Automotive Ethernet AVB Functional and Interoperability Specification

Revision 1.4
12 May 2015

Authors:
Gordon Bechtel, Ben Gale, Max Kicherer, Dave Olsen

All members of the AVnu Automotive Technical Working Group contributed to this document

Table of Contents

Revision History

Revision	*Author*	*Date*	*Description*
0.1	Gordon Bechtel, Ben Gale, Max Kicherer, Dave Olsen and all members of Automotive CDS	2014-03-11	Initial release of full document.
0.2	Gordon Bechtel	2014-03-12	Added editorial note indicating direction for resolution of GM failure in section 9.1.1.
0.3	Gordon Bechtel, Ben Gale, Max Kicherer, Dave Olsen	2014-04-30	Updated document to include all comments received from and discussed with members of the Automotive CDS working group since release 0.2.
1.0	Gordon Bechtel, Ben Gale, Dave Olsen	2014-06-24	Updated to include all comments from 0.3 release and subsequent discussions. Please see the comment resolution spreadsheet for details.[1]
1.1	Gordon Bechtel	2014-07-29	Updated to include all comments on the 1.0 ballot and discussions within the automotive CDS. Please see comment spreadsheet for details.[2]
1.2	Gordon Bechtel, Ben Gale, Dave Olsen	2014-11-03	Updated to include all comments on the V1.1 version discussed within the Automotive CDS. Please see the comment matrix for details.[3]
1.3	Gordon Bechtel	2014-11-24	Updated to include all comments on the V1.2 version discussed within the Automotive CDS. Please see the comment matrix for details.[3]
1.4	Gordon Bechtel	2015-05-12	Updated to include all comments on the V1.3 version discussed within the Automotive CDS during the process of review and PICS generation. Please see the comment matrix for details.[4]

[1] https://groups.avnu.org/wg/AVnu_Technical/document/download/3024 (Formerly: http://members.avnu.org/apps/org/workgroup/technical/download.php/3659/avnuAutomotiveSpecInformalReviewCommentsOnV0_3-collated-2014-06-24.xlsx)

[2] https://groups.avnu.org/wg/AVnu_Technical/document/download/3071 (Formerly: http://members.avnu.org/apps/org/workgroup/technical/download.php/3701/avnuAutomotiveSpecInformalReviewCommentsOnV1-STC.xls)

[3] https://groups.avnu.org/wg/AVnu_Technical/document/3445.

[4] https://groups.avnu.org/wg/AVnu_Technical/document/3474.

1 Introduction

This document is AVnu's interoperability and functional specification for automotive Ethernet AVB devices. The document's primary goal is to provide a baseline specification of Ethernet AVB functionality on automotive AVB devices. The specification has several uses, three of which are key. First, vendors can use this specification to guide their automotive AVB product development. Second, AVnu can use it to develop an effective certification process. Third, and most importantly, automotive OEMs can depend on the specification to provide a foundation for AVB functionality in their automobiles. OEMs can be assured that a device certified to be compliant with this specification would provide baseline AVB functionality, would not require extensive testing on their part, and would thereby allow them the time to focus on other functionality that helps differentiate their final product and enable success in the automotive marketplace.

This specification focuses on establishing interoperability between devices at those levels of the communication stack where the devices share functional compatibility. For instance, it's obvious that a rear-view camera module won't interwork directly with an audio amplifier, simply because these two devices handle different types of media. However, both of these devices do have AVB functionality. Therefore, it is also true that they can interact on more fundamental levels, such as sharing a common wall clock via gPTP and adhering to the traffic shaping rules of FQTSS. This is what is meant by automotive baseline functionality: The functionality required for interoperability at baseline levels of automotive devices' communications stacks.

Common functionality that all automotive Ethernet-AVB devices need in order to operate in the automotive environment.

During its development, this specification has come to be referred to as the AVB "automotive profile." The focus of this version of the automotive profile is infotainment systems and basic rear-view, side-view, and front-view cameras. Devices involved with engine control, advanced driver assistance or safety modules are part of AVnu's charter – since they utilize Time Sensitive Networking (TSN) standards – and will be considered in future versions of this document. Critical features for the infotainment and view camera use case are:

- A common clock for all devices based on the 802.1AS standard, commonly known as generalized Precision Time Protocol (gPTP)
- Guaranteed bandwidth for time-critical streams
- Low-latency and synchronized transmission of audio and video
- Distribution of a common media clock
- Fast AED startup
- Operation on links with speeds at or above 100Mbps

In an effort to deliver these features as efficiently as possible, the automotive profile takes consideration of the fact that the automotive OEM has complete control over the configuration, topology and engineering of the network. The automotive profile also recognizes that the network's structure won't change with time. It is static and largely pre-configured. These facts have several important implications:

- The Ethernet bridge knows which ports are connected to time-aware devices and which are not

- The Ethernet bridge knows the location of the gPTP grandmaster
- The network's AVB media streams and their parameters are fixed and pre-configured
- AVB traffic classes can be optimized for automotive use cases

Using these facts about the automotive environment, this document specifies a concise set of requirements for delivering the features necessary for the infotainment and view camera use cases.

Regarding the lifecycle of this document, this version is the first of many anticipated releases for the automotive AVB specification. It has been driven by two factors. The first is the aforementioned goal of providing a useful baseline level of functionality. The second is to provide this baseline in a timely fashion. Early on in the process, the group recognized that time was "of-the-essence" and that a specification was needed quickly to provide a seed for the automotive AVB market. Because of this second goal, the group decided to limit the breadth of use cases to those considered in this version.

There will be future versions of this specification that cover a variety of topics not addressed in the current version. One group of future topics has to do with the operation and effectiveness of the network. These include topics like security and network management. Another group of future topics includes the super low-latency and redundant transmission capabilities so critical to engine control, safety and driver assistance systems. These topics are currently in active discussion in several IEEE 802.1 working groups and will soon require support from AVnu to develop an appropriate certification profile. Future versions of this specification will supply this necessary and important profile.[1]

2 Glossary

Term	***Meaning***
AAF	AVTP Audio Format. [AVTPa]
AED	Automotive Ethernet Device. An automotive device with Ethernet AVB functionality.
AED-A	Automotive Ethernet Audio Device. An AED that provides support for transmission and/or receipt of uncompressed audio over an Ethernet AVB network. Unless specifically specified otherwise, the term AED-A in this document refers to audio devices that use the formats described in section 7.1.
AED-B	Automotive Ethernet Bridge Device.
AED-C	Automotive Ethernet Clock Device. An AED that provides support for transmission and/or receipt of a media clock reference over an Ethernet AVB network.
AED-E	Automotive Ethernet Endstation Device. AED-E refers to any device with endstation functionality including talker functionality, listener functionality, or a combination thereof.
Time-Critical AED-E	An AED Endstation designated by the system integrator to provide time-critical services that require fast startup.
Non Time-Critical AED-E	An AED Endstation that is not designated as time-critical by the system integrator.
AED-L	Automotive Ethernet Listener. Any number of AED-Ls may be contained in an AED-A or an AED-V.
AED-T	Automotive Ethernet Talker. Any number of AED-Ts may be contained in an AED-A or an AED-V.
AED-V	Automotive Ethernet Video Device. An AED that provides support for transmission and/or receipt of compressed video over an Ethernet AVB network. Unless specifically specified otherwise, the term AED-V in this document refers to video devices that use the formats described in section 7.2.
AED-GM	Automotive Ethernet Grand Master
AVB	Audio Video Bridging
AVDECC	Audio Video Discovery, Enumeration, Connection and Control protocol for AVTP devices
AVTP	Audio Video Transport Protocol. [AVTP][AVTPa]
BMCA	Best Master Clock Algorithm
E-AVB	Ethernet AVB

FQTSS	Forwarding and Queueing for Time Sensitive Streams. Refers to section 34 of IEEE 802.1Q.
gPTP	Generalized Precision Time Protocol
HDCP	High-bandwidth Digital Content Protection
HDCP-IIA	HDCP Interface Independent Adaptation
Hybrid Device	A device that is a combination of AED-B (bridge) and an AED-E (endstation).
MTT	Maximum Transit Time
OAM	Operations, Administration and Management
OEM	Original Equipment Manufacturer. In the automotive context, OEM typically refers to the manufacturer of the automobile.
UDS	Unified Diagnostic Service

3 References

Name	*Reference*
802.1Q	IEEE 802.1Q-2011, "Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks," 31Aug2011.
AltClasses	Dave Olsen, Robert Boatright, "AVnu Automotive Alternate SR Classes," AVnu Technical Note, Revision 1.3, November 13, 2012. https://groups.avnu.org/wg/AVnu_Technical/document/download/2222 (Formerly: http://members.avnu.org/apps/org/workgroup/technical/download.php/3080/Alternate%20Automotive%20SR%20Classes%201.3.pdf)
AVDECC	IEEE 1722.1-2013, "IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722 Based Devices," 23 August 2013.
AVTP	IEEE P1722-2011, "IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks," May 2011.
AVTPa	IEEE P1722a-D12, "IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks," Draft 12, February 9, 2015.
gPTP	IEEE Std. 802.1AS-2011 "IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks"
gPTP-Cor	IEEE Std. 802.1AS-2011/Cor 1-2013 "IEEE Standard for Local and metropolitan area networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks—Corrigendum 1: Technical and Editorial Corrections"
HDCP-IIA	High-bandwidth Digital Content Protection System – Interface Independent Adaptation. http://www.digital-cp.com/files/static_page_files/6FEA6756-1A4B-B294-D0494084C37A637F/HDCP%20Interface%20Independent%20Adaptation%20Specification%20Rev2_2_FINAL.pdf
802.1BA	IEEE Std 802.1BA- 2011, "Audio Video Bridging (AVB) Systems," 30 Sept 2011.

4 Scope

The scope of this document is to provide a baseline specification of Ethernet AVB functionality on automotive AVB devices as a foundation for an effective certification process. The focus of the document is infotainment systems and basic view cameras. Devices involved with engine control, advanced driver assistance or other functionality are not in the scope of this version of the document.

The scope of this specification is at the AVTP service layer [AVTP, AVTPa] and below. This includes the Ethernet PHY, MAC and Bridge, AVB configuration and behavior, and AVTP formats and media clock. Specifically this specification does not cover the behavior of applications built on top of this service layer, or the broader system that contains the AVTP service.

Since this specification provides a baseline set of requirements for certification testing it is necessarily limited to key requirements for the targeted use cases. This limitation should not be interpreted as a restriction on the overall functionality of any particular device. Vendors and OEMs can and are encouraged to implement functionality beyond that specified here.

This specification defines three device types: endpoint, bridge and hybrid. An endpoint is a device that connects at the edge of the Ethernet network and initiates and/or terminates AVB streams. A bridge is within the Ethernet network and connects endstations and other bridges together to create the network. A hybrid device is a device that contains both an endstation and a bridge within the same “enclosure.”

