Proposed option for ASds

Martin Ostertag

ZHAW Zurich University of Applied Sciences

Basis for the following considerations

- IEEE 802.1ASds PAR lists automotive-in-vehicle and industrial automation as target applications
- ASds d1.0 contains a proposal that resolves the challenges caused by multicast MAC addresses for shared medium segments where the sync tree and the roles of nodes on a link (TT, TR) are predefined (engineered network).
- Earlier contributions [3,4] summarize the goals of P802.1ASds: focus on constrained devices and short time to standard
  - Re-use existing PTP code and make minimum changes to the existing standard
  - Exclude “New, complex mathematics algorithms (like using Sync for rate ratio)”
- At the same time
  - IEEE P802.3da is ongoing to extend 10BASE-T1S to support at least 16 nodes for a segment length of 50m
    - The use of few cascaded 10BASE-T1S segments may be a realistic assumption for, e.g., shaft wiring in elevators
  - P60802 did extensive simulations and error cause analysis for time synchronization and prescribes 125 ms as Sync and pDelay interval
- This presentation addresses concerns related to the two items above, which may or may not be addressed by ASds.
- There is an additional issue that needs to be addressed to 802.3: The TSSI is currently excluded for multidrop T1S

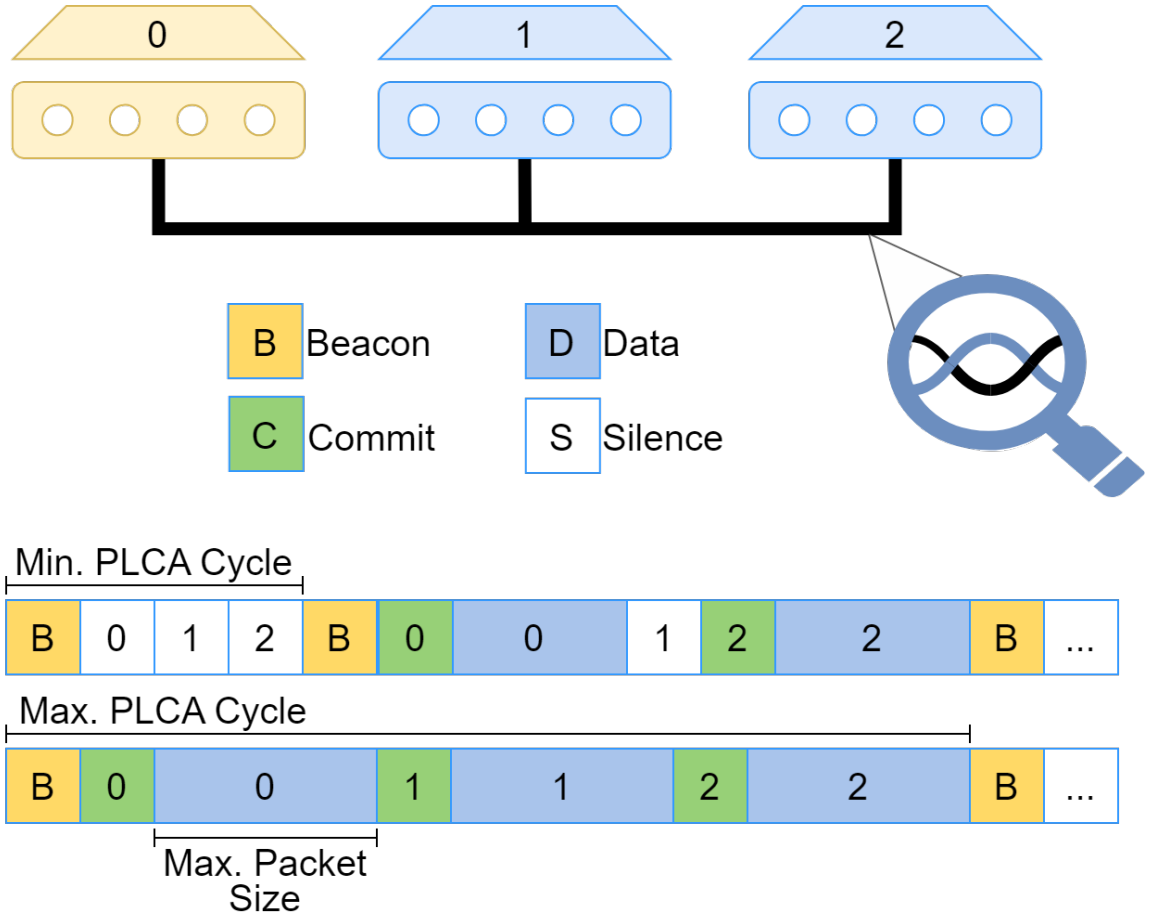
Assumptions used in the following

- pDelay and Sync rates of 125ms (per IEC/IEEE 60802 d2.1)
- 16 nodes on a mixing segment (1 timeTransmitter, 15 timeReceivers per PTP domain)
- MTU 1500 Bytes, C-VLAN tagged
  - Minimum 88 octets / 704 bits; maximum 1542 octets / 12336 Bits including interPacketGap

PLCA operation

- PHYs get statically assigned unique IDs
- The coordinator (ID = 0) sends a BEACON to start a new PLCA cycle
- Transmit opportunities (TO) are granted in sequence, by matching the ID of the node
- A new TO starts at the end of a packet transmission or after a TO is yielded
- Timing

