Annex C
(informative)

AVDECC Proxy Protocol

C.1. Overview

This annex defines the AVDECC Proxy Protocol version zero (0). Future protocol versions may define new commands, responses and/or header formats.

The AVDECC Proxy Protocol (APP) allows an AVDECC Proxy Client (APC) to send and receive ADP, ACMP and AECP PDU’s to a local or remote network via a AVDECC Proxy Server (APS).

The APS is advertised on the local network via DNS-SD.

An APC may:

— Choose an AVDECC Proxy Server automatically based on its priority as it is discovered via DNS-SD.
— Allow a user to select any of the AVDECC Proxy Servers discovered via DNS-SD.
— Allow a user to manually specify the AVDECC Proxy Server’s network address / hostname, IP port, and path.

APP uses the HTTP protocol’s CONNECT method (defined in RFC 2616 Section 9.9) to initiate the tunneling of AVDECC APP messages between the APS and APC. An APC and APS may support and/or require the usage of HTTP authentication and TLS.

Upon initial connection, the APC may request the APS to generate an Entity GUID for the APC to use as its Controller GUID.

C.2. DNS-SD Service Name

An APS shall advertise itself using DNS-SD on the local network using the DNS-SD service name as detailed in Table C.1.

<table>
<thead>
<tr>
<th>IP Protocol</th>
<th>Description</th>
<th>DNS-SD Service Name</th>
<th>Recommended IANA assigned IP Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP</td>
<td>AVDECC Proxy</td>
<td>_avdecc._tcp.</td>
<td>17221</td>
</tr>
</tbody>
</table>

C.3. DNS-SD TXT Record

An APS advertising with DNS-SD shall advertise the TXT record fields as detailed in Table C.2.
Table C.2. DNS-SD TXT Record Items for APP

<table>
<thead>
<tr>
<th>Key</th>
<th>Value Type</th>
<th>Max Length (octets)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Decimal integer ASCII string</td>
<td>4</td>
<td>Highest APP version supported</td>
</tr>
<tr>
<td>Priority</td>
<td>Decimal integer ASCII string</td>
<td>4</td>
<td>User settable APP priority</td>
</tr>
<tr>
<td>Description</td>
<td>UTF-8 String</td>
<td>64</td>
<td>Human readable description of the proxy</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>UTF-8 String</td>
<td>64</td>
<td>Human readable manufacturer of the proxy</td>
</tr>
<tr>
<td>path</td>
<td>UTF-8 String</td>
<td>128</td>
<td>Path to use for initial connection</td>
</tr>
</tbody>
</table>

C.3.1. Version

The **Version** key contains a decimal integer ASCII string which represents the highest APP version that the APS supports.

C.3.2. Priority

The **Priority** key contains a decimal integer ASCII string which represents the priority of the APS, where a larger value represents a higher priority.

An APC that is set to automatically choose an APS selects the highest priority APS that is accepting connections.

C.3.3. Description

The **Description** key contains a UTF-8 encoded description of the APS which would assist a user that needs to manually select an APS.

C.3.4. Manufacturer

The **Manufacturer** key contains a UTF-8 encoded manufacturer name of the APS.

C.3.5. path

The **path** key contains a UTF-8 string that the APC shall use for the 'path' parameter when initiating a connection. The **path** key shall include the leading '/' (ASCII value 2f₁₆) character. If the **path** key does not exist, it is assumed to be '/'.

C.4. APPDU format

The APPDU format is detailed by Figure C.1.
The APPDU contains the following fields:

— **version** (APP version): 1 octet
— **message_type** (Message Type): 1 octet
— **payload_length** (Payload length in octets): 1 doublet
— **address** (MAC Address): 6 octets
— **payload** (Message payload): 0 to 1490 octets

### C.4.1. version field

The **version** field represents the version of the APP protocol used and is set to zero (0) for this version of APP.

### C.4.2. message_type field

The **message_type** field is set to one of the values detailed in Table C.3.

