



Proposed Solutions for P802.1ASds

Presenter: Don Pannell, NXP

Co-Authors:

Olaf Mater, Marvell

Silvana Rodrigues, Huawei *(Not an Avnu member)*

Woojung Huh, Microchip

Vlad Lyalikov, Microchip

September 2023 802.1 Interim



Administration

This presentation is the collaborative work of the Presenter and Co-Authors.

This is not an official publication from Avnu.org, but permission was granted to use the Avnu template to show multi-company consensus among contributing Avnu members and other non-member contributors on the items enclosed.

Consider this as a proposal of the gPTP mechanisms needed to support engineered 10BASE-T1S half-duplex links.

Summary of proposed solutions to date:

[1] Janker et al., proposed modifications to MDPdelayReq state machine:

- <https://www.ieee802.org/1/files/public/docs2021/dg-janker-timesync-in-10BASE-T1S-networks-0521.pdf>

[2] Pannell et al., presented a simplified 10BASE-T1S Use-Case model:

- <https://www.ieee802.org/1/files/public/docs2022/ds-pannell-Avnu-Automotive-UseCase-Requirements-0922-v01.pdf>

[3] Rodrigues et al., presented a shared media filtering mechanism using gPTP ClockID:

- <https://www.ieee802.org/1/files/public/docs2022/ds-Rodrigues-Lv-10BASE-T1S-time-sync-1122-v00.pdf>

[4] Pannell et al., presented Automotive Solution Implications:

- <https://www.ieee802.org/1/files/public/docs2023/ds-pannell-Avnu-Automotive-Solution-Implications-0323-v01.pdf>

Overview

- Unidirectional Pdelay operation was proposed for 10BASE-T1S links in [4]
 - There were no “immediate issues” brought up on this idea at the March 2023 meeting
 - So, this idea was expanded upon including re-looking at [1] & [2]
 - Resulting in the proposed, complete solution, contained herein
- Text changes required if 10BASE-T1S support is added to Clause 11 in [4]
 - Potentially problematic for editor to work with 98 instances of “full-duplex” in Clause 11
 - So, a new Media Dependent Clause was looked at with a proposed solution herein

The Goal of this presentation is to get consensus on the proposed approach such that an initial Draft of ASds can be created by the Editor. Specifically:

- Support for 10BASE-T1S in Engineered networks (the immediate need)
 - Networks with plug-and-play requirements can be supported with a separate amendment
- Create a new Media Dependent Clause for ASds (propose Clause 19)
- Which defines the use of Unidirectional Pdelay

UNIDIRECTIONAL PDELAY REQUIREMENTS AND SOLUTIONS

Status of Unidirectional Pdelay_Req

- Unidirectional Pdelay_Req is an old and proven concept
- It has been used since 2014 where it is part of the Avnu Automotive Interoperability Specification
 - <https://avnu.org/resource/automotive-ethernet-avb-functional-and-interoperability-specification-revision-1-6/>
- Unidirectional Pdelay_Req was also proposed in [1] along with some of the needed changes to the MDPdelayReq state machine (we will show a complete solution)

