# Supporting Our Customer’s Needs for gPTP

Don Pannell, Fellow
Automotive Ethernet Networking, NXP Semiconductor

IEEE 802.1 Interim – September 2023

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

# Disclaimer

- This presentation is the opinion of the presenter only
- It is based on my best recollection of the events that I experienced over the last 25+ years attending and contributing to IEEE 802 and other organizations such as Avnu.org
- I may have made some errors in the exact dates of some events and/or in the specific details of what a given standard supported, etc., but
- Since this presentation is not intended to be technical, these errors, if they exist, will not change the intention nor the message of this presentation

- The intention of this presentation is:
- “Does IEEE 802.1 want to support the PTP needs of our customers even if that means IEEE 802.1AS deviates from being a proper profile of IEEE 1588?”

# OVERVIEW

- History / Background of PTP
- Proposed Solution to the Announce Problem
- Summary & Questions

# History / Background of PTP

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

# IEEE 1588 v1 & v2

- 2002: IEEE 1588 v1 Standard was published
  - The standard supported Profiles to allow targeting specific use cases
  - Major initial use cases were Engineered networks for Telecom & Power plants
  - Circa 2006 an Industrial customer of my company asked us to add their FPGA PTP RTL into our switches
    - This was declined due to the large die area needed for the geometry being used and the purchase volume was too low
- 2008: IEEE 1588 v2 Standard was published
  - "it improves accuracy, precision and robustness but is not backwards compatible with the original 2002 version" – Wikipedia
  - 1588 v2's improvements expanded its customers due to its improvements
  - Integration of PTP H/W was still not done at my company due to its cost, market size, and the different solutions needed to support all the 1588 profiles/options

# IEEE 802.1AS-2011(gPTP)

- 2011: IEEE 802.1AS-2011 Standard was published as a Profile of 1588-2008
  - 802.1AS was developed for the 1st TSN use case: Audio Video Bridging (AVB)
  - A Plug-and-Play profile was needed – but improvements went way beyond
    - Including lower cost product solutions (a need for AVB) by using standard XTALs + software instead of temperature-controlled OSCs, and/or a PLL for the local PTP clock
    - Supported faster GM settling times when BMCA elected a new GM, solved with NRR, etc…
    - Extensive simulations were done to verify its performance would meet the needs – a huge plus!
  - Since this version of PTP was an IEEE 802 standard, and since it didn't have any options, and it supported very low-cost products (thus high volume) AS H/W was added to switches and PHYs at my company and at many other companies as well
    - Only the timestamping H/W was added for AS with some best-effort support for 1588
    - Since a "Bridge" requires a CPU for switch management, this same CPU was used for the gPTP S/W part keeping the BOM costs low for both users of gPTP and those wanting just a switch (i.e., an unmanaged device without PTP as the cost of a CPU was not needed)

# The Attempt at the "Grand Unification"

- Circa ~2012:  1588 v2.1 and AS-Rev (initially ASbt) were started
- 1588 v2.1 completed in 2019, and AS-Rev completed in 2020

- 1588 v2.1 added a lot of AS-2011's cost saving's & performance improvements
- AS-2020 picked up a lot of 1588's support for redundancy which was needed for engineered Industrial and later engineered Aerospace use cases
- Both groups checked with each other for alignment on the new features with the goal (at least from the 802.1AS side) to have common H/W requirements

- Being a H/W silicon architect, I supported this Grand Unification, as did others
- But in my opinion this effort failed in the end
  - For example:  AS-2020's one-step became different from 1588 v2's & thus was not added to much silicon

## Parallel work elsewhere during AS-2020 & 1588-2019 Development

- 2014: Avnu.org published the 1st Automotive Interoperability Specification as customers needed an Engineered version of AS-2011 for Automotive
  - Defined Unidirectional Pdelay, and does not send Announce (as BMCA was dropped)
  - It uses Signaling Messages for a faster start-up, but this is no longer needed
- Year?: AUTOSAR added support for PTP in their S/W
  - Defines Unidirectional Pdelay
  - And does not use Announce, nor Signaling Messages, nor BMCA
  - Defines AS type frame formats as well as AUTOSAR specific formats – not helpful!

