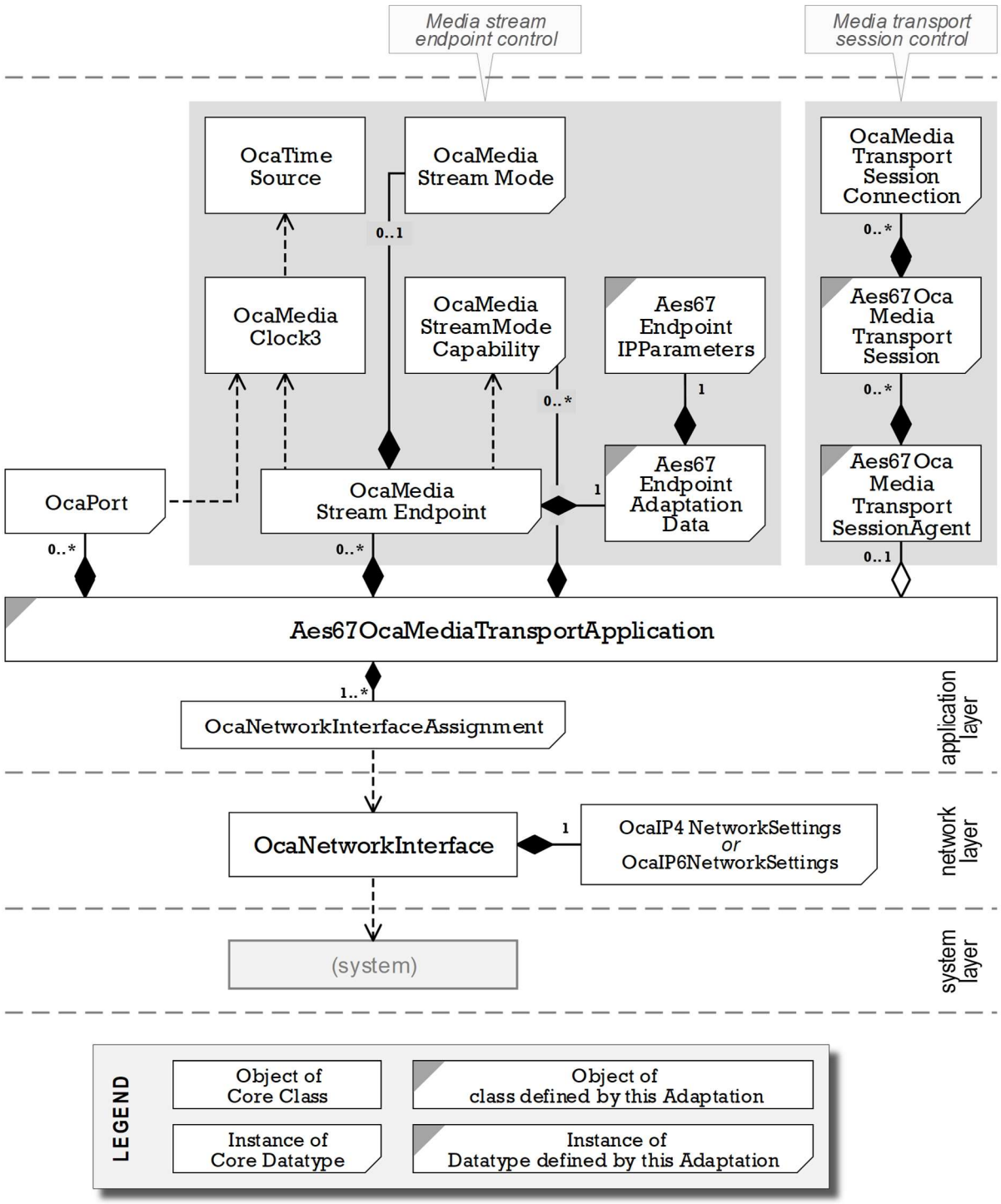


Figure 3. AES67/ST 2110-30 NAC Stack

10. Adaptation details

This normative chapter specifies classes, datatypes, parameter constraints, and operating rules of the AES67/ST 2110-30 NAC Stack.

10.1. Class OcaNetworkInterface

OcaNetworkInterface details for this Standard conform the standard AES70 IP Adaptation defined for IPv4 networks in [AES70-1(IP Adaptation)]. The IPv6 details in that Adaptation are not relevant, since AES67 currently does not support IPv6.

10.2. Class Aes67OcaMediaTransportApplication

Aes67OcaMediaTransportApplication is this Standard's subclass of the Core AES70 class OcaMediaTransportApplication. This subclass shall be the same as its parent, except that it shall define three additional methods SubmitSDP(), GetSubmittedSDP(), and GetActiveSDP(). These methods are defined normatively in clause 11.2.3.

10.2.1. Property NetworkInterfaceAssignments and datatype OcaNetworkInterfaceAssignment

Property Aes67OcaMediaTransportApplication.NetworkInterfaceAssignments shall contain a list of one or more OcaNetworkInterfaceAssignment instances. Each such instance shall designate an OcaNetworkInterface object that corresponds to a network interface the Aes67OcaMediaTransportApplication object uses.

For this Standard, property values of OcaNetworkInterfaceAssignment instances shall be as shown in Table 1.

Table 1. OcaNetworkInterfaceAssignment property values

Content	Property	Value
ID of this network interface assignment	*OcaID16* ID	1..n
Network interface ONo	*OcaONo* NetworkInterfaceONo	Object number of OcaNetworkInterface object
Type of advertisement protocol used	*OcaBlob* AdvertisementProtocolDescriptor	see clause 10.3.
Services advertised for this network application on the given network	*OcaList<OcaParameterRecord>* AdvertisedServices	

10.3. Service advertising

NAC supports the concept of ***service advertising***, by which network applications can advertise their services using various network services and mechanisms. See [AES70-1(Service Advertising)].

Advertising of media transport endpoints and sessions is mentioned as an option in AES67, but no details are specified. Therefore, no advertising of media transport services is specified by AES70-21.

10.4. Counters

Various properties in AES70 network classes and datatypes contain AES70 *countersets* - see [AES70-1(Counters and countersets)] and the normative definitions in [AES70-2A].

In all cases, the particular sets of counters defined for these countersets shall be implementation-dependent, and are out of scope of this Standard. Where an implementation does not need a counterset, the respective counterset property may be empty and the methods for accessing it may return the status value NotImplemented.

10.5. Datatype OcaMediaStreamEndpoint

Aes67OcaMediaTransportApplication.Endpoints shall collect descriptors of all the stream endpoints associated with the respective Aes67OcaMediaTransportApplication object. The OcaMediaStreamEndpoint datatype shall define the format of a stream endpoint descriptor.

For this Standard, values of notable OcaMediaStreamEndpoint properties shall be as shown in Table 2.

Table 2. Notable OcaMediaStreamEndpoint fields

Content	Field		Value
Security type	*OcaSecurityType*	SecurityType	Shall be the enum value None.
AES67-specific endpoint parameters	*OcaBlob*	AdaptationData	the structure Aes67EndpointAdaptationData - see clause 10.5.1

10.5.1. Property OcaMediaStreamEndpoint.AdaptationData

OcaMediaStreamEndpoint.AdaptationData (datatype OcaBlob) is a flexible property whose format and contents depend on the Adaptation.