Unidirectional pDelay_Req

Refresher - From March 2023 preso

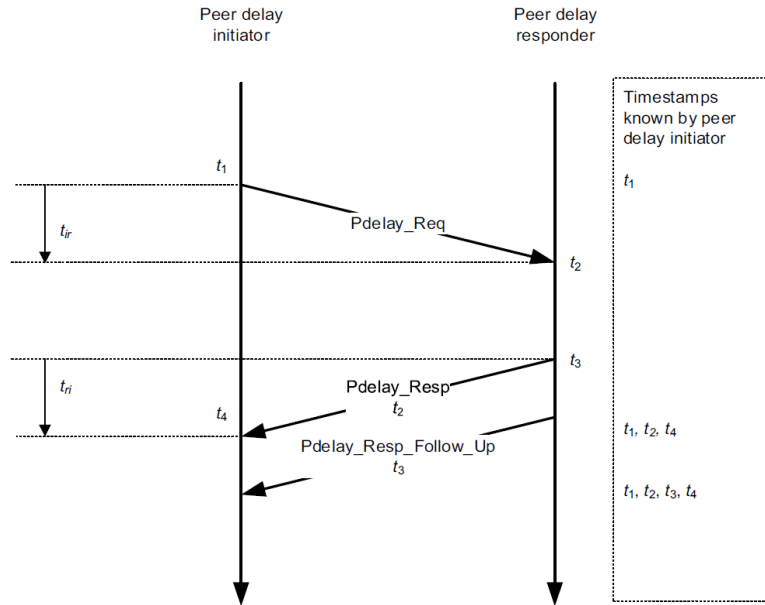


Figure 11-1—Propagation delay measurement using peer-to-peer delay mechanism

$$t_{ir} = t_2 - t_1$$

$$t_{ri} = t_4 - t_3$$

$$D = \frac{t_{ir} + t_{ri}}{2} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

- In full-duplex 802.1 AS, both sides of a link are pDelay_Req initiators
 - The initiator has all the timestamps necessary to calculate the mean link delay (D) relative to the GM timebase
- For ASds, we can restrict the initiator to TimeReceivers only (typically, End Stations) which send Pdelay_Req
 - Thus, TimeTransmitters don't need to keep a database of all its End Stations
 - Needed if Unicast DA's were to be used
 - Simplifies the state machine changes!
 - i.e., handling of multiple Pdelay_Resp in the TT

Operation of Unidirectional Pdelay_Req for 8 nodes, 1 domain, Pdelay PTP frames / Sec

	timeTx		timeRx1	timeRx2	timeRx3	timeRx4	timeRx5	timeRx6	timeRx7	timeTx = TimeTransmitter timeRxN = TimeReceiver N
1	rx	1x/Sec	PDRReq1	filter	filter	filter	filter	filter	filter	
2	rx		filter	PDRReq2	filter	filter	filter	filter	filter	
3	rx		filter	filter	PDRReq3	filter	filter	filter	filter	filter = don't respond to PdelayReq messages
4	rx		filter	filter	filter	PDRReq4	filter	filter	filter	
5	rx		filter	filter	filter	filter	PDRReq5	filter	filter	
6	rx		filter	filter	filter	filter	Filter	PDRReq6	filter	
7	rx		filter	filter	filter	filter	Filter	filter	PDRReq7	
8	PDRResp1		rx	filter	filter	filter	Filter	filter	filter	
9	PDRResp2		filter	rx	filter	filter	filter	filter	filter	Note: Frames on the wire could be in any order!
10	PDRResp3		filter	filter	rx	filter	filter	filter	filter	Note: TimeTx must not send PdelayReq messages
11	PDRResp4		filter	filter	filter	rx	filter	filter	filter	
12	PDRResp5		filter	filter	filter	filter	rx	filter	filter	
13	PDRResp6		filter	filter	filter	filter	filter	rx	filter	
14	PDRResp7		filter	filter	filter	filter	filter	filter	rx	
15	PDR_Fup1		rx	filter	filter	filter	filter	filter	filter	
16	PDR_Fup2		filter	rx	filter	filter	filter	filter	filter	filter = filter out PdelayResp & PdelayRespFollowUp messages that are not due to TimeRx's PdelayReq
17	PDR_Fup3		filter	filter	rx	filter	filter	filter	filter	
18	PDR_Fup4		filter	filter	filter	rx	filter	filter	filter	
19	PDR_Fup5		filter	filter	filter	filter	rx	filter	filter	
20	PDR_Fup6		filter	filter	filter	filter	filter	rx	filter	
21	PDR_Fup7		filter	filter	filter	filter	filter	filter	rx	

8 Sync /Follow_Up frames are also transmitted by timeTx each second. For a total FPS of 37, which is < 0.3% of 10BT1S' bandwidth!