- Profiles started in 802.1: 60802 (Industrial), DG (Automotive) & DP (Aerospace)
  - None of these want to use Announce (for sure) and probably not Signaling either

- In my opinion, IEEE 802.1AS became the PTP profile of choice for TSN use cases
- Due to its low cost, verified high performance, certification options, multi-vendor support, public domain PTP stacks, & AS's support for 802.11 (WiFi)

# Where we are now

- Over the last many years 802.1AS has received clear requirements with well defined reasons to prevent the transmission of ALL Announce messages
  - 60802 (Industrial), DG (Automotive) and DP (Aerospace) all need this change
  - 802.1 went to 1588 to get this support – and 802.1 has an action item that is pending:

    https://listserv.ieee.org/cgi-bin/wa?A2=STDS-802-1-MINUTES;abe5ea72.22&S=
  - We did the right thing in asking 1588 but this must NOT be the end of our actions!

- Maybe the following data can help close this action item to 1588:
  - ASds's Unidirectional Pdelay needs to prevent transmission of PdelayReq from TimeTransmitters, which is not against 1588 (see next section)
  - It was shown that adding a single managed object with a simple SM change solves this!

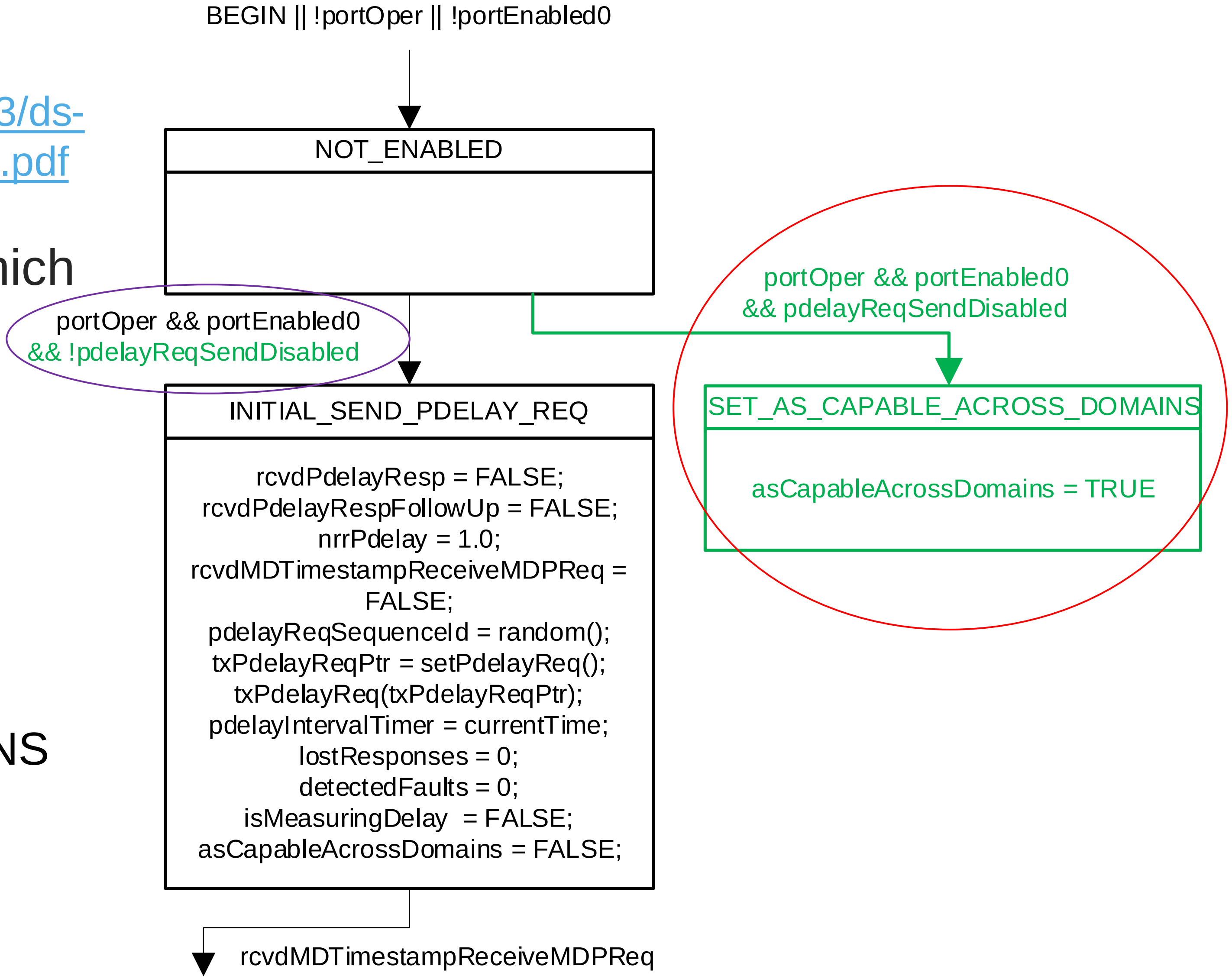
- A similar solution will work for Announce, which is against 1588 (next section)
  - So, this is no longer a technical problem nor a difficult document change problem
  - But it is still an unbounded delay issue as we don't know what will be acceptable to 1588