For this Standard, the format and contents of the AdaptationData property shall be as defined by the datatype Aes67EndpointAdaptationData, defined normatively in Table 3, and its nested datatype Aes67EndpointIPParameters, defined normatively in Table 4.

Field values of these datatypes for specific use cases (input vs output, unicast vs multicast) are specified in Table 5 through Table 8.

Table 3. Datatype Aes67EndpointAdaptationData

Content	Field
UDP/IP parameters	
UDP/IP parameters • *Nested data structure, see Table 5* • *In a Redundancy Scheme, an independent set of these parameters shall be defined for each network interface used. Each parameter set shall be represented by one* Aes67EndpointIPParameters *list item.*	*OcaList<Aes67EndpointIPParameters>* IpParameters
RTP parameters	
RTP payload type • *See [AES67(Payload types)] and Annex clause C.3.2.*	*OcaUint8* PayloadType
RTP timestamp offset • *See [AES67(Media clocks)] and [AES67(RTP and media clock)].Shall be zero for ST2110-30 senders.*	*OcaUint32* RtpTimestampOffset
Packet arbitrator parameters (see clause 11.4.3.2)	
Maximum tolerated skew • *Only used for redundant streams*	*OcaTimeInterval* TimeSkewMaxTolerated
Other Parameters	
Link offset • *Link offset for this stream on the receiving device* • *See [AES67(Link offset)]*	*OcaTimeInterval* LinkOffset
Media information (optional) • *Application-specific text description of stream.*	*OcaString* MediaInfo
Submitted SDP (clause 11.2.2.1) • *SDP string supplied to the most recent call to SubmitSDP().*	*OcaSDPString* SubmittedSDP
Active SDP (clause 11.2.2.2) • *SDP string currently in effect for the endpoint. It shall be returned by calls to GetActiveSDP().*	*OcaSDPString* ActiveSDP

Table 4. Datatype Aes67EndpointIPParameters

Content	Field	
ID of network assignment to which this set of parameters shall be applied	OcaID16	NetworkAssignmentID
IP source address	OcaString	SourceAddress
IP destination address	OcaString	DestinationAddress
IP Time to Live (TTL)	OcaUint8	TimeToLive
UDP source port	OcaUint16	SourcePort
UDP destination port	OcaUint16	DestinationPort
COS parameter (QoS layer 2) • *-1 means leave up to device implementation*	OcaInt8	COS
DSCP parameter (QoS layer 3) • *-1 means leave up to device implementation*	OcaInt8	DSCP

10.5.2. Aes67EndpointAdaptationData field values for specific use cases

There are four distinct use cases for Aes67EndpointAdaptationData: unicast source, multicast source, unicast sink, and multicast sink. These cases are specified normatively in Table 5 through Table 8, respectively. For convenience, the rightmost columns of Table 5 through Table 8 informatively list Session Description Protocol (SDP) elements that correspond to the elements of the Aes67EndpointAdaptationData datatype . More information about these elements is in Annex C.

Table 5. Aes67EndpointAdaptationData field values for unicast source endpoint

Field	Content	Corresponding SDP element (informative)
Aes67EndpointIPParameters		
SourceAddress	IP address of source endpoint - read-only	
DestinationAddress	IP address of sink endpoint	c=connection-address
TimeToLive	IP Time to Live (TTL) value for transmitted packets	
SourcePort	UDP port number used by source endpoint for sending	
DestinationPort	UDP destination port number value for transmitted packets	<port> field of "m=" line
COS	COS value for transmitted packets	
DSCP	DSCP value for transmitted packets	
PayloadType	Payload type of outgoing RTP packets. Value is chosen by controller and must be between 96 and 127 inclusive	<fmt> field of "m=" line
RtpTimestampOffset	RTP timestamp epoch minus media clock epoch; always zero for ST 2110-30.	
TimeSkewMaxTolerated	not applicable; ignored	
LinkOffset	not applicable; ignored	
MediaInfo	Application-specific text description of stream	i=
SubmittedSDP	SDP string supplied to the most recent call to SubmitSDP()	entire SDP string
ActiveSDP	SDP string currently in effect for the endpoint.	entire SDP string

Table 6. Aes67EndpointAdaptationData field values for multicast source endpoint

Field	Content	Corresponding SDP element (informative)
Aes67EndpointIPParameters		
SourceAddress	IP address of source endpoint - read-only	
DestinationAddress	IP multicast group address	c=connection-address
TimeToLive	IP Time to Live (TTL) value for transmitted packets	TTL suffix of c=connection-address
SourcePort	UDP port number used by source endpoint for sending	
DestinationPort	UDP destination port number value for transmitted packets	<port> field of "m=" line
COS	COS value for transmitted packets	
DSCP	DSCP value for transmitted packets	
PayloadType	Payload type of outgoing RTP packets. Value is chosen by controller and must be between 96 and 127 inclusive	<fmt> field of "m=" line
RtpTimestampOffset	RTP timestamp epoch minus media clock epoch; always zero for ST 2110-30.	
TimeSkewMaxTolerated	not applicable; ignored	
LinkOffset	not applicable; ignored	
MediaInfo	Application-specific text description of stream	i=
SubmittedSDP	SDP string supplied to the most recent call to SubmitSDP()	entire SDP string
ActiveSDP	SDP string currently in effect for the endpoint.	entire SDP string

Table 7. Aes67EndpointAdaptationData field values for unicast sink endpoint

Field	Content	Corresponding SDP element (informative)
Aes67EndpointIPParameters		
SourceAddress	Filter criterion for source-specific receive. Null or source IP address value. If non-null, only packets with given value will be accepted.	c=connection-address
DestinationAddress	Own IP address of sink endpoint - read-only	
TimeToLive	not applicable; ignored	
SourcePort	Filter criterion for source-specific receive. Zero or source port value. If nonzero, only packets with given value will be accepted.	<port> field of "m=" line
DestinationPort	UDP port number of sink endpoint	
COS	not applicable; ignored	
DSCP	not applicable; ignored	
PayloadType	Filter criterion. Zero or payload-type value. If nonzero, only packets with given value will be accepted.	<fmt> field of "m=" line
RtpTimestampOffset	RTP timestamp epoch minus media clock epoch; always zero for ST 2110-30.	
TimeSkewMaxTolerated	Maximum tolerated skew	
LinkOffset	Link offset. Determined by controller. See [Aes67(Link Offset)].	
MediaInfo	Application-specific text description of stream	i=
SubmittedSDP	SDP string supplied to the most recent call to SubmitSDP()	entire SDP string
ActiveSDP	SDP string currently in effect for the endpoint.	entire SDP string

Table 8. Aes67EndpointAdaptationData field values for multicast sink endpoint

Field	Content	Corresponding SDP element (informative)
Aes67EndpointIPParameters		
SourceAddress	Filter criterion for source-specific receive. Null or source address value. As an implementation choice, the device may use this value to control the Source-Specific Multicast (SSM) feature ([RFC 4607]) of IP Networks , and/or apply it locally at the receiver's input. In the latter case, if non-null, only packets with given value will be accepted.	
DestinationAddress	IP multicast group address for receiving	c=connection-address
TimeToLive	not applicable; ignored	
SourcePort	Filter criterion for source-specific receive. Zero or source port value. If nonzero, only packets with given value will be accepted.	
DestinationPort	UDP port number of sink endpoint	<port> field of "m=" line
COS	not applicable; ignored	
DSCP	not applicable; ignored	
PayloadType	Filter criterion. Zero or payload-type value. If nonzero, only packets with given value will be accepted.	<fmt> field of "m=" line
RtpTimestampOffset	RTP timestamp epoch minus media clock epoch; always zero for ST 2110-30.	
TimeSkewMaxTolerated	Maximum tolerated skew	
LinkOffset	Link offset. Determined by controller, see [Aes67(Link Offset)].	
MediaInfo	Application-specific text description of stream	i=
SubmittedSDP	SDP string supplied to the most recent call to SubmitSDP()	entire SDP string
ActiveSDP	SDP string currently in effect for the endpoint.	entire SDP string

10.6. Media Stream Mode management

10.6.1. Media Stream Mode parameter mapping

This section normatively defines the mapping of AES67 media Stream Mode parameters to field values of the datatype *OcaMediaStreamMode*, used for the OcaMediaStreamEndpoint property CurrentStreamMode.