Unidirectional PdelayReq Rules for ASds

- **TimeReceivers** using External Port Configuration:
 - 1) Ignore all PdelayResp & PdelayRespFollowUp messages whose requestingPortIdentity.clockIdentity is NOT equal to clockIdentity of this TimeReceiver
 - i.e., those messages that did not come as a result of my PdelayReq – so ignore them
 - These extra messages occur due to the IEEE 802.3 Clause 4 MAC being in Half-Duplex mode
 - 2) Do not respond to received Pdelay_Req messages
 - That's it!
- **TimeTransmitters** using External Port Configuration:
 - 3) Never send any Pdelay_Req messages and set asCapableAcrossDomains
 - That's it!
- Everything else is as defined in Clause 11's MDPdelayReq & MDPdelayResp state machines

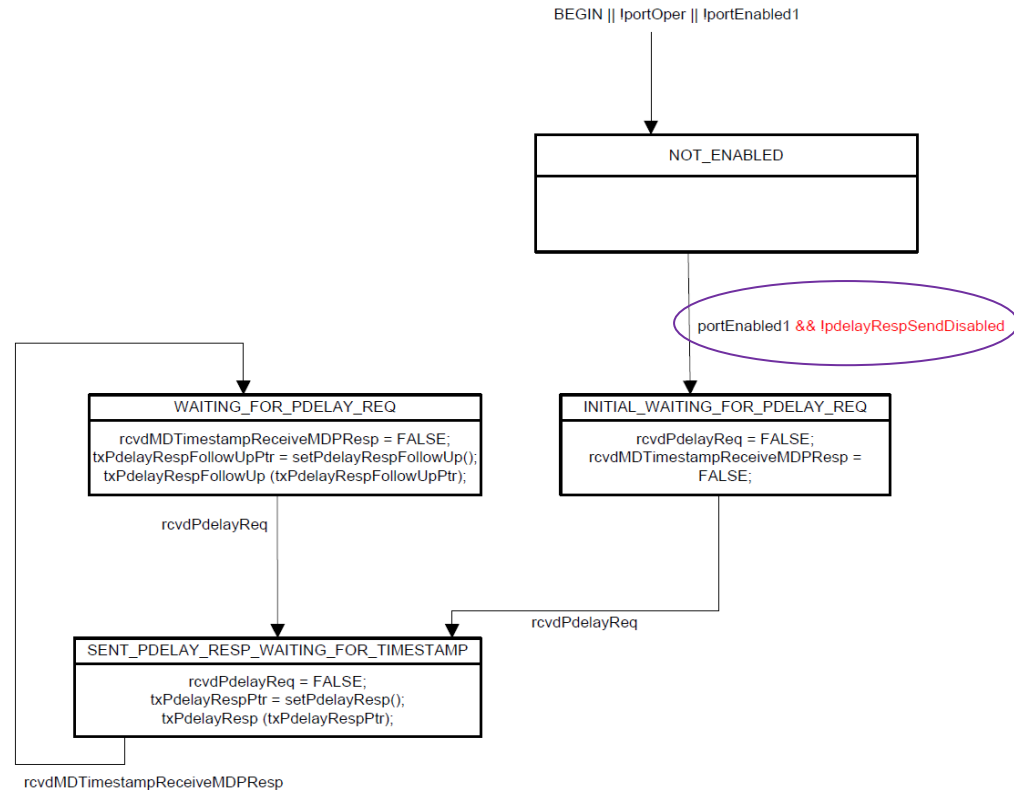
MDPdelayReq SM Changes for TimeReceivers

- Propose to change the SM's (state machine's) variable conditions only without needing to change the state machine!
- New Clause's State Machine variables descriptions:
 - **rcvdPdelayResp**: A Boolean variable that notifies the current state machine when a Pdelay_Resp message is received **and its requestingPortIdentity.clockIdentity is equal to clockIdentity of the current PTP instance**. This variable is reset by the current state machine.
[blue text is added to current AS-2020 11.2.19.2.2 text]
 - **rcvdPdelayRespFollowUp**: A Boolean variable that notifies the current state machine when a Pdelay_Resp_Follow_Up message is received **and its requestingPortIdentity.clockIdentity is equal to clockIdentity of the current PTP instance**. This variable is reset by the current state machine.
[blue text is added to current AS-2020 11.2.19.2.4 text]

MDPdelayResp SM Changes for TimeReceivers

In the new Clause:

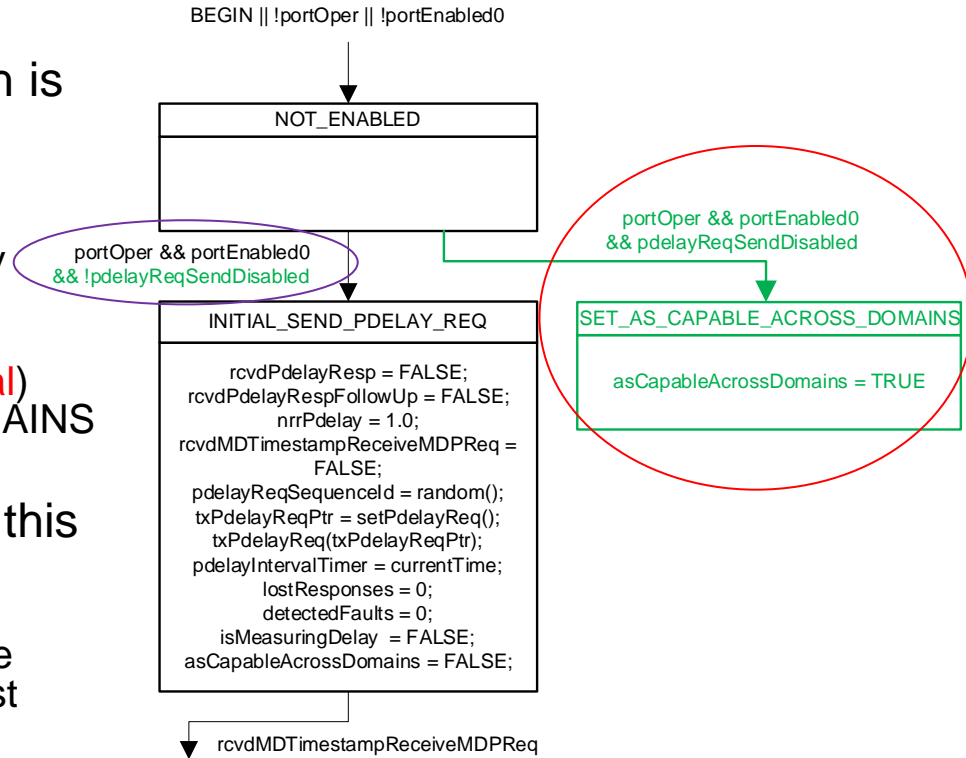
- Create a new managed object, which is set on ASds **TimeReceivers** only, called **pdelayRespSendDisabled**:
 - When set, this port will not transmit any Pdelay_Resp messages (purple oval)
 - It stops the state machine from doing anything – stays in NOT_ENABLED
- That is the only change needed for this State Machine!



MDPdelayReq SM Changes for TimeTransmitters

In the new Clause:

- Create a new managed object, which is set on ASds **TimeTransmitters** only, called **pdelayReqSendDisabled**:
 - When set, this port will not transmit any Pdelay_Req messages (**purple oval**)
 - And it branches to a new state (**red oval**) **SET_AS_CAPABLE_ACCROSS_DOMAINS**
- That is the only changes needed for this State Machine!
 - As previously shown, the state machine variable description changes do the rest



NEW MEDIA DEPENDENT CLAUSE

REASONS AND SOLUTIONS

Why a New Media Dependent Clause 19

- New Clause's Title:
 - Media-dependent layer specification for IEEE 802.3 Clause 4 Media Access Control (MAC) operating in half-duplex (the title of this PAR's amendment)
- Results in a faster Time to STD!
 - Plan is to mimic Clause 16's structure (the Media-dependent Clause of CSN)
 - CSN refers to Clause 10 & 11 (full-duplex Ethernet) then defines it's needed changes only
 - As in Clause 16, open with an Overview including a description of 10BASE-T1S followed by the characteristics of 802.3 Clause 4 half-duplex & 10BASE-T1S with & without PLCA
 - This approach will be easier for the Editor, Reviewers & Users!
 - As all the needed changes are in a self-contained new Clause
 - There is no need to change anything in Clause 10 or 11
 - No state machine changes, no changing the 98 instances of "full-duplex" in Clause 11
 - Takes the risk of "breaking" existing media solutions to zero as nothing is changed!