The challenge with the hybrid device is separating the behavior of the endstation from the bridge for the purposes of testing. The Ethernet link between these on a hybrid device is often a PHY-less and connector-less PCB trace that cannot easily be monitored during testing. In such cases, this specification recommends that the hybrid device be tested into 2 different modes, with a different test focus for each mode:

1. ***Endstation Mode***: For endstation testing, the vendor designates a single external-facing switch port as the endstation connection. All Ethernet packets to/from the endstation must pass through this port, including test status messages. When testing the device, there will be some observable differences relative to testing a pure endstation, including:

    - Some link-level operation on the connection, such as Auto negotiation (see section 5.1) and gPTP Pdelay (see section 6) will not be directly visible. Instead their effects will at best be indirectly visible in the packets sent by the bridge (e.g. in gPTP Sync packet corrections)

    - External visibility of endstation packets is dependent upon the bridge. If the endstation sends packets before the bridge is ready then they won't be seen. This has a particular impact upon the status messages sent during initialization (see section 5.3). The hybrid device maker should therefore ensure that the bridge is at least "Ethernet Ready" before sending these messages. Internal detection of this state is left to device implementation.

    - The bridge will add some delay to packets sent and received by the endstation. This will be barely perceptible for data packets, but could be visible for certain gPTP protocol packets such as Sync and Signal, which are consumed, processed and then re-originated by the bridge. For instance, if the endstation is the GM then Sync packets coming out will experience a delay according to the bridge's local state machines. Therefore for start-up measurement, the delay for 1 bridge (see Table 6 in section 5.5) should be added to the "GM at AVB Sync" timeline in Table-5 for a hybrid device. Generally, this delay should be appropriately accounted for in all event time measurements.

- If the endstation is the GM, then the Bridge will have already corrected the observed Sync packets.
- gPTP protocol packets will have the source address of the bridge, not the endstation.

2. ***Bridge Mode***: To test the bridge, the vendor shall provide a mechanism for disabling the internal port, and testing shall use only the bridge's externally facing ports.

For the purposes of Exceptions and Diagnostics, it is left to implementation choice whether the Hybrid device reports itself as one or two AED's. However the information content obtainable from the hybrid device shall be the superset of both the endstation and bridge requirements described in this specification.

5 Network and Device Startup

5.1 Ethernet Interface Configuration

An AED shall provide at least one Ethernet interface. Each Ethernet interface shall operate at either 100 or 1000 Mbps.

Auto Negotiation on the Ethernet interface shall not be used. The vendor of each device shall either

- Provide a mechanism for configuring its Ethernet interface(s) into master/slave mode, or
- Provide documentation on the device's master/slave mode, if it is preconfigured.

5.2 States and Events

Table 1 lists the key states of AVB readiness during startup of an AED endstation. These states and their definitions are used throughout this section.

State Name	***Description***	***Enter State when…***	***Observable Action when Entering State***
Ethernet Ready	Able to receive and transmit Ethernet packets. All PHY and MAC components are up and running, however the related software necessary for full AVB operation may not yet be running.	Instant that device determines it is Ethernet Ready.	Send "Ethernet Ready" status message according to requirements in Table 10 (AED-T's) and Table 11 (AED-L's). Note: In the event that "Ethernet Ready" is reached before the software necessary to transmit the status message is fully operational, AED's shall transmit the status message as early as they possibly can after Ethernet Ready.
AVB Sync	gPTP module is running and the device has sync with the 802.1AS master clock.	Processes the second sync/follow-up message pair since start up is complete.	Send "AVB Sync" status message according to requirements in Table 10 (AED-T's) and Table 11 (AED-L's).

State Name	***Description***	***Enter State when…***	***Observable Action when Entering State***
AVB Media Ready	The device has established media clock for its particular media and is ready to forward media data to the rendering subsystem in synchrony with the media delivery clock.	An AED-T is ready to accept media data and/or media clocks (according to its design) from media generation hardware or processes and is able to transmit for all streams. An AED-L is ready to deliver media data and/or media clocks (according to its design) to media rendering hardware or processes for all streams. If an AED is both a listener and a talker, then it is AVB Media Ready when it meets both AED-T and AED-L conditions, above.	Send "AVB Media Ready" status message according to requirements in Table 10 (AED-T's) and Table 11 (AED-L's).

Table 1: AED Startup States – Non-Grandmaster Media Enabled Endstations

Table 2 lists the key states of AVB readiness during startup of an AED bridge.

State Name	***Description***	***Enter State when…***	***Observable Action when Entering State***
Ethernet Ready	Same as "Ethernet Ready" state for Endstations (Table 1)	Instant that device determines it is Ethernet Ready.	Bridge begins forwarding Ethernet frames according to 802.1Q and according to the requirements in Table 9.
AVB Sync	Same as "AVB Sync" for Endstations (Table 1)	The instant the bridge determines it is ready to process and respond to all gPTP messages across all ports.	Transmission of Sync/Follow-up messages on master ports, according to preconfigured clock tree and according to the requirements in Table 9.

Table 2: AED Startup States – Bridges

Table 3 lists the key states of AVB readiness during startup of an AED grandmaster.

State Name	***Description***	***Enter State when…***	***Observable Action when Entering State***
Ethernet Ready	Same as "Ethernet Ready" state for Endstations (Table 1)	Instant that device determines it is Ethernet Ready.	The AED-GM should generate an "Ethernet Ready" status message according to the requirements in Table 8.
AVB Sync	Same as "AVB Sync" for Endstations (Table 1)	Transmits the first sync message.	GM transmits its first Sync/Follow-up message according to the requirements in Table 8.

Table 3: AED Startup States – Grandmasters

5.3 Test Mode and Status Message Structure - Endstations

An AED-E shall have a mechanism for placing itself into "test" mode. The manufacturer of an AED-E shall make the details of the mechanism to place the device into test mode available to the AVnu test lab for use during certification testing.

While in test mode, an AED-E autonomously transmits status messages as defined below on all AVB capable interfaces. An AED-E shall:

- Transmit an ETHERNET_READY status message on each AVB enabled Ethernet interface when it enters the ETHERNET_READY state.
- Transmit an AVB_SYNC status message on each AVB enabled Ethernet interface when it enters the AVB_SYNC state.
- Transmit an AVB_MEDIA_READY status message for each stream handled by the AED when that stream enters the AVB_MEDIA_READY state. The status message is sent on the Ethernet interface carrying the indicated stream.

The intention of this message is that it be used only during test scenarios, and the AED shall not transmit the status message during normal operation. However the status reported by the messages and the timing of the associated events should be completely representative of normal operation. Aside from the transmission of the Test message, the behavior of the device while in Test mode should be indistinguishable from its normal operation.

Figure 1 shows the structure of the test status message. The message is based on a standard IEEE 1722.1 AECP GET_COUNTERS response message described in [AVDECC, clause 7.4.42.2]. The fields **message_timestamp**, **station_state**, and **station_state_specific_data** are specific to the Test Status message and are defined below. All other fields in the message shall be filled and interpreted according to their definition in the IEEE 1722 [AVTP] and 1722.1 [AVDECC] standards.

Note: Although the Test Status message is based on an IEEE 1722.1 message, it is sent autonomously by the AED and does not require any IEEE 1722.1 control infrastructure for its processing. The intent is that the AED sends a single status message as soon as it reaches the reported state, and that test engineers capture and process the message on the wire to collect and record the necessary conformance information.

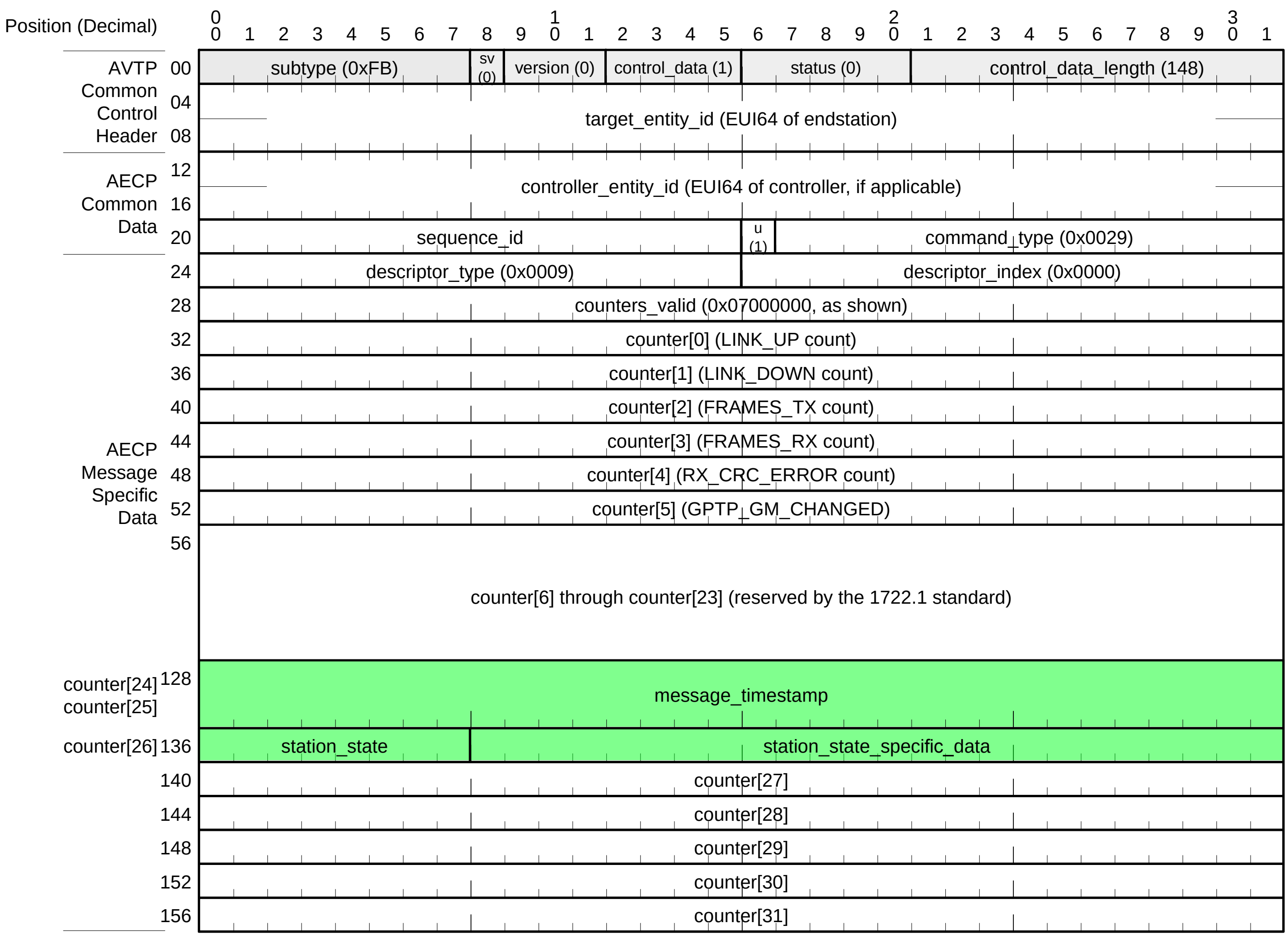

Figure 1: The Test Status Message

5.3.1 subtype field

The **subtype** field indicates the subtype for this IEEE 1722 PDU. The AED-E shall set this field to FB_{16}. As defined in [1722a], Table 5.3, this value indicates an IEEE 1722.1 AECP message.