Table 9. Datatype OcaMediaStreamMode

AES67 Stream Mode element	OcaMediaStreamMode field	
Current sampling rate	*OcaFrequency*	SamplingRate
Current sample encoding	*OcaMimeType*	EncodingType
Current channel count	*OcaUint16*	ChannelCount
Current packet time	*OcaTimeInterval*	PacketTime

For AES67, the OcaMediaStreamMode field FrameFormat shall have the constant value RTP.

10.6.2. Media Stream Mode capability descriptor mapping

This section normatively defines the mapping of AES67 media Stream Mode capability descriptors (see [AES67(Protocol implementation conformance criteria)]) to AES70 Stream Mode capability descriptors (see AES70-1(Media Stream Mode capabilities)].

10.6.2.1. AES70 descriptors (informative)

As specified normatively in [AES70(Media Stream Mode capabilities)], a Device indicates its media Stream Mode handling capabilities by means of one or more media Stream Mode capability descriptors, each of which is an instance of datatype OcaMediaStreamModeCapability defined normatively in [AES70-2A]. A Stream Mode capability descriptor can express a range of media Stream Mode options.

One or more instances of OcaMediaStreamModeCapability are collected in the structure property OcaMediaTransportApplication.MediaStreamModeCapabilities, which indicates the full Stream Mode capability set actually supported by the containing media transport application.

10.6.2.2. AES67 descriptors (informative)

[AES67] defines its Stream Mode capability descriptor as a text string in the following form:

{<sampling rates>}-{< sample encodings>}-{<channel counts>}-{<packet times>}

with fields as follows:

Field	Contents & Format
<sampling rates>	Comma-separated list of sampling rate designators:, e.g. 48000, 96000, 44100
<sample encodings>	Comma-separated list of sample encoding designators e.g. L16, L24
<channel counts>	Comma-separated list of channel counts, or channel count ranges in double-dot notation, or combination of both. Ranges are indicated with inclusive bounds.
<packet times>	Comma-separated list of packet time designators, or packet time designator ranges in double-dot notation, or combination of both. Ranges are indicated with inclusive bounds

A packet time designator is an integer number representing the packet time rounded to integer microseconds. For 48 kHz and 96 kHz sampling rates, the designator indicates the effective packet time value. For 44.1 kHz sampling rate the designator indicates the packet time of a 48 kHz packet with an equal sample count.

A packet time designator range represents all possible packet times in single-sample increments.

The AES67 Stream Mode capability descriptor is used in the AES67 PICS (Protocol Implementation Conformance Statement), defined normatively in [AES67(Protocol implementation conformance criteria)].

10.6.2.3. Mapping of AES67 descriptors to AES70 descriptors (normative)

Table 10 shows the mapping between AES67 and AES70 Stream Mode capability descriptors.

Table 10. AES67 and AES70 Stream Mode capability descriptors.

Content	AES70 OcaMediaStreamModeCapability field(s)	AES67 Stream Mode capability descriptor element
Capability identifier for the Device	ID	Requirement Identifier*
Capability name for the Device	Name	Requirement Name*
Whether capability applies to inputs, outputs, or both	Direction	(no corresponding element)
Supported media frame formats; always RTP in AES67	FrameFormatList	(no corresponding element)
Supported sampling rates	SamplingRateList	<sampling rates>
Supported encodings	EncodingTypeList	<sample encodings>
Supported channel counts	ChannelCountList ChannelCountRange	<channel counts> - See Note, below.
AES67 media Stream Mode capability descriptor:	PacketTimeList PacketTimeRange	<packet times> - See Note, below.

* Not an element of the descriptor *per se*, but used by [AES67] to identify the capability in the AES67 Protocol Implementation Conformance Statement (PICS).

An AES67 media Stream Mode capability descriptor allows inclusion of multiple range elements for each of the <channel counts> and <packet times> attributes, whereas the OcaMediaStreamModeCapability datatype can contain only one range element per attribute. When an AES67 descriptor contains multiple range elements per attribute, it shall be translated to as many instances of OcaMediaStreamModeCapability as needed to accommodate all of its range elements.

One or both of list and range elements may be specified for channel counts and packet times. If both are specified, the aggregate capability set shall equal the union of the two.

Annex A gives AES70 media Stream Mode capability descriptor templates for AES67 devices.

Annex B gives AES67 Stream Mode capability requirements for ST 2110-30 receiver conformance levels.

11. Related protocol support

To accomplish their media transport missions, the AES67 and ST 2110-30 specify the use of several related protocols. This chapter defines Adaptation mechanisms to support the use of those protocols.

11.1. IEEE 1588

AES67 specifies that IEEE 1588 shall be used as the time delivery mechanism for the purpose of synchronizing capture and playout of audio samples in AES67 streams. AES67, and consequently this Standard, specify the use of IEEE 1588-2008 or a functionally equivalent subset of IEEE 1588-2019.

11.1.1. Class OcaTimeSource

In this Standard, a Device's relationship to an IEEE 1588 time delivery mechanism shall be controlled by an instance of class OcaTimeSource. The properties of such instances shall be set as shown in Table 11.

Table 11. OcaTimeSource properties

Content	Property	Value
Time delivery method	*OcaTimeDeliveryMechanism* TimeDeliveryMechanism	IEEE1588_2008
Time delivery parameters	*OcaParameterRecord* TimeDeliveryParameters	see Clause 11.1.4

11.1.2. IEEE 1588 control parameters and profiles (informative)

IEEE-1588 has many control parameters. These are defined normatively throughout [IEEE 1588-2008]. [RFC 8575] provides a concise definition of these parameters and their names in the form of a YANG specification. The YANG language is specified in [RFC 6020]. In this Standard, the parameter names specified in [RFC 8575] shall be used.

IEEE-1588 groups its control parameters into named datasets[1]. The full identifier of an IEEE-1588 control parameter is of the form <dataset-name>.<parameter-name>.

[IEEE 1588-2008] defines a profile mechanism by which sets of control parameter values can be prespecified to address particular applications.

11.1.3. Profiles and control parameters for AES67 (informative)

AES67 profile requirements given in [AES67(Synchronization)], and are as follows:

- AES67 **requires** implementations to support the default IEEE 1588-2008 profiles, including delay request-response ([IEEE 1588-2008(annex J.3)]) and peer-to-peer ([IEEE 1588(annex J.4)] mechanisms.
- AES67 **recommends** that implementations support the IEEE 1588-2008 **Media profile** defined in [AES67(Annex A)].