# Proposed Solution to the Announce Problem

SECURE CONNECTIONS
FOR A SMARTER WORLD

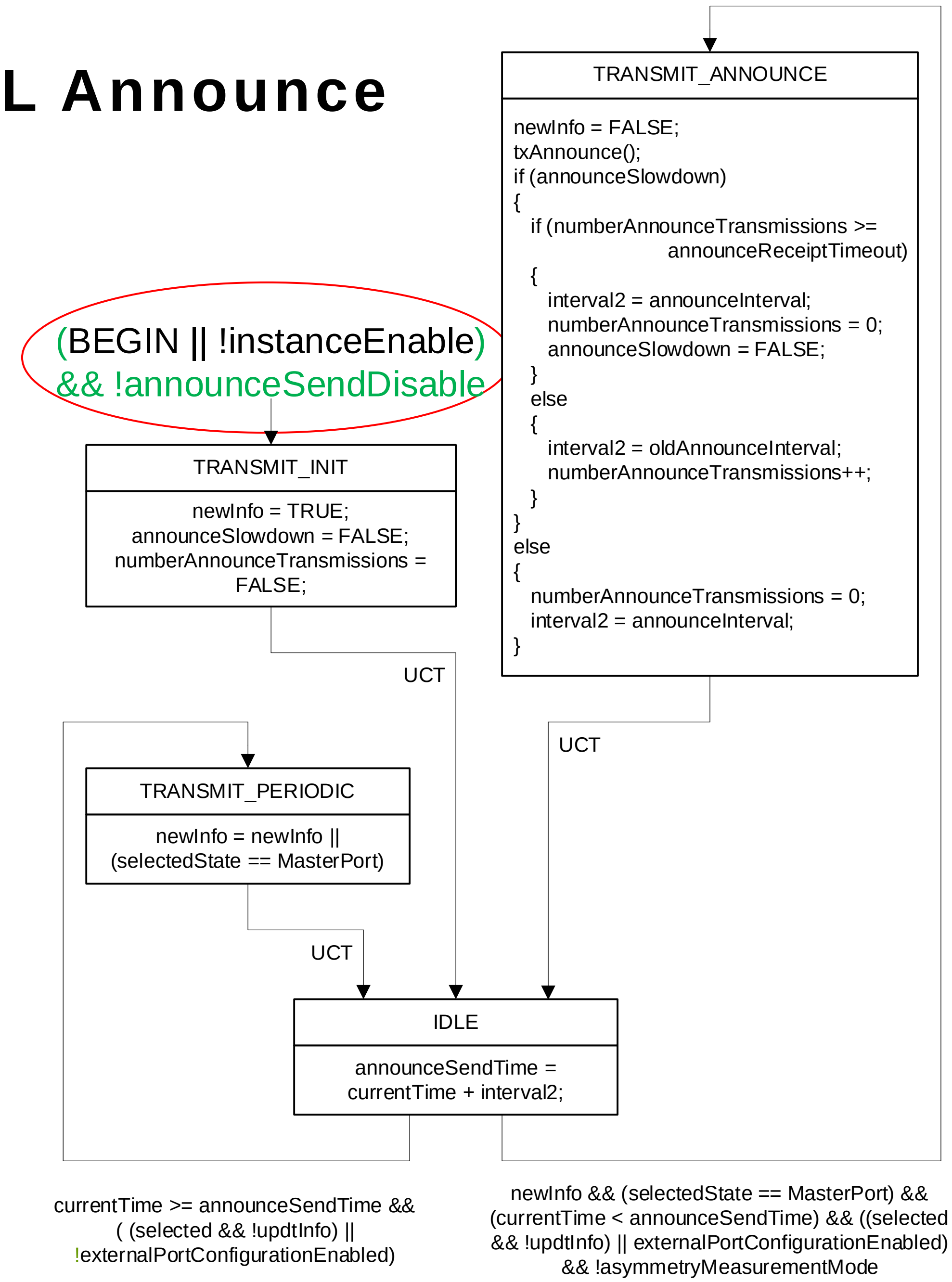
# The Proposed way to Disable PDelayReq Transmissions for ASds