5.3.2 sv field

The **sv** field indicates the validity of the stream ID for this IEEE 1722 PDU. The AED-E shall set this field to zero (0).

5.3.3 version field

The **version** field indicates the version for this IEEE 1722 PDU. The AED-E shall set this field to zero (0).

5.3.4 control_data field

IEEE 1722.1 uses the **control_data** field to indicate the message type of an AECP message. [AVDECC, clause 9.2.1.1.5] The AED-E shall set this field to 1_{16}, indicating that this is an AVDECC entity model response message.

5.3.5 status field

IEEE 1722.1 uses the **status** field to indicate the success/failure of the command/response AECP message pair. [AVDECC, clause 9.2.1.1.6] The AED-E shall set this field to zero (0), indicating that this is a successful response message.

5.3.6 control_data_length field

The **control_data_length** field indicates the number of octets in this message following the **target_entity_id** field. Since the status message has a fixed length, the AED-E shall set this field to 148 (94_{16}).

5.3.7 target_entity_id field

IEEE 1722.1 renames the **stream_id** field from the IEEE 1722 stream header to **target_entity_id** and uses it to indicate the 64-bit IEEE extended unique identifier (EUI-64) for the AED-E transmitting the status message. The AED-E shall set the **target_entity_id** field to its EUI-64.

The AED-E shall set its EUI-64 **target_entity_id** using the derivation from its EUI-48 MAC address, according to the EUI-48 tutorial document[5]. Use of this derivation maintains a one-to-one relationship between the device's MAC address and the EUI-64 used in the status message.

5.3.8 controller_entity_id field

IEEE 1722.1 uses the **controller_entity_id** field to indicate the EUI-64 of the destination AVDECC controller for this response message. However, the status message has no corresponding controller since it will be intercepted by test equipment. Therefore the AED-E shall set the **controller_entity_id** field to zero(0).

5.3.9 sequence_id field

The **sequence_id** field is incremented on each command sent . The AED-E shall increment the **sequence_id** on each status message it sends. The AED-E may initialize the field to any value on startup.

5.3.10 u field

The **u** field indicates whether or not the message is a solicited or unsolicited response. A value of one (1) indicates that it is unsolicited. [AVDECC, clause 7.4.39.1] The AED-E shall set the **u** field to one (1).

5.3.11 command_type field

The AED-E shall set the **command_type** field to value 0029_{16}. As specified in [AVDECC], clause 9.2.1.2 and table 7.125, this value indicates that this is a GET_COUNTERS command.

---

[5] https://standards.ieee.org/develop/regauth/tut/eui48.pdf

5.3.12 descriptor_type field

The AED-E shall set the **descriptor_type** field according to the values listed in Table 4. These values are specified in [AVDECC], clause 7.2. As the table shows, the **descriptor_type** value depends on the state of the AED. For ETHERNET_READY and AVB_SYNC states, the type of the descriptor is the AVB_INTERFACE on which the message is being transmitted. For the AVB_MEDIA_READY state, the type of the descriptor is the stream to which the message pertains.

State	***Message Source***	***Descriptor Type***	***descriptor_type value***
Ethernet Ready	AED	AVB_INTERFACE	0009_{16}
AVB Sync	AED	AVB_INTERFACE	0009_{16}
AVB Media Ready	AED-T stream entity	STREAM_OUTPUT	0006_{16}
AVB Media Ready	AED-L stream entity	STREAM_INPUT	0005_{16}

Table 4: Status Message descriptor_type Values

5.3.13 descriptor_index field

As specified in [AVDECC], clause 7.2, the **descriptor_index** field indicates the index of the descriptor to which this message refers. When **descriptor_type** refers to an AVB_INTERFACE, the AED-E shall set **descriptor_index** to the index of the Ethernet interface on which it is sending this message. The first AVB interface has index zero (0000_{16}).

When **descriptor_type** refers to a STREAM_OUTPUT, the AED-E shall set the **descriptor_index** to the index of the stream to which the message refers. The AED vendor can index the outgoing streams as they wish, as long as the index values are unique across all outgoing streams supported by the AED.

When **descriptor_type** refers to a STREAM_INPUT, the AED-E shall set the **descriptor_index** to the index of the stream to which the message refers. The AED vendor can index the incoming streams as they wish, as long as the index values are unique across all incoming streams supported by the AED.

5.3.14 counters_valid field

As described in [AVDECC], section 7.4.42.2, the **counters_valid** field is a bit field indicating which of the 32 counters in the message contains valid data. A one (1) in a bit indicates that the counter corresponding to that bit position exists and has valid data. A zero (0) indicates that the counter is invalid. The correspondence begins with the least significant bit (bit 31) which refers to counter[0], then bit 30 which refers to counter[1], and continues monotonically up to the most significant bit (bit 0) which refers to counter[31]. For example, if **counters_valid** = 07000023_{16} it would indicate that counters 0 (LINK_UP), 1 (LINK_DOWN), 5 (GPTP_GM_CHANGED), 24 and 25 (message_timestamp), and 26 (station_state and station_state_specific_data) are all valid counters.

The AED-T shall set the **counters_valid** field to match the valid counters contained in each status message it sends. In the specific case of the message_timestamp, which is carried in two counters (24 and 25), a value of 11_2

indicates that the message_timestamp field is valid. Any other value for these two bits (00_2, 01_2, or 10_2) indicate that the message_timestamp field is not valid.

5.3.15 counter[0] field (Link_Up count)

The **counter[0]** field indicates the count of link-up events since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to the link-up count at the time it creates the status message. If an AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

5.3.16 counter[1] field (Link_Down count)

The **counter[1]** field indicates the count of link-down events since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to the link-down count at the time it creates the status message. If the AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

5.3.17 counter[2] field (Frames_Tx count)

The **counter[2]** field indicates the count of transmitted Ethernet frames since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to the transmitted frame count at the time it creates the status message. If the AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

5.3.18 counter[3] field (Frames_Rx count)

The **counter[3]** field indicates the count of received Ethernet frames since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to the received frame count at the time it creates the status message. If the AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

5.3.19 counter[4] field (Rx_CRC_Error count)

The **counter[4]** field indicates the count of received Ethernet frames having a CRC error since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to the received errored-frame count at the time it creates the status message. If the AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

5.3.20 counter[5] field (GPTP_GM_Changed count)

The **counter[5]** field indicates the count gPTP grandmaster changes since the AED-E's startup. It is defined in [AVDECC], section 7.4.42.2.2, Table 7.135. It is recommended but not mandatory that an AED-E supports this field. If the AED-E does support this counter, it shall set it to gPTP grandmaster change count at the time it creates the status message. If the AED-E does not support this counter, it shall set the counter to zero (00000000_{16}).

Note: Since the Best Master Clock Algorithm (BMCA) is not running in AED's conforming to this version of the AVnu automotive specification, it is expected that the **GPTP_GM_Changed** field will always by zero

(00000000_{16}). However, the specification retains the 1722.1 definition of this field since future versions of this specification may allow operation of BMCA.

5.3.21 counter[6] through counter[23] fields

The fields **counter[6]** through **counter[23]** are reserved by the 1722.1 standard [AVDECC]. The AED-E shall set these counters to zero (00000000_{16}).

5.3.22 counter[24] and counter[25] fields (message_timestamp)

The **message_timestamp** field occupies counters 24 and 25 of the base AECP message. The **message_timestamp** field indicates the gPTP time at which the device entered the indicated state. The **message_timestamp** field is in nanoseconds and is constructed from both the gPTP seconds and nanoseconds by the following formula:

$$\text{timestamp} = (AS_{sec} \times 10^9 + AS_{ns}) \bmod 2^{64}$$

where AS_{sec} is the gPTP seconds field, and AS_{ns} is the gPTP nanoseconds field. This value rolls over approximately every 585 years ($\frac{2^{64}}{365 \times 24 \times 60 \times 60 \times 10^9} = 584.9$), giving plenty of room for measuring AED startup time.

If the AED-E is transmitting this message to signal its entry into the "Ethernet Ready" state (as given in the definition for counter[26], octet 0), it does not yet have a valid gPTP clock. In this case, the AED-E shall set the **message_timestamp** field to zero (0000000000000000_{16}).

5.3.23 counter[26], octet 0 field (station_state field)

The **station_state** field holds the state that the device just entered. Table 5 shows the values for each state.

Value	***Name***	***Definition***
0x00	Reserved	
0x01	ETHERNET_READY	Defined in Table 1 for media enabled endstations and in Table 3 for grandmasters.
0x02	AVB_SYNC	Defined in Table 1 for media enabled endstations and in Table 3 for grandmasters.
0x03	AVB_MEDIA_READY	Defined in Table 1 for media enabled endstations.
0x04–0x1F	Reserved	Reserved for future common states.
0x20–0xEF	User Defined	Available for use by specific products.

Value	*Name*	*Definition*
0xF0–0xFF	Experimental	Experimental states. Not for production.

Table 5: Station State Values

5.3.24 counter[26], octet 1-3 field (station_state_specific_data)

The **station_state_specific_data** field holds data that is specific to each state. If not specifically defined by a state, the talker shall set **station_state_specific_data** to zero (0) and the receiver shall ignore it.

5.3.25 counter[27] through counter[31]

The fields **counter[27]** through **counter[31]** are not used in the status message. The AED-E shall set these fields to zero (0).

5.4 Addressing

The AED-E shall use the standard structure for an AVTPDU as defined in IEEE 1722. [1722] For Ethernet addressing, it shall always use the multicast address 01-1B-C5-0A-C0-00 as the destination address when transmitting the test status message. This address has been assigned from AVnu's block of 4096 OUIs. AVnu's base identifier is an OUI–36, with the value 00–1B–C5–0A–C.

5.5 Startup Timing

An individual AED's startup timing must operate within the context of the overall startup process for a vehicle network. Table 6 shows the worst-case startup budget for a vehicle's Ethernet network.

Event	*Worst-Case Duration*	*Worst-Case Accum. Duration*
External Wakeup Event (Note 1)	0 ms	0 ms
Wakeup issued to Ethernet network elements (Note 2)	300 ms	300 ms
GM, Bridge, AED-Es booted and at Ethernet Ready (Note 3)	500 ms	800 ms
GM at AVB Sync. (Note 3)	250 ms	1050 ms
All AED-B's and AED-E's at AVB Sync (Notes 3, 4)	61.25 ms	1111.25 ms
AED-E Talker at media-ready; Media begins transmission (Notes 3)	100 ms	1211.25 ms
AED-E Listener at media-ready; Begins presentation of media along with a media clock. (Notes 3, 5)	500 ms	1711.25 ms
AED-E listener presents media to user (Note 6)	20 ms	1731.25 ms

Table 6: Worst-Case Vehicle Network Startup Budget

Notes on Startup Budget

1. The external wakeup event is dependent on the vehicle and the device. Example wakeup events could be the turning of an ignition key or the opening of the driver's door. Prior to this event, the vehicle and device are in a "quiescent" state, as expected when the vehicle has been parked for a significant period of time.

2. A common technique to trigger operation on an automotive ECU is to send a wakeup signal through an auxiliary interface, such as a CAN port. The process from the time of turning the key to receiving this wakeup signal at the AED typically spans 200 to 300 ms. The Automotive CDS decided to budget 300 ms for wakeup.