[AES67(Synchronization)] lists a number of IEEE-1588 control parameter values that must be used by AES67 implementations. However, only a few of these values need be settable by controllers; the rest can simply be coded directly into implementations. Table 12 lists the parameters that must be settable.

11.1.4. OcaTimeSource.TimeDeliveryParameters

AES70 controllers shall access a Device's IEEE 1588 control parameters via the property OcaTimeSource.TimeDeliveryParameters. This property is an AES70 Parameter Record. A Parameter Record is a JSON object whose properties depend on the application. See [AES70-2(Parameter Records)].

For the TimeDeliveryParameters property, each JSON property name shall represent a single IEEE control parameter for the Device. The property's name shall be the full identifier of the IEEE-1588 parameter it represents.

1 IEEE-1588 datasets are not related to AES70 Datasets, as defined by Core AES70.

Table 12 specifies the Parameter Record properties that shall be in OcaTimeSource.TimeDeliveryParameters. The properties listed constitute the minimum set required by this Standard. If access to additional properties is required, they may be added to the Parameter Record with names constructed as specified in the preceding paragraph.

This Standard defines several AES70-specific Parameter Record properties to represet data items defined in [AES67(Synchronization)]. These are attached to the non-IEEE-1588 dataset "_OCA".

Table 12. Mandatory OcaTimeSource.TimeDeliveryParameters properties

Content	Property	Value	Writable
Protocol version	_OCA.IEEE1588Version	"2008"	N
Identifier of active profile	_OCA.ProfileID	"00-0B-5E-00-01-00"	N
Version of active profile	_OCA.ProfileVersion	"1.0"	N
ID of current grandmaster	parentDS.grandmaster-identity	depends on deployment	Y/N**
Domain number	defaultDS.domain-number	depends on deployment	Y/N**

** Writability depends on Device implementation

11.2. Session Description Protocol (SDP) support (normative)

Session Description Protocol (SDP) [RFC 4566] defines an elaborate description format for representing stream information. AES67 and ST 2110-30 require compliant devices to support the use of SDP for stream connection control.

11.2.1. Concept

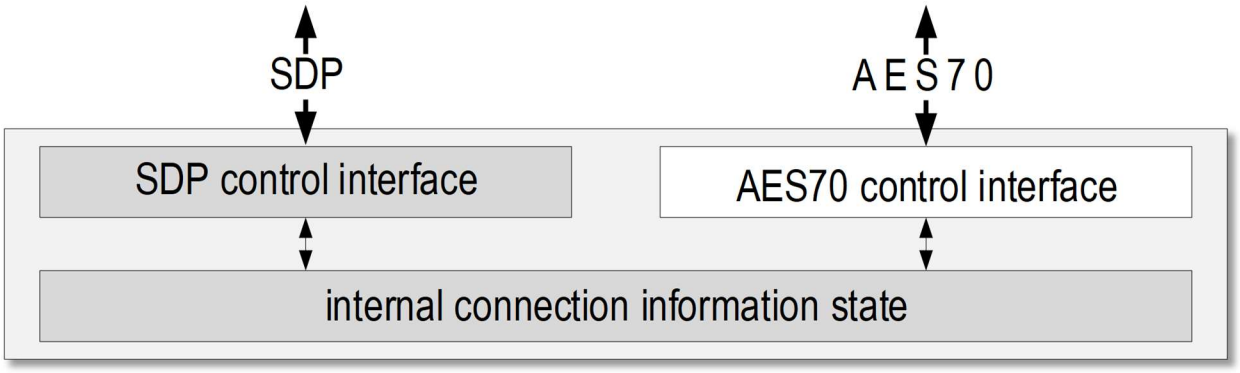
AES70 connection management implements connection control by means of parameters in AES70 method calls and data structures; it does not implement connection control by means of SDP strings directly.

However, AES70 implementations shall offer tunneling of the SDP specification to and from other device mechanisms that support SDP-directed connection control, as required by AES67 and ST 2110-30.

In addition, this Standard informatively indicates correspondences between AES70 parameters and SDP elements.

Coexistence of SDP and AES70 mechanisms is shown in Figure 4. The Device has an SDP control interface and an AES70 control interface. Both of these interfaces are assumed to operate on a common internal connection information state that is specific to the device implementation and hidden from the external interfaces. Either interface can make changes to the internal model, and all such changes are reflected in both interfaces. Thus, a change made via the AES70 control interface will be known to the SDP control interface and vice versa.

The particulars of the SDP control interface are out of scope of this Standard, as are the particulars of the internal connection model.



Grey boxes are out of scope of this standard.

Figure 4. Compliant Device with coexisting SDP and AES70 mechanisms.

11.2.2. SDP mechanisms

A device that complies with this Standard shall provide the following mechanisms in its AES70 control interfaces, to support integration of SDP-dependent devices into systems using AES70:

1. SDP Submission
2. SDP Monitoring.

These mechanisms are detailed in the following sections. In those sections, the methods specified are elements of Aes67OcaMediaTransportApplication. They are illustrated conceptually in Figure 5 and specified normatively in clause 11.2.3.

