

DGei Time Synchronization Profile

Christian Boiger

Public

Agenda

How to integrate time synchronization into IEEE Std 802.1DG?

Time Synchronization Profile Content

How to integrate time synchronization into IEEE Std 802.1DG?

Adding Time Sync Specification to IEEE 802.1DG

- IEEE Std 802.1DG does not specify time synchronization but it is also not silent on it
- There has been some discussion on how to add the time synchronization profile
- There were some concerns raised on backward compatibility / (in)compatibility / conformance with IEEE Std 802.1DG-2025
- IEEE Std 802.1DG mentions time synchronization but is very unspecific about it
- IEEE Std 802.1DG does not provide any interoperability between IEEE 802.1DG compliant implementations regarding time synchronization

Current Specification in IEEE 802.1DG - Bridges

- Current text in .1DG for Bridges:
  - Conformance clause: “Should support at least one time synchronization mechanism as stated in D.5.”
  - PCS: “Does the bridge support time synchronization? If Yes, provide details on the supported time synchronization solution:”
  - D.5 “Time synchronization is an essential component of in-vehicle use-cases of this standard; however, this standard allows different mechanisms to be used for time synchronization, as different mechanism have already been deployed in the field. Some examples of time synchronization specifications that can be used in applications requiring time synchronization are as follows:”

Current Specification in IEEE 802.1DG – End Stations

- Current text in .1DG for end stations
  - Conformance clause: “An end station claiming conformance to this standard may: a) Support time synchronization as stated in D.5.”
  - PCS: “Does the end station support time synchronization? If Yes, provide details on the supported time synchronization solution:”
  - D.5 “Time synchronization is an essential component of in-vehicle use-cases of this standard; however, this standard allows different mechanisms to be used for time synchronization, as different mechanism have already been deployed in the field. Some examples of time synchronization specifications that can be used in applications requiring time synchronization are as follows:”

Conclusion

- IEEE 802.1DG mentions time synchronization but is very unspecific about it
- IEEE 802.1DG does not provide any interoperability between IEEE 802.1DG compliant implementations regarding time synchronization
- Adding a mandatory profile is “backward compatible” in the sense that a device being compliant to the new profile is still compliant to the old
- Interoperability was never provided; this means a mandatory profile does not break interoperability
- Devices compliant to a previous profile might require changes to be compliant to new revisions of a profile

→ Adding a mandatory time synchronization profile to IEEE P802.1DGei seems to be a possible option

Options To Add Time Synchronization

	Add a time synchronization profile in DGei as an option	Add a time synchronization profile in DGei as a recommendation	Add a mandatory time synchronization profile
Time Synchronization support is optional			End stations?
Time Synchronization support is recommended			Bridges?
Time Synchronization support is required			Bridges?

Time Sync Profile Content

Pdelay measurement

- Pdelay is only needed on timeReceiver ports for link delay and rate ratio calculation
- Pdelay on timeTransmitter ports can be used for
  - Fast change of sync tree
  - Validation of rate offsets and link delay values
- IEEE Std 802.1ASds added the option to disable Pdelay in one direction
- Feature is not explicitly limited to 10BASE-T1S links
- Proposal to provide this as optional feature (which would be also aligned with current Automotive use)
- Bridges shall be required to process Pdelay Requests on timeTransmitter and timeReceiver ports
- Processing can be made optional for end stations, only grandmaster capable devices are required to support

Pdelay measurement (2)

- **Should disable Pdelay on timeTransmiter ports**
  - pdelayReqSendDisabled=true
- **Shall enable on timeReceiver Ports**
  - pdelayReqSendDisabled=false
- **Process received Pdelay Requests on timeTransmitter and timeReceiver ports**
  - Bridge shall be able to process (pdelayRespSendDisabled=false)
  - Grandmaster capable devices shall be able to process (pdelayRespSendDisabled=false)
  - Not grandmaster capable end stations may/should be able to process

Pdelay measurement – CMLDS (1)

Should we use CMLDS?