List of New Media Dependent Clause Changes

- List what is not supported in ASds (as done in Clause 16):
 - Signaling messages are proposed to be ignored
 - As they are not needed & will delay the STD! (as it needs a totally new algorithm)
 - Specifics on this still need to be defined – but the writing of the draft can start
 - List of the changes needed for Unidirectional Pdelay:
 - Include a new MDPdelayReq and MDPdelayResp SM figures
 - With the new definitions of the MDPdelayReq SM variable conditions
 - And the new managed objects as shown above
-
- In Clause 14: Need to include the new managed objects
 - pdelayReqSendDisabled & dpdelayRespSendDisabled

**DO THESE CHANGES MEET THE
GOALS DEFINED IN [2]?**

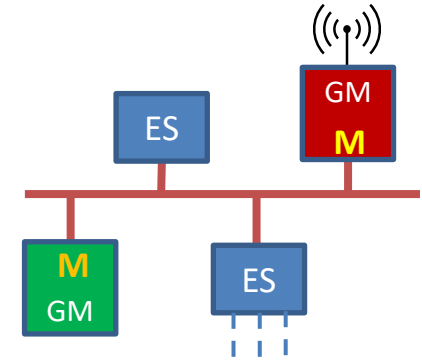
Goals from the Use Case Presentation [2]

- ASds must work on constrained devices:
 - i.e, minimal code changes – DONE
 - No new complex mathematics algorithms – DONE
- External Port Configuration with Single Domain – DONE
- Faster time to STD:
 - Fix Pdelay issues only – DONE
 - Don't change Sync nor Follow_Up – DONE
 - Most, if not all changes in new Media Dependent Clause – DONE
 - Do only what is needed today – See next slide

More Goals from the Use Case Presentation [2]

- External Port Configuration with Multiple Domains
 - Appears to work if CMLDS is not used & if TimeReceivers needing clocks from multiple TimeTransmitters need to send a separate Pdelay_Req per domain
- Plug-and-Play Use Case with one Domain
 - While a known solution is referenced in [2], the above proposed solution also appears to work in this use case
 - Which removes the need of Unicast Destination Addresses and their needed address database & learning mechanisms in the TimeTransmitter
 - As all filtering is done against an TimeReceiver's clockIdentity, which it knows!
- Verifying these will delay the STD! Are these really needed today?
 - This is in line with comments received in the past. So, PASS on both

SUMMARY



- Use case model from [2] is still valid:
- But postpone standardization of multiple Domains & Plug-n-Play
 - As the immediate support for ASds is for Engineered networks
- All the other Use Case goals from [2] are addressed
- Based on this data, we felt it is time to create an initial Draft of ASds:
 - Using a new Media Dependent Clause (Clause 19?)
 - And using Unidirectional Pdelay as defined above
- Are there any concerns on creating an initial ASds Draft?



QUESTIONS?

COMMENTS?

CONCERNS?

Thanks!



BACKUP SLIDES

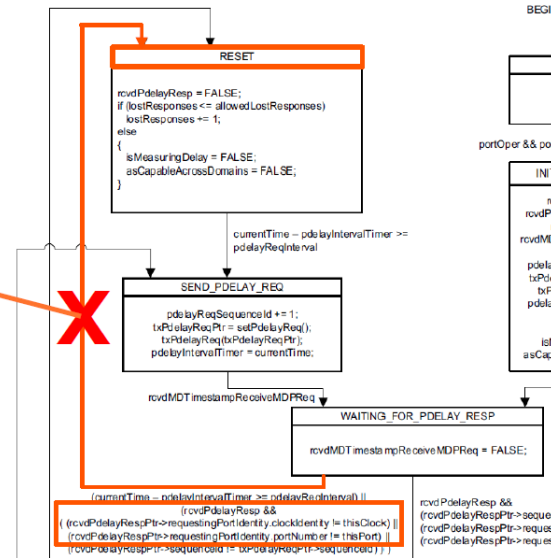
Summary of [1] Georg Janker, et. al

Proposal

Modification of the MDPdelayReq state machine

Proposal:
Modify the MDPdelayReq state machine so that Pdelay_Resp messages with deviating requestingPortIdentity are **ignored** instead of triggering a RESET of the state machine.

Instead, only Pdelay_Resp messages with a matching requestingPortIdentity, but with deviating sequenceId should trigger a RESET.



MDPdelayReq State Machine

Changes reviewed by Geoff Garner

Impact if clockIdentity checks must be in the SM