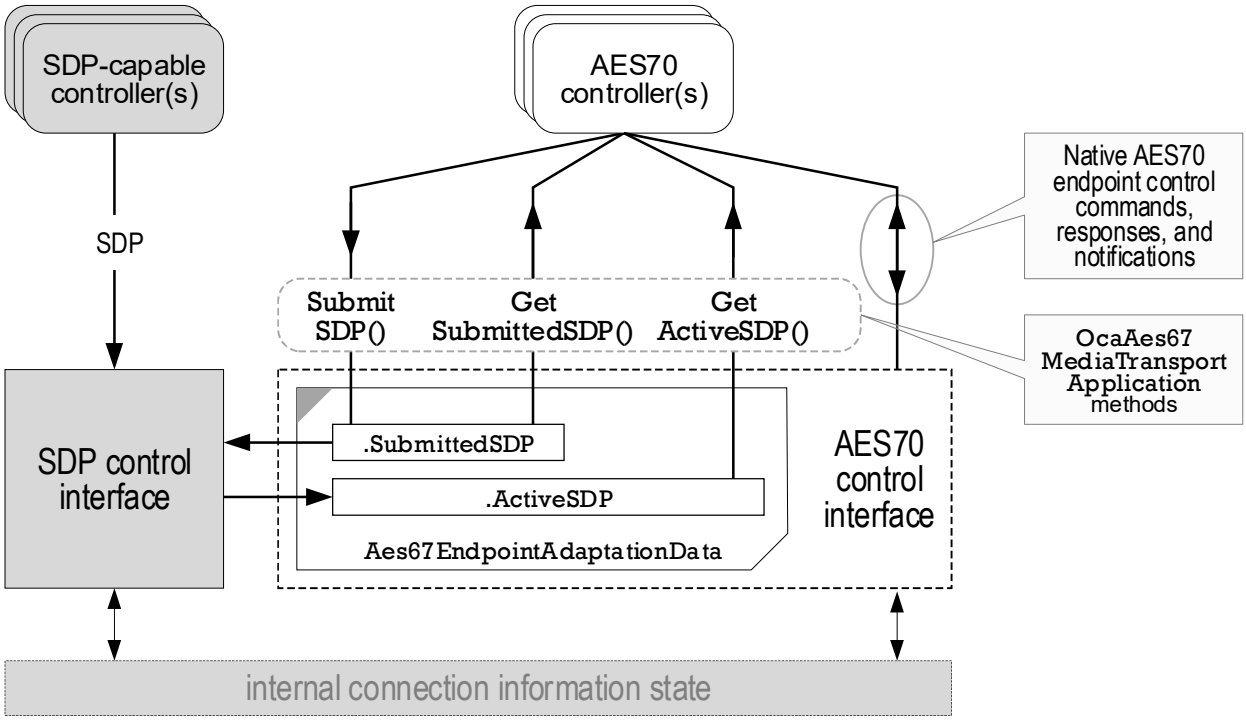
11.2.2.1. SDP Submission

SDP Submission is a mechanism by which an AES70 controller can call the method SubmitSDP() (to submit an SDP string into the device's SDP control interface for processing. This SDP string is neither parsed nor modified by the AES70 control interface - it is simply passed through unprocessed. The submitted string shall be stored in the SubmittedSDP field of the data structure Aes67OcaMediaStreamEndpoint.AdaptationParameters. Its value may be recalled at any time by calling the method GetSubmittedSDP().

11.2.2.2. SDP Monitoring

SDP Monitoring is a mechanism by which a controller can observe the SDP string that the SDP control interface is currently applying to the endpoint. This value shall be maintained in the ActiveSDP field of the data structure Aes67OcaMediaStreamEndpoint.AdaptationParameters. Its value may be recalled at any time by calling the method GetActiveSDP().

Note: By subscribing to changes of property Aes67OcaMediaTransportApplication.Endpoints, a controller can keep itself informed of changes to active SDP values.



Grey boxes are out of scope of this standard.

Figure 5. SDP mechanisms

11.2.3. SDP methods in Aes67OcaMediaTransportApplication

Aes67OcaMediaTransportApplication shall add the following methods to its parent base class OcaMediaTransportApplication. In the c++-like pseudocode below a prefix asterisk denotes a returned parameter:

```
// SDP Submission mechanism

OcaStatus SubmitSDP(                    // Shall submit an SDP string to the device's SDP implementation
    OcaID32 EndpointID;                 // ID of endpoint to which SDP string is being submitted
    OcaSDPString SDPtext;               // supplied SDP text
);

OcaStatus GetSubmittedSDP(              // Shall return the current value of the SubmittedSDP field
    OcaID32 EndpointID;                 // ID of endpoint whose submitted SDP string is being retrieved
    *OcaSDPString SDPtext;              // Value of SubmittedSDP field.
);

// SDP Monitoring mechanism

OcaStatus GetActiveSDP (                // Shall return the current value of the ActiveSDP field.
    OcaID32 EndpointID;                 // ID of endpoint whose active SDP string is being retrieved
    *OcaSDPString SDPtext;              // Value of ActiveSDP field.
);
```

11.2.4. ST 2110-30 channel ordering support (informative)

[ST 2110-30] defines a method for stream senders and receivers to include channel ordering information in the SDP strings that specify stream connections (see [RFC 3190]). Further, [ST 2110-30] defines a set of standard channel ordering options for compliant devices to support.

The handling of channel ordering specifications included in SDP strings is out of scope of this Standard, because:

- The focus of AES70 connection management features is the interconnection of streams, whereas stream's internal layout by channel content is considered to be an application concept of a higher semantical level; and
- AES70 control interface does not implement connection control by means of SDP strings.

That said, every AES70 media stream endpoint does include a mechanism - the *channel map* - for directing stream signal channels to and from internal device signal channels in any desired order. A controller wishing to support ST 2110-30 channel ordering may use this mechanism to implement the channel mappings it requires, as discovered from parsing the relevant SDP strings or by other means.

The channel map of a media stream endpoint shall reside in the property ChannelMap of the OcaMediaStreamEndpoint datatype.

11.3. Session Initiation Protocol (SIP) support (normative)

[AES67] specifies that AES67 implementations shall support the making and breaking of stream connections using the Session Initiation Protocol (SIP) [RFC 3261]. This Standard specifies the use of AES70 session control features to control SIP operations.

[AES67] does not restrict the use of other session control means, but such means are outside the scope of this (AES70) standard.

11.3.1. Class Aes67OcaMediaTransportSessionAgent

This Standard defines the Aes67OcaMediaTransportSessionAgent class that adds a SIP session control mechanism to the Core AES70 parent class OcaMediaTransportSessionAgent.

For each session, Aes67OcaMediaTransportSessionAgent shall create and maintain an instance of the Aes67OcaMediaTransportSession datatype.

In inheriting from its parent, Aes67OcaMediaTransportSessionAgent shall include two Adaptation-specific features.

1 The datatype of the property Sessions shall be OcaList<Aes67OcaMediaTransportSession> (clause 11.3.2).

2 Methods shall be included for accessing the elements of Sessions.

11.3.2. SIP parameters and the datatype Aes67OcaMediaTransportSession

SIP has a number of parameters that are passed with SIP commands for Connection Negotiation and control. [RFC 3261] refers to such parameters as ***header fields***. [AES67] mandates the use of only a few of

them. However, implementations may need to access additional ones. This Standard specifies features that provide controllers with access to all header fields, not just to the ones required by [AES67].

For this Standard, the datatype of the property Aes67OcaMediaTransportSession.AdaptationData shall be OcaParameterRecord, and the value of the property shall be an AES70 Parameter Record (see [AES70-2 (Parameter records)]) that contains values of SIP header fields and other parameters.

11.3.3. Property Aes67OcaMediaTransportSession.AdaptionData's SIP Parameter Record

The Aes67OcaMediaTransportSession.AdaptionData Parameter Record shall contain parameters of the following two categories:

1. Header field parameters (clause 11.3.5).
2. Non-header-field parameters (clause 11.3.6).

These parameters are listed in Table 13. Names of non-header field parameters shall begin with "_".

Table 13. Aes67OcaMediaTransportSession.AdaptationData Parameter Record elements

Element	Content	Notes
to	sip: or sips: URI of call target	SIP header field, writeable, required. Example: "<sip:alice@atlanta.com>"
from	sip: or sips: URI of caller	SIP header field, writeable, required. Example: "<sip:me@mysite.com>"
user-agent	user-agent descriptor	SIP header field, read-only, optional. See [AES67(User-Agent)] and [RFC 2616]. Example: "CERN-LineMode/2.15 libwww/2.17b3v"
_server	IP hostname or address of SIP server or proxy	Non-header-field parameter, writeable, conditional. Required in server mode. Unused in peer-to-peer mode, where the local SIP user agent sends connection requests directly to the host identified in the "to:" URI. Example: "sip.mysite.com"
_sdpInitial	Initial SDP string to be offered in Connection Negotiation	Non-header-field parameter, writeable, conditional. May be omitted if the initial SDP string is obtained by other means, e.g. if it is preconfigured into the device. Content is SDP text.
_sdpFinal	Final SDP string resulting from Connection Negotiation	Non-header-field parameter, read-only, required. Content is SDP text.

When a connection has been set up via SIP:

1. The device shall fill the _sdpFinal field of the AdaptationData Parameter Record;
2. The device shall update the relevant AES70 control interface parameters of the connection's local endpoint.