3. This duration comes directly from the AED startup duration requirements found later in this section.

4. The duration for reaching AVB sync at bridges and endstations depends on several variables, including network topology, gPTP sync message repetition rate and individual endstation response times to gPTP sync messages. The calculation shown here assumes a two-bridge/three hop network topology. The worst case time to AVB sync occurs at the endstation the farthest from the GM, which, in such a topology requires gPTP sync messages to undergo two residence time delays while traversing the two bridges between the GM to the AED. Table 7 sums up the durations for each step on the way to AVB sync.

Step	*Delay Contributor*	*Value*	*Quantity*	*Total*
First gPTP sync/followup sent from GM. Then GM waits to send second gPTP sync/followup	Wait period options defined by 802.1AS and configured to be as short as possible	31.25 ms	1	31.25 ms
Second gPTP sync/followup sent from GM to farthest AED	Residence times at intermediate bridges	10 ms	2	20 ms
AED processes second gPTP sync/followup and establishes AVB Sync	Processing delay per requirement set in this section	10 ms	1	10 ms
Total				**61.25 ms**

Table 7: AVB Sync Duration Calculation

Note, in the case of the maximum sized AVB topology [802.1BA] of six-bridges/seven hops, the AVB Sync duration calculation would increase to 101.25 ms. The Automotive CDS felt that a 2/3 network topology would be typical for automotive networks. Of course, if the automotive network in an actual vehicle has a 6/7 topology, then an OEM should use the larger AVB Sync duration in its design.

5. Per simulation done in the 1722a working group by Jim Czekaj,[6] this interval may be a bit short. Mr. Czekaj's simulations show that the acquisition and settle time for recreating a high-quality media clock to be between 1000 and 1500 ms. However, the Automotive CDS decided to use a shorter 500 ms period as a compromise between startup time and media clock quality. A low- to medium-quality media clock is sufficient for initial audio playback.

6. Although it is out-of-scope for EAVB testing, the automotive CDS decided to budget 20 ms for actual presentation of the media to the user. This value is reasonable for presentation of a partial frame of video – as would be done for a rear-view camera – and is probably longer than the time required to present LPCM audio data.

Using this overall vehicle network startup budget as context, individual AED's shall meet the startup timing requirements shown in the following tables. In the following tables the vendor shall provide the testers with the mechanism to create the wakeup event.

---

6 http://grouper.ieee.org/groups/1722/contributions/2013/Media%20Clock%20Stabilization%20Part%20Two_one.pdf

213 An AED-GM shall meet the startup actions and durations in Table 8.

Transition Starts …	***Transition Ends…***	***Observable Action***	***Max Transition Duration***
Wakeup Event occurs	Ethernet Ready achieved	Issue startup message indicating ETHERNET_READY (optional) [7]	500 ms
Wakeup Event occurs	Ethernet Ready and AVB Sync achieved	Begins transmission of gPTP sync/followup. Issue startup message indicating AVB_SYNC (optional)	750 ms

214 **Table 8: Startup Actions and Durations for AED-GM's**

215 An AED-B shall meet the startup actions and durations in Table 9.

Transition Starts …	***Transition Ends…***	***Observable Action***	***Max Transition Duration***
Wakeup Event occurs	Ethernet Ready achieved	Observable Ethernet packet forwarding between ports on the bridge	500 ms
Wakeup Event occurs	Ethernet Ready AND AVB Sync achieved	Bridge forwards "corrected"[8] sync/followups	750 ms

216 **Table 9: Startup Actions and Durations for AED-B's**

217 A time-critical AED-T shall meet the startup actions and durations in Table 10.

[7] Generation of the Ethernet Ready status message is recommended but not required. This is to avoid the additional cost for adding the status message capability to a Grandmaster that doesn't already transmit AVTPDUs, of which the test message is a specific example. However, if the GM is part of a product that does have AVTP capability, it is strongly recommended that the GM transmit the Ethernet Ready message. Note that if the device does not support the status message, then there will be no verification of the time of the device to reach Ethernet Ready during conformance testing.

[8] "Corrected" sync/followup messages refers to sync and followup messages that have been corrected to include bridge residence time, link delay on the bridge's slave port, and rate ratio. The bridge makes its best effort in making the correction, but may have to rely on stored information immediately after boot.

Transition Starts …	***Transition Ends…***	***Observable Action***	***Max Transition Duration***
Wakeup Event occurs	Ethernet Ready achieved.	Issue startup message indicating ETHERNET_READY	500 ms
Wakeup Event occurs	Ethernet Ready AND AVB Sync achieved.	Issue startup message indicating AVB_SYNC	750 ms [9]
AVB_Sync achieved.	Media Ready achieved.	Issue startup message indicating MEDIA_READY.	100 ms

218 **Table 10: Startup Actions and Durations for Time-Critical AED-T's**

219 A time-critical AED-L shall meet the startup actions and durations in Table 11.

Transition Starts …	***Transition Ends…***	***Observable Action***	***Max Transition Duration***
Wakeup Event occurs	Ethernet Ready achieved.	Issue startup message indicating ETHERNET_READY	500 ms
Wakeup Event occurs	Ethernet Ready AND AVB Sync achieved.	Issue startup message indicating AVB_SYNC	750 ms [9]
(AVB_Sync achieved) AND (Receipt of first media AVTPDU)	Media Ready achieved.	Issue startup message indicating MEDIA_READY.	500 ms

220 **Table 11: Startup Actions and Durations for Time Critical AED-L's**

221 Vendors of non time-critical AED-E's shall supply their own startup state transition durations.

222

[9] Meeting this time is dependent upon receiving valid gPTP messages in a timely manner. For instance, a test might initialize the device while Syncs are already being sent at 31.25ms on the link, such that the device will start to process them as soon as it is ready to do so.

6 Generalized Precision Time Protocol (gPTP)

The Generalized Precision Time Protocol (gPTP) provides an accurate time base to all elements in an Ethernet network. In the automotive environment this time base can be used in both Audio/Video applications and Control applications to provide synchronization between events. The use for gPTP in an automotive audio/video system is to provide a timing reference for recovery of media clocks in listener devices and providing synchronous media delivery across multiple listener devices.

The automotive environment is unique in that it is a closed system. Every network device is known prior to startup and devices do not enter or leave the network, except in the case of failures. Because of the closed nature of the automotive network, it is possible to simplify and improve gPTP startup performance. Specifically, functions like election of a grand master and calculations of wire delays are tasks that can be optimized for a closed system.

6.1 Scope

This section covers only the differences between the plug-and-play[10] implementation and the implementation for a closed automotive environment of the gPTP standard. In both implementations, the gPTP standard provides the functional basis, and an AED shall implement all requirements of gPTP that are not discussed in this section in accordance with the relevant standards. [gPTP][gPTP-Cor]

This section proposes concepts that may require changes to the plug-and-play implementation of the gPTP state machines and other operations. Details of these changes are not specified here and left to the implementer.

All references of the form [gPTP x.x.x.x] refer to the specified clause in IEEE Std. 802.1AS-2011. [gPTP][gPTP-Cor]

6.2 gPTP Configuration

Due to the closed and static architecture of an automotive network, an OEM can optimize the startup time of the timing system by preconfiguring certain gPTP values and making some gPTP values persistent through a power cycle.

The requirements in this document are based on the assumption that there is a single AED-GM (GrandMaster) assigned within a vehicle network.

For the purposes of AVnu certification testing, the AED vendor shall provide a mechanism for configuring the values described in this section at the time they submit a device for certification testing. Certification test personnel will use this mechanism to properly configure the device for testing.

[10] "Plug-and-Play" refers to those implementations designed to work "out-of-the-box" with other AVB products and not require any engineering on the part of the user or network provider. The IEEE 802.1BA-2011 describes the overall scenario for the "plug-and-play" scenario.

6.2.1 Static gPTP Values

Static gPTP values are configured prior to system startup, stored in non-volatile storage, and are not expected to change during system operation.

6.2.1.1 Grandmaster Information and Topology

It is expected in an automotive environment to have a fixed gPTP GrandMaster (AED-GM). An OEM will often configure the automotive network to have a single, fixed AED-GM and with all other AED's operating as slaves to that grandmaster. This means that:

- The gPTP port role of all time-aware ports. The port of a standalone AED-GM shall be in master role. For all non-GM end-stations, the port should be in the slave role. For a device that contains bridge functionality (AED-B), ports will be in either the master or slave role, depending on whether or not the port points towards the AED-GM.

- AEDs do not support BMCA and the GrandMaster needs to be identified. A new variable isGM should be defined for each AED. isGM is set TRUE for the AED-GM and FALSE for all other AEDs.

6.2.1.2 asCapable [gPTP 10.2.4.1]

All AEDs that have time-aware ports on which they will receive or transmit gPTP timing information shall set the value of asCapable to TRUE for these ports when the link is up. This allows time aware ports to begin receiving or sending timing information with as little delay after startup as possible. The state returned for asCapable for these ports when link is down is an implementation choice.

6.2.1.3 initialLogPdelayReqInterval [gPTP 11.5.2.2]

The AED shall have a mechanism to preconfigure initialLogPdelayReqInterval on all time-aware ports to a value that corresponds with the desired initial Pdelay request interval for the port. See 6.2.3.2.

In the case of ports in the Master port role, the OEM or system integrator can, if they desire, set the initialLogPdelayInterval to 127 [gPTP 11.5.2.2] to disable the sending of Pdelay Requests. Note that the AED with a slave port must still adhere to [gPTP] and always respond to any Pdelay Requests it receives on that slave port.

6.2.1.4 initialLogSyncInterval [gPTP 10.2.4.4]

The AED shall have a mechanism to preconfigure initialLogSyncInterval on all time-aware ports to a value that corresponds with the desired initial Sync interval for the port. See 6.2.3.1.

6.2.1.5 operLogPdelayReqInterval

operLogPdelayReqInterval is the operational Pdelay request interval. A device moves to this value on all slave ports once the measured values have stabilized. The AED shall have a mechanism to preconfigure operLogPdelayReqInterval on all time-aware ports to a value that corresponds with the desired Pdelay request interval for the port. See 6.2.3.2.

6.2.1.6 operLogSyncInterval

operLogSyncInterval is the Sync interval that a device moves to and signals on a slave port once it has achieved synchronization. The AED shall have a mechanism to preconfigure operLogSyncInterval on all time-aware ports to a value that corresponds with the desired Sync interval for the port. See 6.2.3.1.

6.2.2 Persistent gPTP Values

Persistent values are values that an AED stores in non-volatile storage upon update and restores to the stored value on startup. Saving these values allows an AED to decrease its startup time and improve initial timing accuracy.

6.2.2.1 neighborPropDelay [gPTP 10.2.4.7]

A Pdelay calculation determines the propagation delay between nodes and stores this value in neighborPropDelay. The neighborPropDelay is used in calculating the current time. In a closed automotive network with fixed wire lengths the calculated neighborPropDelay should change little over time. An AED shall store neighborPropDelay in non-volatile memory and use the stored value during its next startup. The stored neighborPropDelay value for a port should be set in production or service and otherwise does not change. This allows the device to calculate accurate time prior running the Pdelay algorithm. As soon as possible after startup, the AED shall return to standard Pdelay message processing and begin calculation of neighborPropDelay according to standard techniques.