#### Table C.3. message_type Field Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Function</th>
<th>Meaning</th>
<th>Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 16</td>
<td>NOP</td>
<td>No operation.</td>
<td>C.5.1.1</td>
</tr>
<tr>
<td>01 16</td>
<td>ENTITY_GUID_REQUEST</td>
<td>Request an Entity GUID from the APS for the APC</td>
<td>C.5.1.2</td>
</tr>
<tr>
<td>02 16</td>
<td>ENTITY_GUID_RESPONSE</td>
<td>Response of an Entity GUID for the APC from the APS</td>
<td>C.5.1.3</td>
</tr>
<tr>
<td>03 16</td>
<td>LINK_UP</td>
<td>Network port link is active.</td>
<td>C.5.1.4</td>
</tr>
<tr>
<td>04 16</td>
<td>LINK_DOWN</td>
<td>Network port link is inactive.</td>
<td>C.5.1.5</td>
</tr>
<tr>
<td>05 16</td>
<td>AVDECC_FROM_APS</td>
<td>An AVDECC PDU is transferred from the APS.</td>
<td>C.5.1.6</td>
</tr>
<tr>
<td>06 16</td>
<td>AVDECC_FROM_APC</td>
<td>An AVDECC PDU is transferred from the APC.</td>
<td>C.5.1.7</td>
</tr>
<tr>
<td>07-ff</td>
<td>RESERVED</td>
<td>Reserved for future use.</td>
<td></td>
</tr>
<tr>
<td>ff 16</td>
<td>VENDOR</td>
<td>A vendor specific message.</td>
<td>C.5.1.8</td>
</tr>
</tbody>
</table>
C.4.3. payload_length field

The payload_length field is set to the length of the payload field in octets. The payload_length field can be from zero (0) to 1490 (inclusive).

C.4.4. address field

The address field is a MAC-48 or EUI-48 value. The contents of the address field is dependent on the message_type field as defined in Table C.4:

<table>
<thead>
<tr>
<th>message type</th>
<th>address field contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY_GUID_REQUEST</td>
<td>The MAC address of the primary network port of the APC.</td>
</tr>
<tr>
<td>ENTITY_GUID_RESPONSE</td>
<td>The MAC address of the primary network port of the APC.</td>
</tr>
<tr>
<td>LINK_UP</td>
<td>The network port's MAC address.</td>
</tr>
<tr>
<td>LINK_DOWN</td>
<td>The network port's MAC address.</td>
</tr>
<tr>
<td>AVDECC_FROM_APS</td>
<td>The ethernet frame's source address.</td>
</tr>
<tr>
<td>AVDECC_FROM_APC</td>
<td>The ethernet frame's destination address.</td>
</tr>
<tr>
<td>VENDOR</td>
<td>A vendor specific EUI-48 message type value.</td>
</tr>
</tbody>
</table>

All other message types have the address field reserved and set to zero (0).

C.4.5. payload field

The payload field can be from zero (0) to 1490 (inclusive) octets in length. The length of the payload field in octets shall be represented by the payload_length field.

The contents of the payload field is dependent on the message_type as defined in Table C.5:

<table>
<thead>
<tr>
<th>message type</th>
<th>payload field contents</th>
<th>payload length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY_GUID_REQUEST</td>
<td>The requested Entity GUID, or zero (0).</td>
<td>Eight (8) octets.</td>
</tr>
<tr>
<td>ENTITY_GUID_RESPONSE</td>
<td>The assigned Entity GUID</td>
<td>Eight (8) octets.</td>
</tr>
<tr>
<td>LINK_UP</td>
<td>none</td>
<td>zero (0) octets.</td>
</tr>
<tr>
<td>LINK_DOWN</td>
<td>none</td>
<td>zero (0) octets.</td>
</tr>
<tr>
<td>AVDECC_FROM_APS</td>
<td>The AVDECC message payload</td>
<td>zero (0) to 536 (inclusive).</td>
</tr>
<tr>
<td>AVDECC_FROM_APC</td>
<td>The AVDECC message payload</td>
<td>zero (0) to 536 (inclusive).</td>
</tr>
<tr>
<td>VENDOR</td>
<td>A vendor specific message payload</td>
<td>zero (0) to 1490 (inclusive).</td>
</tr>
</tbody>
</table>
C.5. Protocol Description

C.5.1. Messages

C.5.1.1. NOP

The NOP message is sent by both the APC and the APS after a period of inactivity.

The NOP message has:

- The version field set to zero (0).
- The message_type field set to NOP.
- The payload_length field set to zero (0).
- The address field set to zero (0).

C.5.1.2. ENTITY_GUID_REQUEST

The ENTITY_GUID_REQUEST message is sent by an APC to an APS to request that the APS allocate a GUID for the APC to use as an Entity GUID and respond with a ENTITY_GUID_RESPONSE message.