11.3.4. AdaptationData access methods

Aes67OcaMediaTransportSession shall provide four methods for accessing the AdaptationData property. The methods are as follows ([out] designates a returned parameter):

Retrieve entire Parameter Record:
GetSIPParameterRecord(*OcaID16* SessionID, [out] *OcaParameterRecord* rec)

Set entire Parameter Record:
SetSIPParameterRecord(*OcaID16* SessionID, *OcaParameterRecord* rec)

Retrieve an individual parameter value:
GetSIPParameter(*OcaID16* SessionID, [out] *OcaString* ParameterKey, *OcaJsonValue* ParameterValue)

Set an individual parameter value:
SetSIPParameter(*OcaID16* SessionID, *OcaString* ParameterKey, *OcaJsonValue* ParameterValue)

where SessionID shall be the ID of the Aes67OcaMediaTransportSession instance being accessed.

In a call to GetSIPParameter or SetSIPParameter:

- The parameter being accessed shall be identified by the string ID value in the ParameterKey argument.
- The parameter value shall be expressed as a JSON [RFC 8259] string value in the ParameterValue argument.

Rules for constructing these arguments are given in clause 11.3.5 for header field parameters, clause 11.3.6 for non-header field parameters.

11.3.5. GetSIPParameter() and SetSIPParameter() arguments for header field parameters

[RFC 3261 (7.3.1)] defines SIP header field syntax as follows:

hdr-field-name: hdr-field-value *(;hdr-parameter-name=hdr-parameter-value)
e.g.
To: <sip:operator@cs.columbia.edu>;tag=287447

Header field elements shall be mapped to SipParameterRecord access method arguments as follows:

- hdr-field-name shall be mapped to ParameterKey
- **entire header field content** shall be mapped to ParameterValue

According to [RFC 3261], header field names and header parameter names are processed in a case-insensitive manner. For the AES70 mappings described here, all header field names and header parameter names shall be in lower case.

Example

SIP header fields:

To: <sip:operator@cs.columbia.edu>;tag=287447
to: <sip:operator@cs.columbia.edu>;tAg=287447

Are both translated to the method arguments:

ParameterKey: "to"
ParameterValue: "to:<sip:operator@cs.columbia.edu>;tag=287447"

11.3.6. GetSIPParameter() and SetSIPParameter() arguments for non-header-field parameters

For non-header-field parameters, SipParameterRecord access method arguments shall be as follows:

- ParameterKey: Name of the non-header-field parameter
- ParameterValue: JSON value of the non-header-field parameter

11.4. ST 2022-7 redundancy support

The ST 2022-7 standard defines a scheme for using multiple redundant IP Networks to increase media stream transport reliability. ST 2110-30 (but not AES67) specifies support for SMPTE ST 2022-7 as an option. The intended result of this approach is what ST 2022-7 calls ***Seamless Reconstruction***, in which a media stream, when transmitted over a set of unreliable IP connections, is received either fully intact or with a packet loss ratio significantly lower than that of any one of the IP connections alone.

In this document, the term ***Redundancy Scheme*** designates a set of IP Networks and AES70 mechanisms assembled and configured to implement the Seamless Reconstruction as defined by ST 2022-7.

11.4.1. ST 2022-7 mechanism (informative)

For reliable transport of RTP packets over IP Networks, ST 2022-7 defines a mechanism for using multiple IP connections in parallel, with receivers selecting acceptable media packets from the parallel connections on a packet-by-packet basis. A packet is deemed acceptable if it arrives within a certain time window and is otherwise intact.

The element responsible for selecting packets from the parallel connections is referred to by this Standard as a ***Packet Arbitrator***.

ST 2022-7 defines four time skew tolerance classes , intended to suit networks of diverse performance properties. This Standard defines the time skew tolerance setting in physical time units.

An AES70 controller may select a time skew value specified for one of the ST 2022-7 classes or choose other values if desired.

11.4.2. Core AES70 redundancy mechanisms (informative)

The Core AES70 standard defines a pair of interrelated mechanisms to support Redundancy Schemes. See clause 10 for definitions of the classes and objects involved.

1. **A NAC stack can use multiple network interfaces** by including multiple network interface objects**.** Such NAC stacks are constructed by incorporating multiple OcaNetworkInterface objects and instantiating OcaNetworkInterfaceAssignment items that link those network interface objects to the Aes67OcaMediaTransportApplication object.

2. **A media stream endpoint can receive packets from multiple network interfaces.** The property that supports this mechanism is OcaMediaStreamEndpoint.NetworkInterfaceAssignmentIDs, which contains IDs of the OcaNetworkInterfaceAssignment items that link to the OcaNetworkInterface objects that represent the network interfaces the endpoint uses.

11.4.3. Adaptation details (normative)

This section details this Standard's specifications for support of ST 2022-7 redundancy, using the mechanisms summarized above.

11.4.3.1. NAC stack configuration

A NAC stack that supports the ST 2022-7 Redundancy Scheme shall be configured as follows:

1. The stack shall incorporate an OcaNetworkInterface object for each data network interface used.
2. Each OcaNetworkInterface object shall be configured as specified in clause 10.
3. The stack shall include a single Aes67OcaMediaTransportApplication object, configured with the usual ancillary classes and data structures as described in clause 10.
4. All the OcaNetworkInterface objects shall be assigned to the Aes67OcaMediaTransportApplication object as described in clause 10.2.1.
5. The Aes67OcaMediaTransportApplication object may contain some endpoints that use ST 2022-7 redundancy and some that do not. Every the endpoint that does use ST 2022-7 shall be configured as described in clause 11.4.3.2.

NOTE: The design of this scheme allows a single Aes67OcaMediaTransportApplication object to support multiple redundant and non-redundant schemes using diverse sets of networks.

11.4.3.2. OcaMediaStreamEndpoint configuration

For each instance of OcaMediaStreamEndpoint that uses the Redundancy Scheme:

- The ID of each network interface assignment the endpoint uses shall be listed in OcaMediaStreamEndpoint.NetworkInterfaceAssignmentIDs.
- An appropriate time skew tolerance value shall be set in property Aes67EndpointAdaptationData.TimeSkewMaxTolerated.

The full normative specification of Aes67EndpointAdaptationData is in clause 10.5.1.

Annex A. (informative) Media Stream Mode Capability Descriptor templates for AES67

A.1. General

This annex is informative and gives a set of AES70 media Stream Mode capability descriptor templates matching the media Stream Mode capability requirements of [AES67] at the moment of writing. The actual Stream Mode support requirements themselves are defined by [AES67] and are outside the scope of this Standard. [AES67] and [ST 2110-30] may evolve over time, which may obsolete the templates given in this annex.

OcaMediaStreamModeCapability field values specified below cover the fullest Stream Mode support requirements of the current revision of [AES67], including the mandatory ('shall") , the recommended ("should"), and the additional ("may") Stream Modes within the scope of AES67. The actually translated capability descriptors may be equal to, or be subsets or supersets of the templates specified in this annex.

The templates given in this annex are transmitter/receiver-non-specific. The transmitter and receiver capability descriptors shall be translated independently, and they may differ between the I/O directions.

Stream modes exceeding the MTU size are excluded from AES67 implementations and, consequently, they shall not appear in the Stream Mode capability descriptors. Although some of the templates given in this annex allow attribute combinations exceeding the MTU size, such combinations would be excluded from the original AES67 capability descriptors and would not be translated into AES70 capability descriptors.