- Goal of profile: Eliminate unnecessary options
- 10BASE-T1S: Requires instance specific Pdelay
- **Full-Duplex:**
  - Option to use CMLDS or instance specific for domain 0 only
  - More domains require CMLDS
- Other Automotive specifications also allow instance specific Pdelay on other domains or define it to be used on lowest used domain number

Pdelay measurement – CMLDS (2)

Proposal:

- Require CMLDS on all full-duplex point to point links
- Require instance specific on 10BASE-T1S links

Benefits:

- Only one option
- Supports multiple domains
- IEEE Std 802.1AS compliant

Drawbacks

- Two different mechanisms required, CMLDS for full-duplex and instance specific for 10BASE-T1S

Announce Message

- Announce messages are typically not used in Automotive networks
- Disabling them would be in line with current usage
- **Announce messages shall be not supported**
  - announceMessageDisabled = TRUE (IEEE P802.1ASeb)

10BASE-T1S support in DG

- Do we support 10BASE-T1S in IEEE P802.1DGei?
- So far DG is kind of limiting itself to full-duplex (this is not limited by scope but mentioned in text)
- “A station claiming compliance with this standard shall support an IEEE Std 802.3 point-to-point full-duplex ([802.3] 1.4.345) link segment ([802.3] 1.4.379)”
- 10BASE-T1 is highly relevant for Automotive and IEEE Std 802.1ASds is available
- **Proposal: Include 10BASE-T1S for time synchronization**

Basis of Profile

- IEEE 802.1AS-2025 as amended by IEEE 802.1ASds and IEEE P802.1ASeb

- Need ASds for Pdelay in only one direction
- Need ASds for 10BASE-T1S
- Need ASeb for optional Announce messages

Domains

- **At least one domain with the domain number 0**
  - Domain number 0 mandatory or only default?
- Deviate for domain 0 from IEEE 802.1AS defaults and require externalPortConfigurationEnabled = TRUE
- **How many domains should we support?**
  - Define maximum 1, 2, 4?
  - Leave open only minimum 1?
- If we support more than one domain number, do we split between a base time sync profile and an extended?

BTCA

- Automotive networks are engineered and BTCA is not required
- Removing BTCA simplifies implementations
- **BTCA shall not be used**
- Only externalPortConfigurationEnabled=TRUE supported
  - This includes domain 0

VLAN Tag

- Time synchronization messages as defined in IEEE Std 802.1AS shall not be VLAN tagged
- There are implementations in the Automotive field using VLAN tags
- **IEEE P802.1 DGei should explicitly disallow VLAN tag usage**
- If time synchronization messages with a VLAN tag are received the VLAN tag shall be ignored
- **Reasons for having VLAN tag**
  - Used for prioritization
  - Should add statement that time synchronization messages shall be transmitted with high priority

One-step vs Two-step

- IEEE Std 802.1AS one-step is quite complex
- Pdelay is anyway requires two-step
- Currently two-step is used in Automotive applications
- **Keep solution simple and limit to two-step operation**

Support of Drift Tracking TLV

- Designed for big networks
- Automotive networks do not contain long chains
- **Keep solution simple and disallow usage of Drift Tracking TLV**
- AUTOSAR currently lists it as an option?

Signaling Messages

Request to not use signaling messages

Two types of messages:

Message interval request TLVs

- Are optional
- Are used by the Avnu profile (fast startup and reduce rate later) -> is this used?
- Proposal to not support message interval request TLVs

gPTP-capable TLV

- Can be disabled by setting gptpCapableStateMachinesEnabled=false
- But implementation of gPTP capable signaling is required (see 5.4.2 g) and h))
- Proposal to require that gptpCapableStateMachinesEnabled is false
- Ideally fix in IEEE Std 802.1AS?


infineon