- See: https://www.ieee802.org/1/files/public/docs2023/ds-pannell-proposed-solutions-for-ASds-0923-v02.pdf
- It proposes 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_ACROSS_DOMAINS
- That is the only change needed in the MDPdelayReq SM for this function


BEGIN || !portOper || !portEnabled0
NOT_ENABLED
portOper && portEnabled0
&& !pdelayReqSendDisabled
portOper && portEnabled0
&& pdelayReqSendDisabled
INITIAL_SEND_PDELAY_REQ
rcvdPdelayResp = FALSE;
rcvdPdelayRespFollowUp = FALSE;
nrrPdelay = 1.0;
rcvdMDTimestampReceiveMDPReq = FALSE;
pdelayReqSequenceId = random();
txPdelayReqPtr = setPdelayReq();
txPdelayReq(txPdelayReqPtr);
pdelayIntervalTimer = currentTime;
lostResponses = 0;
detectedFaults = 0;
isMeasuringDelay = FALSE;
asCapableAcrossDomains = FALSE;
SET_AS_CAPABLE_ACROSS_DOMAINS
asCapableAcrossDomains = TRUE
rcvdMDTimestampReceiveMDPReq

# A Proposed way to Disable ALL Announce Transmissions

- Create a new managed object, called announceSendDisabled:
  - When set, this port will not transmit any Announce messages (red oval)
  - As the PortAnnounceTransmit state machine (Fig 10-18) will never start
  - Allows a hard-coded value of ‘1’ to totally remove ALL logic
  - A key requirement of Aerospace
- That is the only change needed in this SM for this function



(BEGIN || !instanceEnable)
&& !announceSendDisable

TRANSMIT_INIT

```
newInfo = TRUE;
announceSlowdown = FALSE;
numberAnnounceTransmissions =
FALSE;
```

UCT

TRANSMIT_PERIODIC

```
newInfo = newInfo ||
(selectedState == MasterPort)
```

UCT

TRANSMIT_ANNOUNCE

```
newInfo = FALSE;
txAnnounce();
if (announceSlowdown)
{
  if (numberAnnounceTransmissions >=
                        announceReceiptTimeout)
  {
    interval2 = announceInterval;
    numberAnnounceTransmissions = 0;
    announceSlowdown = FALSE;
  }
  else
  {
    interval2 = oldAnnounceInterval;
    numberAnnounceTransmissions++;
  }
}
else
{
  numberAnnounceTransmissions = 0;
  interval2 = announceInterval;
}
```

UCT

IDLE

```
announceSendTime =
currentTime + interval2;
```

currentTime >= announceSendTime &&
( (selected && !updtInfo) ||
!externalPortConfigurationEnabled)

newInfo && (selectedState == MasterPort) &&
(currentTime < announceSendTime) && ((selected
&& !updtInfo) || externalPortConfigurationEnabled)
&& !asymmetryMeasurementMode

# Summary & Conclusions

## Summary

- Today, IEEE 802.1AS is the premier PTP solution for TSN & possibly beyond!
- Our customers want & need some simple options, that are easily solvable and don’t require idle logic, and they need us to support them NOW
- Our customers are starting to walk away from AS, will TSN be next?
- There are signs of this already happening (in DG) and we must “stick our finger in the dike to stop the leak before the entire dike breaks and goes away completely”
- In my opinion we need to move forward with an AS solution in parallel with closing the action item with 1588 – as we can’t wait for a new 1588 Std before we start!
- Now is the time to support our customers as the alternative is worse, i.e., AS &/or TSN becoming irrelevant
- 1588 said “Announce can’t be disabled”
- Do we have time to wait for 1588?
- This is clearly NOT a technical issue
- What does the TSN group think?

NXP

SECURE CONNECTIONS
FOR A SMARTER WORLD