The gPTP GrandMaster does not use its neighborPropDelay in calculating time since it is the source of time. Therefore the AED-GM is not required to calculate or save neighborPropDelay.

It is possible that the neighborPropDelay may change due to cabling changes while an AED is in the power off state. For instance, this may occur in situations where the car has undergone repair or replacement of parts. In such a case, there may be timing inaccuracies on system startup. These inaccuracies will resolve after standard Pdelay calculations resume.

If the stored neighborPropDelay value is zero (0) then the value has never been initialized and shall be updated with the actual value once neighborPropDelay has stabilized. In normal operation -- except in the case of a wiring change -- this value should not change. If the actual value of neighborPropDelay differs from the stored value by more than 100ns then the AED shall update the stored value to the actual value.

NOTE: Raw Pdelay values tend to jitter due to clock resolutions and need to be filtered in order to obtain an accurate and stable Pdelay. The details of this filtering are left to the implementation. When this document discusses Pdelay and neighborPropDelay it is referring to the filtered/stabilized values and not raw values.

6.2.2.2 rateRatio [gPTP 10.2.2.1.7]

The rateRatio is equal to the ratio of the frequency of the grandmaster to the frequency of the LocalClock. The rateRatio is used in calculating the current time. An AED may save this value prior to power down and restore it on power up. Use of a stored rateRatio may allow the device to compute accurate time more quickly on startup than using the default initial rateRatio of unity. However as this ratio does vary with time and temperature, the additional accuracy may not be significant.

6.2.2.3 neighborRateRatio [gPTP 10.2.4.6]

The neighborRateRatio is an estimate of the ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the frequency of the LocalClock entity of this time-aware system. The neighborRateRatio is used in calculating the current time. An AED may save this value prior to power down and restore it on power up. Use of a stored neighbor RateRatio may allow the device to compute accurate time more quickly on startup than using the default initial neighborRateRatio of unity. However as this ratio does vary with time and temperature, the additional accuracy may not be significant.

6.2.3 Management Updated Values

Management updated values are values that an AED stores in volatile storage, but have default configurations available in non-volatile memory. A management program in response to specific events occurring or states being entered can update these values.

6.2.3.1 SyncInterval [gPTP, clause 10.2.4.5]

The SyncInterval defines the mean time-synchronization event message generation interval for the port. The default value defined in [gPTP] is 125 milliseconds. The SyncInterval has a direct impact on system synchronization time at startup as an AED must receive a minimum of two (2) Sync messages to obtain basic synchronization. However after initial synchronization is achieved, synchronization can be maintained with a much larger SyncInterval, with the corresponding benefits of reduced CPU load and network overhead.

To take advantage of reduced CPU load and network overhead, a gPTP slave port on an AED may request that the SyncInterval be increased through use of the gPTP Signaling message with the message interval request TLV. The initial sync interval is set with the variable initialLogSyncInterval, and the operational sync interval is set with the variable operLogSyncInterval. A master port on an AED shall process and correctly act upon a received gPTP Signaling message within the larger of the following intervals after receiving the message:

- Two (2) of the current sync message intervals,
- 250ms.

See 6.2.6 for required values for initialLogSyncInterval and operLogSyncInterval.

AED-E devices that support the gPTP signaling message and whose value of operLogSyncInterval differs from the value of initialLogSyncInterval, shall send the gPTP Signaling message to reduce the logSyncInterval no longer than sixty (60) seconds after achieving gPTP synchronization.

An AED-B may use the gPTP signaling technique to alter the SyncInterval on its Slave Port based upon the sync intervals on its master ports. In such a case the AED-B's Slave port's SyncInterval should track a value equal to the shortest SyncInterval across all the active AS master role ports on the bridge. If the the AED-B does reduce the SyncInterval on its slave port, it shall do it within 4 seconds of the event that causes the update.

If an AED port experiences a link down/up event then, on link up, the Sync Interval state machine shall revert to its initialization state. Specifically:

- On an AED-E slave port, the device shall clear any state associated with prior signaling, and the device shall await time synchronization before signaling a slower rate to the master.

- On an AED-B master port, the device shall revert SyncInterval to initialLogSyncInterval. The AED-B shall signal for an updated interval on its slave port as required.
- On an AED-B slave port, the AED-B shall expect the link partner port to revert to it's initialLogSyncInterval, and then signal the required rate on the port according to the current rate on each of its slave master ports.
- On an AED-GM master port, the device shall revert SyncInterval to initialLogSyncInterval and the device should await further signals.

6.2.3.2 pdelayReqInterval

The pdelayReqInterval defines the mean Pdelay_Req message generation interval for the port. Pdelay_Req messages are sent to determine the neighborPropDelay. In a fixed topology network this value should not change over time making it desirable to run the calculation very infrequently. The variable initialLogPdelayReqInterval controls the startup rate of pdelayReqInterval.

Once the neighborPropDelay has stabilized, the value of LogPdelayReqInterval should be set to a value that reduces network overhead and CPU load associated with computing neighborPropDelay. AED-E devices whose value of operLogPdelayInterval differs from the value of initialLogPdelayInterval, shall set their logPdelayReqInterval to operLogPdelayInterval no longer than sixty (60) seconds after achieving gPTP synchronization.

See 6.2.6 for required values for initialLogPdelayReqInterval and operLogPdelayReqInterval.

6.2.4 gPTP Signaling Message [gPTP 10.5.4]

The gPTP Signaling Message with Message interval request TLV [gPTP, clause 10.5.4.2.2] is designed to allow a port to adjust the rate of reception of messages from its link partner port. This TLV can be used in gPTP to request a change in timeSyncInterval, linkDelayInterval and announceInterval on the partner port. In the automotive profile, this message is used only to request modification of the timeSyncInterval on the partner port. AEDs with a slave role port should support transmission of this TLV. AEDs with a master role port shall support processing of this TLV on each master role port. An AED shall ignore any gPTP Signaling Messages received on a slave role port.

If an AED supports transmission of the gPTP Signaling Message on a slave role port, then, when doing so, it shall set the timeSyncInterval to the desired value (see section 6.2.3.1). The values for linkDelayInterval and announceInterval shall be set to 127, indicating that such messages should not be sent.

When processing a received gPTP Signaling Message on a master role port, an AED shall update the syncInterval for the receiving port accordingly and shall ignore values for linkDelayInterval and announceInterval.

6.2.5 FollowUp TLV's

AED's shall include the FollowUp information TLV on all FollowUp messages [gPTP, clause 11.4.4.3].

6.2.6 Required Values for gPTP

All ports shall support the full range of possible values of gPTP variables for time critical and non-time critical ports as listed in Table 12 and Table 13.

Time critical Port	Min	Max
initialLogPDelayReqInterval	1s	1s
operLogPdelayReqInterval	1s	8s
initialLogSyncInterval	31.25ms	125ms
operLogSyncInterval	125ms	1s

Table 12 - gPTP values for time critical systems

Non-time critical Port	Min	Max
initialLogPDelayReqInterval	1s	1s
operLogPdelayReqInterval	1s	8s
initialLogSyncInterval	125ms	125ms
operLogSyncInterval	125ms	1s

Table 13 - gPTP values for non-time critical systems

6.3 gPTP Operation

In the automotive environment, AED's execute gPTP operations somewhat differently than in a standard implementation. These differences are:

- *BMCA shall not execute.* Conceptually, each device should be preconfigured with the result that BMCA would have arrived at if it had been previously running and had reached a quiescent operating state. Announce messages are neither required nor expected in an automotive network since BMCA is not being executed. With no BMCA there will always be exactly one device in the network that is configured as GM and no other device will ever become the GM.

- *No verification of sourcePortIdentity.* Because there are no Announce messages in the automotive network, it is not possible for an AED slave to know the sourcePortIdentity of its link partner until the first Sync message is received. Therefore an AED Slave port shall not perform verification of the sourcePortIdentity field of Sync and Followup messages.

- The AED-GM should begin sending Sync and FollowUp messages as soon as possible after startup. (See Clause 5 Network and Device Startup.)

- syncReceiptTimeout [gPTP 10.6.3.1] behavior for non GM AED-B devices is modified as follows:

- *AED-B Sync processing during loss of Sync*: During normal operation, the AED-B will receive and process Sync messages on its Slave port. If the AED-B detects a termination of these Sync messages on its Slave port, the AED-B shall continue sending Sync messages out of its Master Ports using the most recent valid record received on the slave port as its basis. The AED-B shall use the preciseOriginTimestamp from the most recent valid record received from the GM in all FollowUp messages it sends, and it shall advance the correctionField in all Sync and FollowUp messages it sends, to reflect the time that has passed since receipt of the most recent valid record. The AED-B shall use its local timebase and any known cumulativeScaledRateOffset to make the calculations for these messages.

- *AED-B Sync recovery:* If the AED-B receives a valid Sync message on its Slave port then it shall update that record and resume normal Sync operation. The Bridge is not required to avoid, mitigate or report a phase jump in this circumstance.

- *AED-B Sync Absence:* On startup if an AED-B does not receive a Sync message on a Slave port within a configurable time, then it shall use a default time record and begin sending Sync messages with constant preciseOriginTimestamp and advancing correction on its Master role ports. The time to wait for an initial Sync message is user configurable, but shall never be greater than 20 seconds.

- *AED-B Fixed GM Information:* Regardless of the status of Sync message reception, the AED-B shall not modify gmTimeBaseIndicator, lastGMPhaseChange or scaledLastGMFreqChange fields in any Sync or FollowUp messages that it sends.

- *AED-E Time Jumps:* An AED-E Slave may see jumps in the time it receives in Sync messages due to the switches in Sync message generation between the AED-GM and the AED-B. Time jumps may also occur due to daylight savings time or other clock adjustments. The AED-E shall handle the discontinuity in time according to the exception handling procedure described in section 9.3.2.

- *AED-E Holdover:* An AED-E Slave that is not currently receiving Sync messages will continue to advance its gPTP wall clock based on its local timebase and any known cumulativeScaledRateOffset.

- *AED-E Sync recovery:* If the AED-E receives a valid Sync message on its Slave port then it shall update its time record and resume normal Sync operation. The AED-E shall handle the discontinuity in time according to the exception handling procedure described in section 9.3.2.

- Since there is a fixed clock-spanning tree, AED-B implementations may choose to only perform delay measurement in one direction. An AED-B is not required to initiate peer delay requests on its time-aware ports that are in the Master role.

- In the case where an AED-B is the gPTP Grandmaster, if the GM operation ceases then the AED-B shall stop sending Syncs on master ports. Connected AED-E devices should go into holdover, while a connected AED-B should implement the syncReceiptTimeout behavior described above.

6.4 Automotive Network Diagrams

Figure 2 shows a typical automotive Ethernet network with the port role and asCapable indicated for each port.