A.2. OcaMediaStreamModeCapability templates for AES67

A.2.1. Template for AES67 requirements T1 "T-48k-1ms" and R1 "R-48k-1ms"

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= (L16,L24}	16- and 24-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {48000}	48 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= {1,8,0x0003}	1 to 8 channels
OcaList<OcaTimeInterval>	PacketTimeList	= {1000}	1000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

A.2.2. Template for AES67 requirements T2 “T-96k-1ms” and R2 “R-96k-1ms”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L24}	24-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {96000}	96 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= (1,5,0x0003}	1 to 5 channels
OcaList<OcaTimeInterval>	PacketTimeList	= {1000}	1000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

A.2.3. Template for AES67 requirements T3 “T-44k-1ms” and R3 “R-44k-1ms”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L16}	16-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {44100}	44.1 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= {1,8,0x0003}	1 to 8 channels
OcaList<OcaTimeInterval>	PacketTimeList	= {1000}	1000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

A.2.4. Template for AES67 requirements T4 “T-48k-non1ms” and R4 “R-48k-non1ms”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L16,L24}	16- and 24-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {48000}	48 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	={1,8,0x0003}	1 to 8 channels
OcaList<OcaTimeInterval>	PacketTimeList	={125,250,333,4000}	125, 250, 333, 4000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

Note: This template allows attribute combinations exceeding the MTU size.

A.2.5. Template for AES67 requirements T5 “T-96k-non1ms” and R5 “R-96k-non1ms”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L24}	24-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {96000}	96 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= {1,8,0x0003}	1 to 8 channels
OcaList<OcaTimeInterval>	PacketTimeList	={125,250, 333,4000}	125, 250, 333, 4000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

Note: This template allows attribute combinations exceeding the MTU size.

A.2.6. Template for AES67 requirements T6 “T-44k-non1ms” and R6 “R-44k-non1ms”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L16}	16-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {44100}	44.1 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= {1,8,0x000F}	1 to 8 channels
OcaList<OcaTimeInterval>	PacketTimeList	={125,250,333,4000}	125, 250, 333, 4000 µs
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

Note: This template allows attribute combinations exceeding the MTU size.

A.2.7. Template for AES67 requirements T7 “T-other” and R7 “R-other”

Datatype	Field	Value	Comment
OcaID16	ID	(set by implementation per device capability)	
OcaString	Name	(set by implementation per device capability)	
OcaBitSet16	Direction	(set by implementation per device capability)	
OcaList<OcaMediaFrameFormat>	FrameFormatList	= RTP	constant for AES67
OcaList<OcaMimeType>	EncodingTypeList	= {L16,L24}	16,24-bit fixed-point LPCM
OcaList<OcaFrequency>	SamplingRateList	= {44100,48000, 96000}	44.1 kHz, 48 kHz, 96 kHz
OcaList<OcaUint16>	ChannelCountList	= {}	
OcaInterval<OcaUint16>	ChannelCountRange	= {1,8,0x0003}	1 to 8 channels; may be extended with other channel counts
OcaList<OcaTimeInterval>	PacketTimeList	={125,250, 333, 1000,4000}	125, 250, 333, 1000, 4000 µs; may be extended with other packet times
OcaInterval<OcaTimeInterval>	PacketTimeRange	= {}	

Note: This template allows attribute combinations exceeding the MTU size.

Annex B. (informative) AES67 Stream Mode capability requirements for ST 2110-30 receiver conformance levels

B.1. General

[ST 2110-30] does not explicitly list Stream Mode capability requirements. ST 2110-30 senders and receivers shall comply in entirety with media Stream Mode capability requirements for AES67 devices - see [AES67]. For AES67 devices, the field values of OcaMediaStreamModeCapability, the AES70 media Stream Mode capability descriptor, shall be populated as defined in Annex A.

Additionally, [ST 2110-30] defines a number of "Receiver Conformance Levels", to assist interoperability assessment. This annex gives mapping of ST 2110-30 receiver conformance levels to Stream Mode capability requirements of AES67.

[ST 2110-30] requires support of conformance level A; support of the other levels is optional.

B.2. ST 2110 Level A

ST 2110-30 receiver conformance Level A includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time

To conform to Level A, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety

B.3. ST 2110 Level AX

ST 2110-30 receiver conformance Level AX includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time.
- Reception of 96 kHz streams with 1 to 4 channels, with 1 ms packet time.

To conform to Level AX, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety
- AES67 requirement R2 "R-96k-1ms", excluding channel count 5

B.4. ST 2110 Level B

ST 2110-30 receiver conformance Level B includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time.
- Reception of 48 kHz streams with 1 to 8 channels, with 125 μs packet time.

To conform to Level B, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety
- AES67 requirement R4 "R-48k-non1ms" – subset 125 μs modes only

B.5. ST 2110 Level BX

ST 2110-30 receiver conformance Level BX includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time
- Reception of 96 kHz streams with 1 to 4 channels, with 1 ms packet time
- Reception of 48 kHz streams with 1 to 8 channels, with 125 μs packet time
- Reception of 96 kHz streams with 1 to 8 channels, with 125 μs packet time

To conform to Level BX, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety
- AES67 requirement R2 "R-96k-1ms", excluding channel count 5
- AES67 requirement R4 "R-48k-non1ms" – subset 125 μs modes only
- AES67 requirement R5 "R-96k-non1ms" – subset 125 μs modes only

B.6. ST 2110 Level C

ST 2110-30 receiver conformance Level C includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time
- Reception of 48 kHz streams with 1 to 64 channels, with 125 μs packet time

To conform to Level C, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety
- AES67 requirement R4 "R-48k-non1ms" – subset 125 μs modes only, all channel counts 1 to 64

B.7. ST 2110 Level CX

ST 2110-30 receiver conformance Level CX includes:

- Reception of 48 kHz streams with 1 to 8 channels, with 1 ms packet time
- Reception of 96 kHz streams with 1 to 4 channels, with 1 ms packet time
- Reception of 48 kHz streams with 1 to 64 channels, with 125 μs packet time
- Reception of 96 kHz streams with 1 to 32 channels, with 125 μs packet time

To conform to Level CX, the ST 2110-30 receiver is required to support:

- AES67 requirement R1 "R-48k-1ms" in entirety
- AES67 requirement R2 "R-96k-1ms", excluding channel count 5
- AES67 requirement R4 "R-48k-non1ms" – subset 125 μs modes only, all channel counts 1 to 64
- AES67 requirement R5 "R-96k-non1ms" – subset 125 μs modes only, all channel counts 1 to 32

Annex C. (informative) Related Session Description Protocol (SDP) elements

C.1. General

This informative Annex describes certain Session Description Protocol (SDP) [RFC 8866] elements that are mentioned in Table 5 through Table 8 in the main body of this document. These elements are in SDP text lines as follows:

1. Media description (**m=…** line)
2. RTP map (**a=rtpmap…** line)
3. Connection information (**c=…** line)
4. Session information (**i=…** line)

This Annex describes these elements and the relationships of their values to the parameters in Table 5 through Table 8.