The ENTITY_GUID_REQUEST message has:

- The version field set to zero (0).
- The message_type field set to ENTITY_GUID_REQUEST.
- The payload_length field set to eight (8).
- The address field set to the primary MAC address of the APC.
- The payload field is set to the APC's current GUID, or zero (0) if it has none.

C.5.1.3. ENTITY_GUID_RESPONSE

The ENTITY_GUID_RESPONSE message is sent by an APS to an APC in response to an ENTITY_GUID_REQUEST message.

The ENTITY_GUID_RESPONSE message has:

- The version field set to zero (0).
- The message_type field set to ENTITY_GUID_RESPONSE.
- The payload_length field set to eight (8).
- The address field set to the primary MAC address of the APC.
- The payload field set to the generated Entity GUID that the APC shall use.

C.5.1.4. LINK_UP

The LINK_UP message is sent by an APS to an APC to notify the APC that the network port link is connected.

The LINK_UP message has:
• The version field set to zero (0).
• The message_type field set to LINK_UP.
• The payload_length field set to zero (0).
• The address field set to the MAC address of the network port.

C.5.1.5. LINK_DOWN
The LINK_DOWN message is sent by an APS to an APC to notify the APC that the network port link is disconnected.

The LINK_DOWN message has:
• The version field set to zero (0).
• The message_type field set to LINK_DOWN.
• The payload_length field set to zero (0).
• The address field set to the MAC address of the network port.

C.5.1.6. AVDECC_FROM_APS
The AVDECC_FROM_APS message is sent by an APS to an APC to transport a layer 2 AVDECC message to the APC.

The AVDECC_FROM_APS message has:
• The version field set to zero (0).
• The message_type field set to AVDECC_FROM_APS.
• The payload_length field set to the IEEE Std. 1722 PDU length in octets.
• The address field set to the PDU’s original source address.
• The payload field set to the IEEE Std. 1722 PDU.

C.5.1.7. AVDECC_FROM_APC
The AVDECC_FROM_APC message is sent by an APC to an APS to transport a layer 2 AVDECC message to the APS.

The AVDECC_FROM_APC message has:
• The version field set to zero (0).
• The message_type field set to AVDECC_FROM_APC.
• The payload_length field set to the IEEE Std. 1722 PDU length in octets.
• The address field set to the destination MAC address.
• The payload field set to the IEEE Std. 1722 PDU.

C.5.1.8. VENDOR
The VENDOR message is sent by an APS or an APC to transport an APS vendor specific message.
The VENDOR message has:

- The version field set to zero (0).
- The message_type field set to VENDOR.
- The payload_length field set to the length of the payload field.
- The address field set to an EUI-48 specifying the vendor specific message.
- The payload field set to the appropriate payload for the vendor specific message.

C.5.2. APS State Machine

C.5.2.1. State machine variables

C.5.2.1.1. a

The a variable is set to the address field of the message received from the APC.

C.5.2.1.2. apcMsg

The apcMsg variable is a Boolean which is set to TRUE if and only if the out variable contains an AVDECC PDU from the APC.

C.5.2.1.3. assignGuidRequest

The assignGuidRequest variable is a Boolean which is set to TRUE when the APS receives a ENTITY_GUID_REQUEST message.

C.5.2.1.4. currentTime

The currentTime variable is an unsigned integer which increases by one every second.

C.5.2.1.5. finished

The finished variable is a Boolean which is set to TRUE when the APS is required to stop operation.

C.5.2.1.6. guid

The guid variable is set to the GUID payload of the received ENTITY_GUID_REQUEST message.

C.5.2.1.7. in

The in variable contains the AVDECC PDU and source address received from a layer 2 network port.
C.5.2.1.8. incomingTcpClosed

The incomingTcpClosed variable is a Boolean which is set to TRUE when the incoming socket from the APC is closed.

C.5.2.1.9. linkStatus

The linkStatus variable is a Boolean which is set to TRUE when the APS's network port has an active link.

C.5.2.1.10. linkStatusChanged

The linkStatusChanged variable is a Boolean which is set to TRUE when the APS's network port link status changed.

C.5.2.1.11. L2Msg

The L2Msg variable is a Boolean which is set to TRUE when the in variable contains an AVDECC PDU from the layer 2 network.

C.5.2.1.12. nopTimeout

The nopTimeout variable is an unsigned integer that represents the currentTime value when a NOP message is to be sent to the APC.