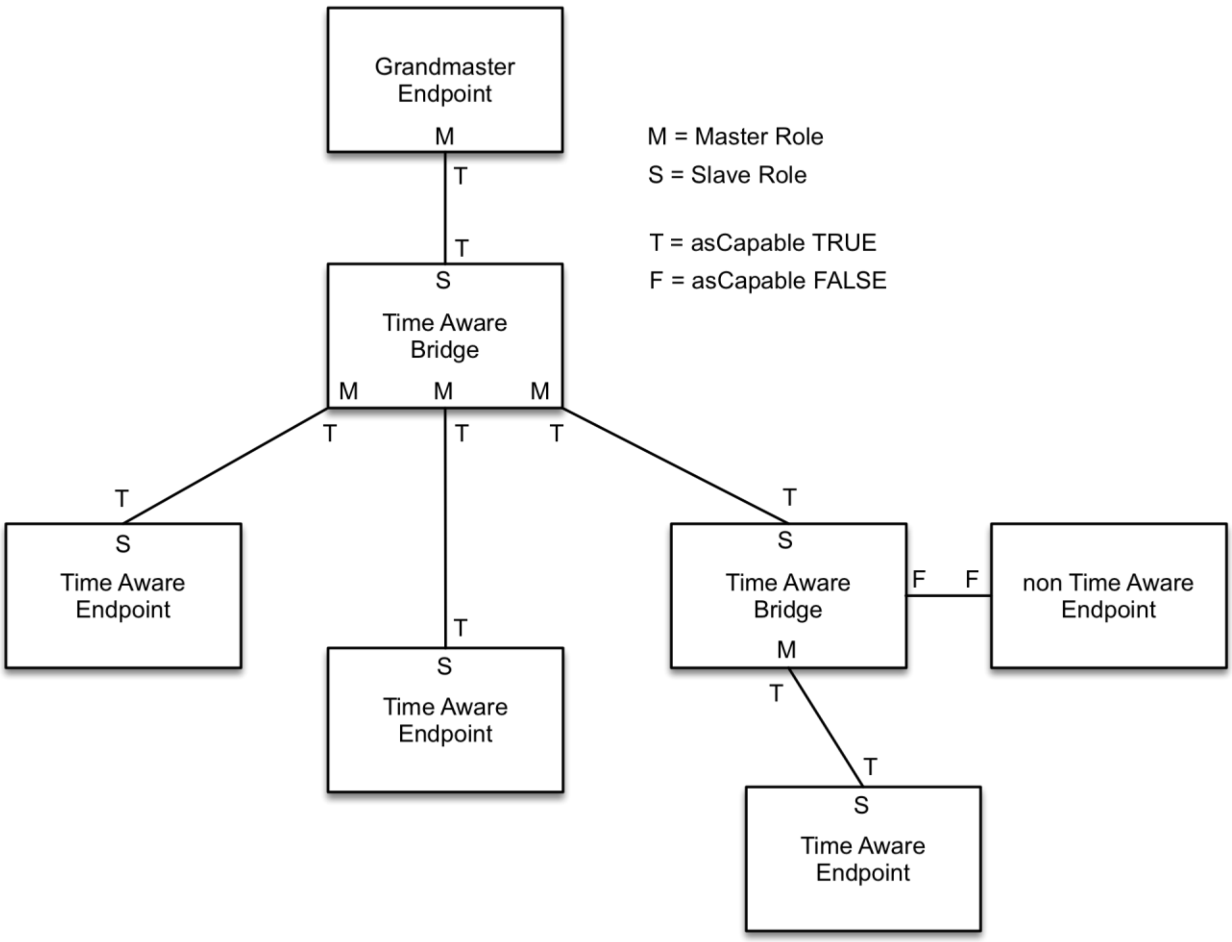

Figure 2: Typical Automotive Network

Figure 3 shows the flow of timing messages in a typical automotive network.

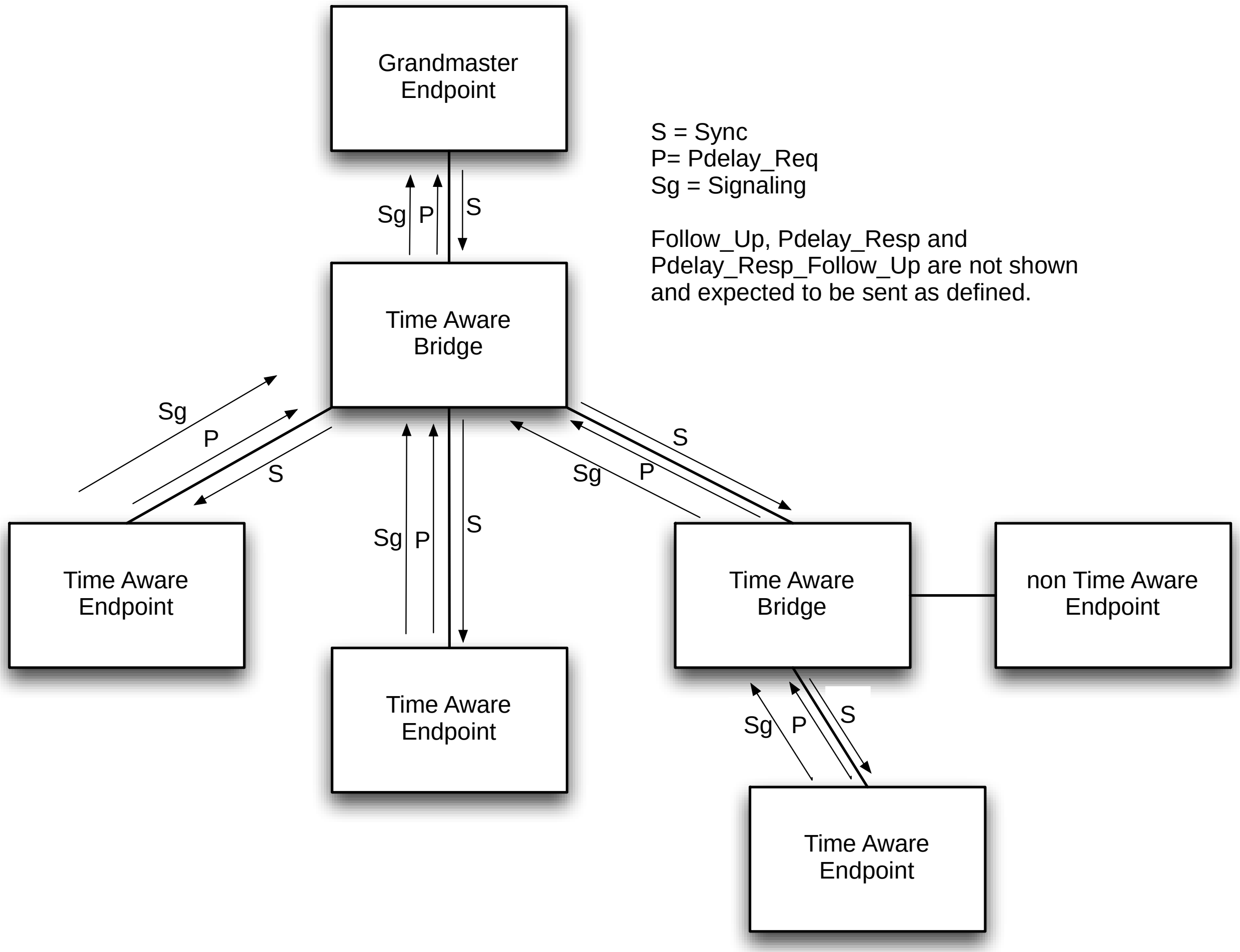

Figure 3: Flow of Timing Messages

7 Media Formats

Automotive infotainment use cases require the Ethernet-AVB network to carry a variety of media formats. They include uncompressed audio, compressed video, container formats that carry a variety of compressed media, and clock distribution formats.

Note: The AVnu Automotive Working Group will define additional media transport formats in future revisions of this document. As these formats are added, vendors of AEDs that support the added formats will need to seek certification for them.

7.1 AVTP Audio Format (AAF)

If an AED-A supports the AAF format, then it shall support the AVTP Audio Format (AAF) as defined in IEEE-1722a. [AVTPa, clause 8] AED-A's that support other audio formats, such as the 61883-6 format, will not have their audio format certified under the AVnu Automotive profile.

An AED-A shall support the sixteen (16) bit integer sample format as defined in IEEE-1722a. [AVTPa, clause 8.2.1]

An AED-A shall support at least one of 48kHz and 44.1kHz audio sample rates, as defined in IEEE-1722a. [AVTPa, clause 8.2.2]

An AED-A talker shall support at least one of either "full" or sparse timestamp mode as defined in IEEE-1722a. [AVTPa, Clause 8.2] The vendor of the AED-A needs to communicate which timestamp mode their device supports in its application for certification testing.

An AED-A listener shall support both full and sparse timestamp modes as defined in IEEE-1722a. [AVTPa, Clause 8.2.5]

Dynamic changes in audio layout can occur during normal operation of a vehicle audio system. For instance the user may switch from a 5.1 surround song to a stereo song. In this case, the number of actual audio channels being transmitted to the amplifier doesn't change, but the layout for using those channels does change. To accommodate this, the event field in the AAF header [AVTPa, Clause 8.2.6] is defined in Table 14 to signal a dynamic change in audio channel layout. An AED-A shall support at least one of the dynamic layouts 0, 1, 2, 3 or 4 as defined in Table 14. If an AED-A supports the static layout (layout 0), then the vendor needs to communicate which static mode audio layouts the device supports in the application for certification testing.

Event Field	*Layout Description*	*Valid Channels*	*Meaning of Valid Channels*
0_{16}	Static layout	Defined in 1722a	Defined in 1722a
1_{16}	Mono	0	mono
2_{16}	Stereo	0, 1	1722a, Table 8.4
3_{16}	5.1	0, 1, 2, 3, 4, 5	1722a, Table 8.5
4_{16}	7.1	0, 1, 2, 3, 4, 5, 6, 7	1722a, Table 8.6
$5_{16} - F_{16}$	Defined by System Integrator	Defined by System Integrator	Defined by System Integrator

Table 14: Dynamic Channel Layouts

The total number of channels per AAF frame (specified by the channels_per_frame field in the AAF header [AVTPa, clause 8.2.3]) shall be no less than the number of valid channels in a layout, in order to transport that layout. For instance, to carry a stereo layout, the number of channels per frame must be at least two (2).

An AED-A talker shall transmit the event field as defined in Table 14 for all layouts that it supports. *Note*: For testing purposes, the provider of the AED-A must provide testers with a mechanism for switching between layouts.

An AED-A listener shall use the event field to determine its output channel layout as defined in Table 14. For layouts that the AED-A supports, it shall render them with the meanings listed in Table 14. For layouts that it doesn't support, the AED-A shall not render unintended audio output or suffer other malfunction. For instance, if an AED-A only supports the stereo and 5.1 layouts, when it receives a mono stream, it should continue to operate normally and not render the audio data received on channels 1, 2, 3, and 4.

If an AED-A listener supports two or more non-static layouts (1, 2, 3 or 4) as defined in Table 14 then it shall support dynamic switching between its supported modes.

An invalid channel is any channel carried in the current stream, but not valid according to the current channel layout, as defined in Table 14. An AED-A talker shall fill invalid streams with all zeroes. An AED-A listener shall ignore any data received in invalid channels and not render any audio output for invalid channels.