C.2. Structure of an SDP description

From [RFC 8866]:

> *An SDP description consists of a session-level section followed by zero or more media descriptions. The session-level section starts with a "**v=**" line and continues to the first media description (or the end of the whole description, whichever comes first).*
>
> *Each media description starts with an "**m=**" line and continues to the next media description or the end of the whole session description, whichever comes first. In general, session-level values are the default for all media unless overridden by an equivalent media-level value.*

C.3. Media description (m=… line)

The general format of an SDP media description is:

m=<media> <port-spec> <proto> <fmt> …

Fields are as follows:

<media>	media type. For AES70, the value is "audio".
<port-spec>	specifies the transport port or ports to which the media is sent. For this Standard, the value is an IP version 4 UDP port number.
<proto>	name of media transport protocol. For this Standard, the value is "RTP/AVP", which denotes RTP [RFC3550] used under the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] running over UDP.
<fmt>	media format description identifier. For RTP/AVB protocols, <fmt> is an *RTP payload type* number that identifies a particular media encoding and clock rate - see below.

C.3.1. Destination port <port-spec>

As described above in C.3 of this Annex, <port-spec)> is part of the media description "m=" line which specifies the transport port to which the media is sent.

The meaning of the transport port depends on the network being used as specified in the relevant "c=" line, and on the transport protocol defined in the <proto-spec> subfield of the media-field.

Where hierarchically encoded streams are being sent to a unicast address, it may be necessary to specify multiple transport ports. This is done using a similar notation to that used for IP multicast addresses in the "c=" line:

m=<media> <port>/<number of ports> <proto> <fmt> ...

In such a case, the ports used depend on the transport protocol. For RTP, the default is that only the even-numbered ports are used for data, with the corresponding one-higher odd ports used for the RTCP [RFC3550] belonging to the RTP session, and with <number of ports> denoting the number of RTP sessions. For example:

m=audio 49232/2 RTP/AVP 98

would specify that ports 49232 and 49233 form one RTP/RTCP pair, and 49234 and 49235 form the second RTP/RTCP pair. RTP/AVP is the transport protocol, and 98 is the format (see the description of <fmt> in C.3.2).

For more complex <port-spec> and connection information requirements, refer to [RFC8866].

C.3.2. RTP payload type

Table 5 through Table 8 specify a PayloadType parameter that contains the RTP payload type number that applies to an AES67 streaming connection.

RTP payload type numbers fall into two ranges, as follows:

- Types in the range 0...95 are called *static* types; their meanings are specified in [RFC3550].
- Types in the range 96...127 are called *dynamic* types; their meanings are application-defined.

Each dynamic payload type value identifies a particular media encoding specification in an accompanying **a=rtpmap…** element - see next.

AES67's media formats are not in the list of standard static types, and therefore must always be represented by dynamic types; hence the payload type range for this Standard is 96...127.

C.4. RTP map (a=rtpmap … line)

A media encoding specification is given in an **a=rtpmap…** SDP line. The general format is as follows:

a=rtpmap:<payload-type> <encoding-name>/<sampling-frequency>/<channel-count>

Fields are as follows:

<payload-type>	RTP payload type number that identifies this media encoding specification
<encoding-name>	Programmatic name of the encoding type
<sampling-frequency>	Sampling frequency for this media encoding specification
<channel-count>	Number of audio channels in the media stream

In this Standard, these fields correspond to properties of the OcaMediaStreamMode (Table 9) and OcaMediaStreamModeCapability (Table 10) datatypes as follows:

Field	OcaMediaStreamMode	OcaMediaStreamModeCapability
<encoding-name>	MediaEncoding	MediaEncodingList
<sampling-frequency>	SamplingRate	SamplingRateList
<channel-count>	ChannelCount	ChannelCountList

The semantics of these datatypes and their properties are specified in clause 10.6.

C.4.1. Examples

The following is a typical media encoding specification:

```
m=audio 49232 RTP/AVP 98
```

This specification means:

Media type is **audio**
The stream connection uses port **49232**
Media transport protocol is **RTP/AVB**
Payload type number is **98 .**

The following example illustrates the use of the media encoding specification with the media description:

```
m=audio 49232 RTP/AVP 98
a=rtpmap:98 L16/44100/2
```

These two lines describe a two-channel connection that is L16 encoded and has a sampling frequency of 44100 Hz. In this example, the payload-type value is 98, but any value from 96 to 127 could be used with identical effect.

C.4.2. AES67 media encoding types and sampling rates

[AES67] identifies two media encoding types:

L16 16-bit linear format defined in [RFC 3551(4.5.11)]
L24 24-bit linear format defined in [RFC 3190(4)]

... and three sampling frequencies: 48kHz, 44.1kHz, and 96kHz.

As specified in [AES67(Encoding and streaming)] various combinations of these values are mandatory or optional. The requirements differ for senders and receivers. Please refer to the cited chapter for details.

C.5. Connection Information (c=... line)

The Connection Information ("c=") line of an SDP description contains information necessary to establish a network connection and has the following format:

```
c=<net-type> <addr-type> <connection-address>[/<optional-subfield>...]
```

Elements are as follows:

<net-type> Text string giving the type of network. [RFC8866] currently only defines the type "IN", which means "Internet".

<addr-type> Address type. [RFC8866] currently only defines "IP4" and "IP6" for IP versions 4 and 6, respectively.

<connection-address> Connection address.

<optional-subfield> Optional additional connection address detail, depending on the value of <addr-type>.

C.5.1. Connection Information requirement

[RFC 8866] requires a session description to contain:

- A single Connection Information line at the session level; or
- At least one Connection Information line in each media description; or
- A single session-level Connection Information line and additional media-level Connection Information line(s) in the media descriptions, in which case the media-level values override the session-level settings for the respective media.

C.5.2. Connection addresses for AES67

For AES67, <addr-type> is always "IP4" and the connection address is defined as follows:

- **Unicast sessions**. The connection address contains the unicast IP address of the expected data source, data relay, or data sink as determined by additional attribute-fields (**"a="**) - see [RFC8866] Section 5.13.
- **Multicast sessions:**
  - Connection address is the IP multicast group address.
  - A Time To Live (TTL) value is subfield is appended to the multicast address.

    The TTL and the address together define the scope with which multicast packets sent in this session will be sent. TTL values are required to be in the range 0-255.

For example:

```
c=IN IP4 233.252.0.1/127
```

This specification means:

Network type is **IN** (internet)
Address type is **IP4** (IP version 4)
Connection address is **233.252.0.1**
Time to live is **127 .**

C.6. Session information ("i=..." line)

The optional Session Information ("**i=**") line of an SDP description provides textual information about the session. Such lines can appear in two places within an SDP description, the first at the session-level, the second at the media-level.

There can be at most one Session Information line per session description, and at most one Session Information line in each media description. When a media-level Session Information line is not provided, the session-level line applies to that media description.

In this Standard, session information text is given in the MediaInfo property of the structure Aes67EndpointAdaptationData. See clause 10.5.2 and especially Table 3.

C.6.1. Example

The following is an example of the Session Information line used at both the session-level and the media-level:

```
v=0
o=jdoe 3724394400 3724394405 IN IP4 198.51.100.1
s=Call to John Smith
i=SDP Offer #1
u=http://www.jdoe.example.com/home.html
e=Jane Doe <jane@jdoe.example.com>
p=+1 617 555-6011
c=IN IP4 198.51.100.1
t=0 0
m=audio 49230 RTP/AVP 96 97 98
i=Three audio streams
a=rtpmap:96 L16/48000
a=rtpmap:97 L16/48000/2
a=rtpmap:98 L24/96000/2
```

Session-level parameters

Media-level parameters