C.5.2.1.13. out

The out variable contains the AVDECC PDU and destination address received from the APC in the AVDECC_FROM_APC message.

C.5.2.1.14. requestValid

The requestValid variable is set to the appropriate HTTP response code for the HTTP request, as defined in RFC 2616 Section 6.1.1.

C.5.2.1.15. tcpConnected

The tcpConnected variable is a Boolean which is set to TRUE when an APC connects to the APS.

C.5.2.2. State machine functions

C.5.2.2.1. closeTcpConnection()

The closeTcpConnection() function closes the TCP connection with the APC and sets the incomingTcpClosed variable to FALSE.
C.5.2.2.2. initialize()

The initialize() function sets:

- `apcMsg` to FALSE
- `assignGuidRequest` to FALSE
- `currentTime` to zero (0)
- `finished` to FALSE
- `L2Msg` to FALSE
- `linkStatus` to FALSE
- `nopTimeout` to zero (0)
- `tcpConnected` to FALSE

C.5.2.2.3. sendAvdeccToApc(in)

The sendAvdeccToApc(in) function forms the AVDECC_FROM_APS message with the address and payload from the `in` parameter and sends it over the TCP socket to the APC. This function also sets the `L2Msg` variable to FALSE.

C.5.2.2.4. sendAvdeccToL2(out)

The sendAvdeccToL2(out) function takes the AVDECC PDU and destination address in the `out` parameter and sends it to the layer 2 network port. This function also sets the `apcMsg` variable to FALSE.

C.5.2.2.5. sendGuidAssignment(a,guid)

The sendGuidAssignment() function validates the MAC-48 address in the `a` parameter and the `guid` parameter which are from the ENTITY_GUID_REQUEST message. If the `guid` is known to not be unique then an appropriate GUID assignment is calculated.

The `a` parameter is stored in the `address` field, and the appropriate assigned GUID is stored in the `payload` field of the ENTITY_GUID_RESPONSE message.

This function also sets the `assignGuidRequest` variable to FALSE.

C.5.2.2.6. sendHttpResponse(httpCode)

The sendHttpResponse(httpCode) function forms and sends an HTTP response as defined in RFC 2616 Section 6 to the APC with the `httpCode` as the HTTP Status-Code.

The minimum required HTTP response for a successful request (in Augmented BNF form) is:

```
"HTTP/1.1" SP "200" SP "OK" CRLF CRLF
```

The minimum required HTTP response for an unsuccessful request is:
"HTTP/1.1" SP Status-Code SP Reason-Phrase CRLF CRLF

Where Status-Code and Reason-Phrase are appropriate values from RFC 2616 Section 6.1.1.

C.5.2.2.7. sendLinkStatus(linkStatus)

The sendLinkStatus(linkStatus) function forms and sends a LINK_UP message to the APC if linkStatus is TRUE, or a LINK_DOWN message to the APC if linkStatus is FALSE. This function also sets the linkStatusChanged flag to FALSE.

C.5.2.2.8. sendNopToApc()

The sendNopToApc() function sends a NOP message to the APC.

C.5.2.2.9. validateHttpRequest()

The validateHttpRequest() function parses and validates the incoming HTTP request header from the APC. The return value of the validateHttpRequest() function is an HTTP Status code from RFC 2616 Section 6.1.
C.5.2.3. State machine diagram

Figure C.2. APP APS state machine
C.5.3. APC State Machine

C.5.3.1. State machine variables

C.5.3.1.1. addr

The *addr* variable is set to the TCP address, TCP port, and path of the APS.

C.5.3.1.2. apcMsg

The *apcMsg* variable is set to the contents of the AVDECC_FROM_APC message that the APC is sending to the APS.

C.5.3.1.3. apcMsgOut

The *apcMsgOut* variable is set to TRUE when the APC Entity has an AVDECC message to send to the APS.

C.5.3.1.4. apsMsg

The *apsMsg* variable is set to the contents of the AVDECC_FROM_APS message received from the APS.

C.5.3.1.5. apsMsgIn

The *apsMsgIn* variable is set to TRUE when an AVDECC_FROM_APS message is received from the APS.

C.5.3.1.6. currentTime

The *currentTime* variable is an unsigned integer which increases by one every second.

C.5.3.1.7. finished

The *finished* variable is set to TRUE when it is requested for the APC to shut down processing.