7.2 Compressed Video Formats

If an AED-V supports any of the following video formats, then it shall support them as defined in IEEE-1722a. AED-V's that support video formats not described here do not need to seek certification on such formats. AVnu reserves the right to add video formats in future versions of this document, at which time an AED-V that supports one or more of these added formats will need to seek certification.

- H.264 [AVTPa, Clause 9.4]

- Motion JPEG (MJPEG) [AVTPa, Clause 9.3]

7.2.1 H.264

An AED-V talker that supports H.264 shall produce an AVTP packet stream that is compliant with the H.264 AVTP mapping. [AVTPa, Clause 9.4]

An AED-V listener that supports H.264 shall render video with no user observable glitches or artifacts when receiving an AVTP stream compliant with the H.264 mapping. [AVTPa, Clause 9.4] *Note*: Full testing of the video decoder is beyond the scope of AVnu testing. However, an AED-V listener should be able demonstrate video decode and rendering capabilities sufficient to satisfy visual inspection by a human tester.

7.2.2 Motion JPEG (MJPEG)

An AED-V talker that supports MJPEG shall produce an AVTP packet stream that is compliant with the MJPEG AVTP mapping. [AVTPa, Clause 9.3]

An AED-V listener that supports MJPEG shall render video with no user observable glitches or artifacts when receiving an AVTP stream compliant with the MJPEG mapping. [AVTPa, Clause 9.3] *Note*: Full testing of the video decoder is beyond the scope of AVnu testing. However, an AED-V listener should be able demonstrate video decode and rendering capabilities sufficient to satisfy visual inspection by a human tester.

7.3 MPEG2-TS Container Format

An AED talker that supports MPEG2 transport stream container format shall produce an AVTP packet stream that is compliant with the 61883-4 AVTP mapping. [AVTP, Clause 6]

7.4 Clock Reference Format (CRF)

An AED-C audio talker shall support an audio sample CRF (CRF_AUDIO_SAMPLE) as defined in IEEE-1722a. [AVTPa, Clause 11] In addition to verifying compliance with the standard, testing will record frequency and jitter of the timestamps in the transmitted CRF stream. The AED-C audio CRF talker shall produce a clock reference stream whose observed frequency value as measured from CRF timestamps varies from nominal by no more than +/-100PPM.

If an AED-C is both a CRF listener and a talker generating a stream (of any format) that is synchronized to the received CRF stream, then the presentation timestamps of the generated stream shall be within 5% of the associated timestamp positions in the original CRF stream.

Note: Because access to the recovered media clock cannot be guaranteed in the test lab, CRF requirements on listeners are not currently part of this specification.

8 Stream Reservation (SR) Classes

The automotive Ethernet-AVB network has several constraints that a typical Ethernet network does not. In particular, automotive networks are often limited to a 100Mbps data rate. This is due to the need to choose a physical layer that operates on a single pair and without shielding to reduce wiring harness cost and weight. This tight limit to 100Mbps strongly encourages bandwidth efficient engineering.

Simultaneously, engineers of Ethernet AVB networks face the age-old requirement of designing cost effective solutions. Implementations of the two existing AVB SR classes (class A and class B) require AVTPDUs to be transmitted in 125µs and 250µs intervals, respectively. These intervals place another constraint – especially for uncompressed audio streams – that result in media data blocks that are unnatural for today's DSP and DMA engines that typically operate on data blocks of 32 or 64 audio samples.

These two issues – network bandwidth efficiency and natural data sizing – have driven some creative thinking regarding SR classes. AVnu's automotive working group has taken advantage of the self-contained nature of the automotive network to define SR classes that are specific to the automotive use case and that help to solve bandwidth efficiency and data sizing issues. Note that even though the automotive working group has defined more than two SR classes, this specification does not require any device to simultaneously support more than 2 SR classes.

8.1 Stream Reservation Class Requirements

An AED shall support at least one of the SR classes from Table 15. Design of these SR classes is described in [AltClasses] and pays careful attention to overall latency.

Note that most of the current AVB network implementations are limited to a maximum of two SR classes, since they were designed to support only classes A and B from [802.1BA]. This specification purposely does not impose this limitation, in order to allow future implementations that may need more than two traffic classes.

Class	***A***	***B***	***64 Sample, 48kHz***	***64 sample, 44.1kHz***
Measurement Interval	125µs	250µs	1333µs	1451µs

Table 15: SR Class Measurement Intervals for Automotive Networks

For each SR class that it supports, an AED shall support a measurement interval as described in Table 15.

An AED shall use class IDs 5 and 6 to identify the SR classes associated with each stream. An OEM will typically configure the vehicle network to use no more than two SR classes and so that streams with higher priority use class ID 6, and streams with lower priority use class ID 5. Table 16 gives an example of SR class ID use for the classes defined in this specification. Note that even when the vehicle's network uses SR classes not defined in this specification, the network devices still use class IDs 5 and 6.

SR Class	***A***	***B***	***64 Sample, 48kHz***	***64 sample, 44.1kHz***
A only	6	--	--	--
B only	--	6	--	--
64x48k only	--	--	6	--
64x44.1k only	--	--	--	6
A + B	6	5	--	--
A + 64x48k	6	--	5	--
A + 64x44.1k	6	--	--	5
B + 64x48k	--	6	5	--
B + 64x44.1k	--	6	--	5
64x48k + 64x44.1k	--	--	6	5

Table 16: SR Class ID Values for Automotive Networks

The AVnu Automotive CDS group recommends mapping SR Class Priorities to the two classes on a network so that priority 3 goes to the class with the shortest observation interval (the "high class") and that priority 2 goes to the class with the longer observation interval (the "low class"). This is consistent with Table 6-6 of 802.1Q. The high class should also have a max transit time shorter than or equal to the max transit time of the low class. Table 17 describes this mapping for the classes defined in this section. For each SR class that it supports, an AED should support the SR class priority as described in Table 17. Note that, in a manner consistent with Table 34-1 or 34-2 of 802.1Q, the priority assigned to the SR class with the shortest observation interval (e.g. 3) must map to the highest priority traffic class available on each device in the AVB network. If supported, a second SR class must be similarly mapped in a manner consistent with Table 34-1.

For each SR class that it supports, an AED shall support a SR class priority as described in Table 17.

SR Classes on Vehicle Network	*A*	*B*	*64 Sample, 48kHz*	*64 sample, 44.1kHz*
A only	3	--	--	--
B only	--	3	--	--
64x48k only	--	--	3	--
64x44.1k only	--	--	--	3
A + B	3	2	--	--
A + 64x48k	3	--	2	--
A + 64x44.1k	3	--	--	2
B + 64x48k	--	3	2	--
B + 64x44.1k	--	3	--	2
64x48k + 64x44.1k	--	--	3	2

Table 17: SR Class Priority Values for Automotive Networks

An AED shall set regeneration override values according to 802.1Q-2011, clause 6.9.4.

Note: AVnu compliance testing will test all SR classes supported by the AED. Testing will be a standard FQTSS test, including class performance and packet jitter measurements. The vendor of the AED must either

- Disclose the procedure for configuring the streams on their product, including SR class ID, SR class priority, and measurement interval, or
- Communicate the static settings for the streams, if they are preconfigured into the device.

8.2 Presentation Time Parameters

8.2.1 Maximum Transit Time

The Maximum Transit Time (also known as Presentation Time Offset) is the time offset added at the talker to the gPTP observation time to create the presentation time of the packet at the listener. Table 18 specifies the maximum transit times for each SR class. For each class it supports, the AED shall support the maximum transit time specified in Table 18.

Class	*Maximum Transit Time*
A	2 ms
B	10 ms (Note 1)
64x48k	15 ms
64x44.1k	15 ms

Table 18: SR Class Maximum Transit Times

Notes on Table 18:

1. The standard maximum transit time for class B streams [802.1BA] assumes two wireless segments in its calculations. This does not apply to automotive applications. Therefore this specification appropriately reduces MTT for class B.

The Maximum Transit Times shown in Table 18 are defaults. OEMs may specify alternative values for their applications.

The MTT values for classes A and B come from IEEE 1722. The MTT values for the 64 sample classes are based on the example data shown in Table 19, originally from [AltClasses]. These calculations use the formula defined in IEEE 802.1BA, section 6.5 and assumes 8 mono channels of audio in 16-bit AAF format. A 15ms MTT provides sufficient headroom so that the MTT is sufficient to cover worst-case network topologies and stream bandwidths.

Class	***A***	***B***	***64x48kHz***	***64x44.1kHz***
Class measurement Interval	**125us**	**250us**	**1333us**	**1451us**
tDevice (us)	5.12	5.12	5.12	5.12
tMaxPacketSize + IPG (us)	123.36	123.36	123.36	123.36
tStreamPacket (us)	8.32	10.24	86.24	86.24
tStreamPacket+IPG (us)	9.28	11.2	87.20	87.20
Rate (Mb/s)	100	100	100	100
MaxAllocBand (Mb/s)	75	75	75	75
tInterval (us)	125	250	1333	1451
tAllStream (us)	93.75	187.5	1000.0	1088.4
Max Hop Latency (us)	**249.43**	**373.79**	**1431.8**	**1549.7**
Max Transit Time (us) over seven hops	**1746.0**	**2616.5**	**10,023**	**10,848**

Table 19: SR Class Example Hop Latency and Maximum Transit Time for a Seven-Hop Network

8.2.2 Maximum Timing Uncertainty

The Maximum Timing Uncertainty is defined in IEEE 1722A [1722A, section 5.3.3]. It represents the uncertainty in the delivery of AVTP packets to/from the media interface and the AVTP layer of the device. The Max Timing Uncertainty is used in calculating the buffer size in listeners so that they can accommodate all data, even in scenarios presenting the worst-case network delay. [1722a, section 5.3.3, equation 2] Table 20 specifies the maximum transit times for each SR class. For each class it supports, the AED shall support the maximum timing uncertainty specified in Table 20.

Class	***Maximum Timing Uncertainty***
A	125 µs
B	1000 µs
64x48k	500 µs
64x44.1k	500 µs

Table 20: SR Class Maximum Timing Uncertainty

8.3 Stream Reservation Requirements

An AED shall support statically configured reservations for all AVB streams.

Note: AVnu compliance testing will test all streams supported by the AED. Testing will be a standard FQTSS test, including class performance and packet jitter measurements. The vendor of the AED shall either

- Disclose the procedure for configuring the streams on their product, including SR class ID, SR class priority, and measurement interval, or

- Communicate the static settings for the streams, if they are "hardwired" into the device.

9 Exception Handling

All implementations of Automotive Ethernet networks will likely need to comply with the OEM or System Integrator requirements for error and exception handling. These will differ, and so the specifics are beyond the scope of this document. However this section does identify a subset of exceptions associated with the AVB and Ethernet service layers, and provides guidance on how these should be handled.

These exception types are discussed in the following sections:

- AVB Device Failure
- Ethernet link state events and data loss
- 802.1AS exceptions
- IEEE 1722 Media Stream exceptions

This document assumes that all devices operating in a vehicular environment shall be developed to be highly robust and resilient to exception conditions during startup, shutdown and normal operation. Generally speaking, devices should employ techniques that allow exception conditions to be reported and, where possible, automatically corrected without driver or service intervention, and without significant functional degradation in the vehicle.

The AVB device shall retain error logs or diagnostic codes for subsequent service retrieval and analysis via a mechanism defined by the System Integrator, such as a UDS based method. For the purposes of AVnu certification testing, the AED vendor shall provide a mechanism for retrieving logged data at the time they submit a device for certification testing. Certification test personnel will use this mechanism to verify all logged data.

9.1 AVB Device Failure

While not required by this specification, it is generally expected that each device attached to the AVB network will have the following exception handling properties:

- Recover and continue operation when encountering recoverable exceptions
- Restart in the case of fatal errors
- Watchdog timer support to self-restart in the case of a software hang
- Log exceptions and provide a mechanism for their retrieval
- Operate with the in-car OAM and network management sub-system

While the AVB protocols can conceivably be used to perform certain OAM functions (e.g. partner health check), it is not their primary purpose and, unless otherwise stated, they should not be relied upon for this purpose.

9.2 Ethernet Exceptions

9.2.1 Link State Events

9.2.1.1 Link-Down

A link-down event on a port indicates the loss of Ethernet connectivity to the link partner on that port. All traffic, including media and AVB protocol traffic, will stop. In a traditional, non-automotive network, link-down will typically occur due to a failure or restart of the link partner device, a cable fault, a cable unplug event or the link partner device being administratively disabled. Since an automotive network is typically static and unmanaged, this event is most likely due to either a link partner failure/restart or a cable fault. Note that a link down event during a system shut down is not considered to be an exception. The System Integrator will often define additional system behavior in the case of a link-down event, but at a minimum the AED shall always timestamp and log any link-down event for later retrieval.

9.2.1.2 Link-Up

When link-up follows a link-down event on a port, this indicates the recovery from the root failure of the original link-down event. Often this is the recovery of the failing link partner or cable. The System Integrator will often define additional system behavior in the case of a link-up event, but at a minimum on a link-up event, the AED:

- Shall timestamp and log the link-up event for later retrieval.
- If it is configured in test mode, the AED-E shall issue status messages as it progresses through ETHERNET_READY, AVB_SYNC and MEDIA_READY states, as described in section 1.
- Make its best effort to discard any unprocessed data received on the port prior to the link-up event.
- Make its best effort to detect the validity of any data that was queued for transmission on the port prior to the link-up event, and discard invalid data.
- Shall restart all talker streams the AED-T has been configured to provide.
- Shall restart all listener streams the AED-L has been configured to support.

Note that link-up at initialization time is not considered to be an exception.

9.2.2 Data Loss

A stream of data may be unexpectedly lost or interrupted for a number of reasons, including the link-down event described above. This section describes how an AVB device should handle loss of one or more Ethernet frame(s).

An AED-L:

- Shall timestamp and log the event for later retrieval.
- Shall ensure that no unintended audio or video output is generated due to data loss.

9.3 IEEE 802.1 AS Exceptions

9.3.1 Loss of Sync Messages

In an IEEE 802.1AS network, if a time-aware port in the slave role experiences a loss of sync (detected by a link loss or a SyncReceiptTimeout event on the port) then appropriate BMCA activity will be triggered. However, per section 6, BCMA does not run in an automotive Ethernet network. Instead, an AED detecting a loss of sync shall behave as follows:

- Timestamp and log the event for later retrieval.

9.3.2 Non-Continuous Sync Values

Per IEEE 802.1AS, a gPTP slave is required to be able to gracefully handle a discontinuity in the Sync values it receives. An instance of this is the GM recovery condition described in section 6.2.6.

To detect a gPTP timebase discontinuity, the AED-E

- Shall monitor the correction and preciseorigintimestamp fields. During the period when the GM is offline, the correction field will grow because the AED-B is in proxy operation. Then, when the GM comes back online, the correction field will suddenly jump back to a very small value.

- Shall monitor the gmTimeBaseIndicator field for change. A change in the gmTimeBaseIndicator field indicates that the GM has a new timebase, which will happen when the GM faults and then comes back online.

- Shall monitor the tu bit, if the AED-E is an AVTP listener.

On detection of a discontinuity of Sync values,

- The AED shall timestamp and log an exception event of the Sync discontinuity.

- A talker shall transition and begin using the new timebase as the basis for presentation times in AVTPDUs within five (5) of the current sync periods. The AED-T shall set the AVTP tu bit as appropriate during this transition period.

- A listener shall accommodate the change in timebase without a discernable glitch in the media playout. To achieve this, an AED-L is likely to use its local clock to play out data at the normal rate during this indeterminate period, but the exact algorithm is left to the vendor. Note that the change in timebase could occur before presentation timestamps change, or vica versa. The listener doesn't know and should be prepared to handle both cases.

9.3.3 Pdelay Response Timeout

In an IEEE 802.1AS device, if 3 successive Pdelay response timeout events occur then the asCapable variable for the port is set to false, which in turn causes the cessation of specified 802.1AS protocol activity on the port. In an automotive network, this is not the case – asCapable will remain true for the port and the protocol will continue as-is (including sending Sync messages at the configured interval), using the prior Pdelay measurement. Therefore the Pdelay Response Timeout is for diagnostics only, and has no effect upon other AVB behavior, and an AED shall continue to send Pdelay messages at the configured interval. For the diagnostic purpose, with reference to the MDPdelayReq state machine, when the number of lostResponses exceeds the allowedLostResponses on a port, the event should be logged and timestamped for later retrieval. The `ieee8021AsPortStatPdelayAllowedLostResponsesExceeded` counter should be incremented per the IEEE 802.1AS standard. The `ieee8021AsPortStatPdelayAllowedLostResponsesExceeded` should not continue to increment for further successive lost responses until at least one successful response is received (and therefore lostResponses is reset to 0).

9.3.4 neighborPropDelay value

In an IEEE 802.1AS device, a reading of neighborPropDelay that is outside of a configured threshold (neighborPropDelayThresh) will cause asCapable on the port to go false. This behavior is not required in this specification, and an associated exception event is also not required.

9.4 IEEE 1722 Media Stream Exceptions

Each 1722 AVTPDU indicates a sequence number for the stream. This allows missing or duplicate AVTPDUs to be detected at the receiver. However since the sequence number is a bit field of limited size, it is possible to have an error condition that would be missed by only checking the sequence number. Therefore the AED-L should use the sequence number in conjunction with the 1722 Presentation Timestamp to determine whether the discontinuity represents a duplicate AVTPDU or a set of missing AVTPDUs.

If an AED-L is capable of detecting missing AVTPDUs, it shall

- Timestamp and log the event for later retrieval,
- Increment the 1722A diagnostic counter “SEQ_NUM_MISMATCH,”
- If a number less or equal to a vendor-defined number of AVTPDUs is missing, the AED-L may replay the previous AVTPDU information, if still available, to prevent a gap.
- If more than a vendor-defined number of AVTPDUs are missing, the AED-L shall stop output.

If an AED-L is capable of detecting duplicate AVTPDUs, it shall

- Timestamp and log the event for later retrieval,
- Increment the 1722A diagnostic counter “SEQ_NUM_MISMATCH,”
- Discard the duplicate frames.

If the AED-L is receiving safety-relevant video information (for example, rear view camera data), and it experiences lost or late AVTPDUs, it shall display blank video during any period of lost or late video data. Video data is considered late if the AED-L cannot display it until after its AVTP presentation timestamp. The AED vendor notifies the testing facility if the AED is handling safety-relevant video streams.

In the case where the AED-L is receiving entertainment audio and/or video information and experiences lost or late AVTPDUs, this specification recommends that it use suitable and practical methods to gracefully mitigate the loss of data.

Under no conditions shall an AED-T send outdated data, which might have accumulated during an error condition, into the network. Outdated data is defined here as any AVTPDU that cannot be sent such that it reaches all listeners in time to be presented at its presentation timestamp.

10 Security

Due to the complexities related to automotive security in general and Ethernet security in particular, this section is deferred to a later release of the document.

11 Content Protection

Content Protection provides cryptographic protection on copyrighted material as it traverses an AVB network. System integrators and OEMs will supply specific requirements for AEDs in the area of content protection. For reference on Content Protection on AVB traffic, please refer to Annex H in IEEE 1722a [1722a].

12 Network Management

Deferred to a later release of this document.

13 Diagnostic Counters

Ethernet diagnostic counters are important to have in Ethernet devices in order to aid monitoring and diagnostics of network performance and troubleshooting of networking faults.

This section discusses and defines requirements for diagnostic statistics for the following service layers:

- Ethernet interfaces (PHY, MAC)
- Ethernet Bridging
- AVB Protocols
- 1722 Transport

Counters for higher service layers, such as TCP/IP or application level, are out of this document's scope.

For the purposes of AVnu certification testing, the AED vendor shall provide a mechanism for accessing each counter that it supports, and the format for returning the value of the counter. For all statistical counters supported by an AED, the AED shall provide a read operation.

13.1 Ethernet Interfaces

Most available Ethernet hardware offers counters at the Physical or MAC service layers that offer insight to the performance of the Ethernet interfaces on the device. These can be collected at many levels, including device, port, priority queue and VLAN, and mostly count packets, both good and bad, of various types. Such counters are standardized in the following specifications:

- RFC 3635 – MIB for Ethernet-like Interface Types (see `dot3StatsTable`)
- RFC 2819 – RMON MIB (see the Ethernet Statistics Group)

Because it is generally good practice to offer such statistics vendors of AED's are encouraged to provide support for them. However, AED's are not required to support them.

13.2 Ethernet Bridging

RFC 4188 (Managed Objects for Bridges) specifies some standard statistics, but is mostly comprised of configuration and status. Vendors of AED-B's are encouraged to provide support for them. However, AED-B's are not required to support them.

13.3 AVB Protocols

The 802.1AS standard specifies the `ieee8021AsPortStatIfTable` in its MIB. All AED's shall maintain the counters in this table and make them available for certification testing. These counters include:

- `ieee8021AsPortStatRxSyncCount`
- `ieee8021AsPortStatRxFollowUpCount`
- `ieee8021AsPortStatRxPdelayRequest`
- `ieee8021AsPortStatRxPdelayResponse`
- `ieee8021AsPortStatRxPdelayResponseFollowUp`
- `ieee8021AsPortStatRxAnnounce`
- `ieee8021AsPortStatRxPTPPacketDiscard`
- `ieee8021AsPortStatRxSyncReceiptTimeouts`
- `ieee8021AsPortStatAnnounceReceiptTimeouts`
- `ieee8021AsPortStatPdelayAllowedLostResponsesExceeded`
- `ieee8021AsPortStatTxSyncCount`
- `ieee8021AsPortStatTxFollowUpCount`
- `ieee8021AsPortStatTxPdelayRequest`
- `ieee8021AsPortStatTxPdelayResponse`
- `ieee8021AsPortStatTxPdelayResponseFollowUp`
- `ieee8021AsPortStatTxAnnounce`

The 'Announce' related values are optional and should return zero where BMCA is not implemented.

13.4 1722 Transport

An AED-E shall support the counters listed in IEEE 1722a Annex E [1722a].