C.5.3.1.8. guid

The *guid* variable is set to the APC Entity's requested preferred Entity GUID, or zero (0) if there is none.

C.5.3.1.9. guidAssigned

The *guidAssigned* variable is set to TRUE when a ENTITY_GUID_RESPONSE message is received from the APS.
C.5.3.1.10. incomingTcpClosed

The `incomingTcpClosed` variable is set to TRUE when the socket connection to the APS is closed.

C.5.3.1.11. linkMsg

The `linkMsg` variable is set to the contents of the LINK_UP or LINK_DOWN message received from the APS.

C.5.3.1.12. linkStatusMsg

The `linkStatusMsg` variable is set to TRUE when a LINK_UP or LINK_DOWN message is received from the APS.

C.5.3.1.13. newGuid

The `newGuid` variable is set to the assigned Entity GUID from the ENTITY_GUID_RESPONSE message payload received from the APS.

C.5.3.1.14. nopTimeout

The `nopTimeout` variable is an unsigned integer that represents the `currentTime` value when a NOP message is to be sent to the APS.

C.5.3.1.15. primaryMac

The `primaryMac` variable is set to the primary MAC address of the APC.

C.5.3.1.16. responseValid

The `responseValid` variable is TRUE when the HTTP response from the APS indicates success.

C.5.3.1.17. tcpConnected

The `tcpConnected` variable is set to TRUE when a socket connection is made to the APS.

C.5.3.2. State machine functions

C.5.3.2.1. closeTcpConnection()

The closeTcpConnection() function closes the TCP connection with the APS.

C.5.3.2.2. connectToProxy(addr)

The connectToProxy() function initiates a TCP connection with the APS with the address `addr`. 
C.5.3.2.3. getHttpResponse()

The getHttpResponse() function receives and parses an HTTP response header as defined in RFC 2616 from the APS. It returns TRUE if the response indicates a successful HTTP CONNECT.

C.5.3.2.4. initialize()

The initialize() function sets:

- `apcMsgOut` to FALSE
- `apsMsgIn` to FALSE
- `finished` to FALSE
- `guidAssigned` to FALSE
- `incomingTcpClosed` to FALSE
- `linkStatusMsg` to FALSE
- `responseValid` to FALSE
- `tcpConnected` to FALSE

C.5.3.2.5. notifyLinkStatus(linkMsg)

The notifyLinkStatus(linkMsg) function notifies the APC Entity that the APS's layer 2 network link status has changed. The `linkMsg` parameter contains a LINK_UP APP message or a LINK_DOWN APP message. The notifyLinkStatus(linkMsg) function also sets the `linkStatusMsg` variable to FALSE.

C.5.3.2.6. processMsg(apsMsg)

The processMsg(apsMsg) function notifies that APC Entity that the APS's layer 2 network link had received an AVDECC message. The `apsMsg` parameter contains a AVDECC_FROM_APS APP message. The processMsg(apsMsg) function also sets the `apsMsgIn` variable to FALSE.

C.5.3.2.7. sendGuidRequest(primaryMac, guid)

The sendGuidRequest(primaryMac, guid) function sends a ENTITY_GUID_REQUEST APP message to the APS. The `primaryMac` parameter is placed in the address field of the APP message and the `guid` parameter is placed in the payload field of the APP message.

C.5.3.2.8. sendHttpRequest(addr)

The sendHttpRequest(addr) function sends the appropriate HTTP CONNECT method request to the APS as defined in RFC 2616 Section 5. The `addr` parameter contains the hostname, port, and path used to form the HTTP request header.

The minimum required HTTP request (in Augmented BNF form) is:

```
"CONNECT" SP path SP "HTTP/1.1" CRLF
```
"Host:" SP host ":" port CRLF CRLF

Where path is the path value from the addr parameter, host is the host or TCP address value from the addr parameter, and port is the TCP port from the addr parameter.

C.5.3.2.9. sendMsgToAps(apcMsg)

The sendMsgToAps(apcMsg) function sends the apcMsg parameter, which is an AVDECC message encapsulated in an AVDECC_FROM_APC APP message, to the APS. The sendMsgToAps(apcMsg) function also sets the apcMsgOut variable to FALSE.

C.5.3.2.10. sendNopToAps()

The sendNopToAps() function sends a NOP APP message to the APS.
C.5.3.3. State machine diagram

Figure C.3. APP APC state machine