	Bit times	**duration**
T_{Beacon}	20	2.0 µs
T_{TO}	default 32	3.2 µs
	∃ lower limit, cf. IEE 802.3-2022 Eq. 148-1	
$T_{MaxPacket}$	12336	1233.6 µs
$T_{Cycle, Idle}$	20 + 16*32 = 532	53.2 µs
$T_{Cycle, Max}$	20 + 16*12336 = 197396	19739.6 µs



Note: interframe gap and commit times neglected

Message Generation at 60802 rates, per PTP domain

- ASds d1.0 defines:
  - TimeTransmitter sends Sync / Follow-Up and responds to pDelay, TimeReceivers send Pdelay_Req

	Sync	**pDelay**	**Total at 125ms intervals**	
TimeTransmitter (1*)	1 Sync 1 Follow_Up	15 pDelay_Resp 15 pDelay_Resp_F/Up	128 Event Msg 128 General Msg	(8 Sync, 120 pDelay_Resp) (8 Follow_Up, 120 pDelay)
TimeReceiver (15*)	-	1 pDelay_Req	120 Event Msg	(15 * 8 pDelay_Req)
Total messages	1 Sync 1 F/Up	15 pDelay_Resp 15 pDelay_Resp_F/Up 15 pDelay_Req	248 Event Msg 128 General Msg	(8 Sync, 240 pDelay_Req/Resp) (8 Follow_Up, 120 pDelay_Resp_F/Up)

- Observation 1
  - The timeTransmitter
    - Needs **256 transmit opportunities per second** (in average **one TO every 3.9 ms**) and
    - Needs to **retrieve 128 Tx Timestamps per second**
- Observation 2
  - pDelay generates the bulk of the messages

What is pDelay used for?

- meanLinkDelay
  - Is expected to be very stable over time and, in engineered, static systems, even over device restarts.
  - This is exploited in the AVNU AVB profile [7], where meanLinkDelay is persistently stored as *neighborPropDelay* and at startup consistency checks are done.
- NeighborRateRatio (NRR)
  - Is directly only required to correct the upstream link delay. This correction can be neglected.
    - 50 m distance is 250 ns link delay, the error at 100ppm NRR is 25 ps
- RateRatio (RR)
  - RR * ResidenceTime is the dominant error contributor.
    - As observed in the 60802 time sync analysis [6].
  - For the worst-case PLCA cycle, the residence time may be prolonged by up to ~20 ms ($T_{Cycle, Max}$).
    - The error at 100 ppm is 1 µs per 10 ms.
    - The need for a good estimate of RateRatio is undisputable.
- Per AS-2020, RateRatio is the aggregation of the downstream NRRs, so also NRR is still required.

Proposals for discussion

- Keep the current proposal in ASds d0.1 as default.
- Add an option to:
  - **have meanLinkDelay a RW parameter**, and use
  - **syncEgressTimestamp** from Drift_Tracking TLVs in Follow_Ups **to calculate NeighborRateRatio**
    - Not a new complex algorithm, very similar to using pDelay
    - No need to calculate nor evaluate rateRatioDrift (algorithms are outside scope of AS per ASdm)
- This will allow two additional modes of operation
  a) For simple, small systems: Use a fixed value for meanLinkDelay, e.g., 50% of segment length
    - In automotive or in-cubicle applications, the length of a mixing segment will often not exceed 5 - 10 m (25 – 50ns meanLinkDelay).
    - An additional static error of ±10 – 25 ns on the 10BASE-T1S segment could be tolerated in these systems (to be confirmed).
  b) Calibrate / Measure meanLinkDelay during commissioning or at first system start and check for plausibility limits, e.g., ±10 ns, with a single or few pDelay message exchanges at system re-start.
    - Very similar to the AVNU Automotive Ethernet AVB Functional Specification

What would be required?

a) 19.1.2: Add the option to calculate neighborRateRate from syncEgressTimestamp information and use the Drift_Tracking TLV for this.

b) 19.1.2: Add the option to set pDelayReqSendDisabled to TRUE also on the TimeReceiver → no pDelay messages will be exchanged and the link is asCapableAcrossDomains per the current proposal.
   - The AVNU automotive profile uses pDelayInterval = 127 to disable sending pDelayRequest messages by the TimeTransmitter (and there is no need to handle asCapableAcrossDomains with AS-2011).

c) 14.8.33/34/37/38: Optionally set “mgmtSettableComputeMeanLinkDelay” and “mgmtSettableComputeNeighborRateRatio” to FALSE and “useMgmtSettableIsMeasuringDelay” and “useMgmtSettableComputeNeighborRateRatio” to TRUE.
   - CMLDS is not used in ASds, so 14.16.16/17/20/21 can remain unchanged.

d) Make the per-Instance meanLinkDelay (14.8.8) a RW parameter.
   - corresponds to the neighborPropDelay parameter of the AVNU profile if set at start-up.
   - CMLDS meanLinkDelay can remain unchanged.

An issue that needs to be addressed to IEEE 802.3

- IEEE 802.3de-2022 added frame preemption and the TSSI interface to IEEE 802.3 for certain 10BASE-T1x technologies
- However, there is one restriction – no multidrop!

IEEE Std 802.3de-2022
IEEE Standard for Ethernet—Amendment 5:
Enhancements to MAC Merge and Time Synchronization Service Interface for Point-to-Point 10 Mb/s Single-Pair Ethernet

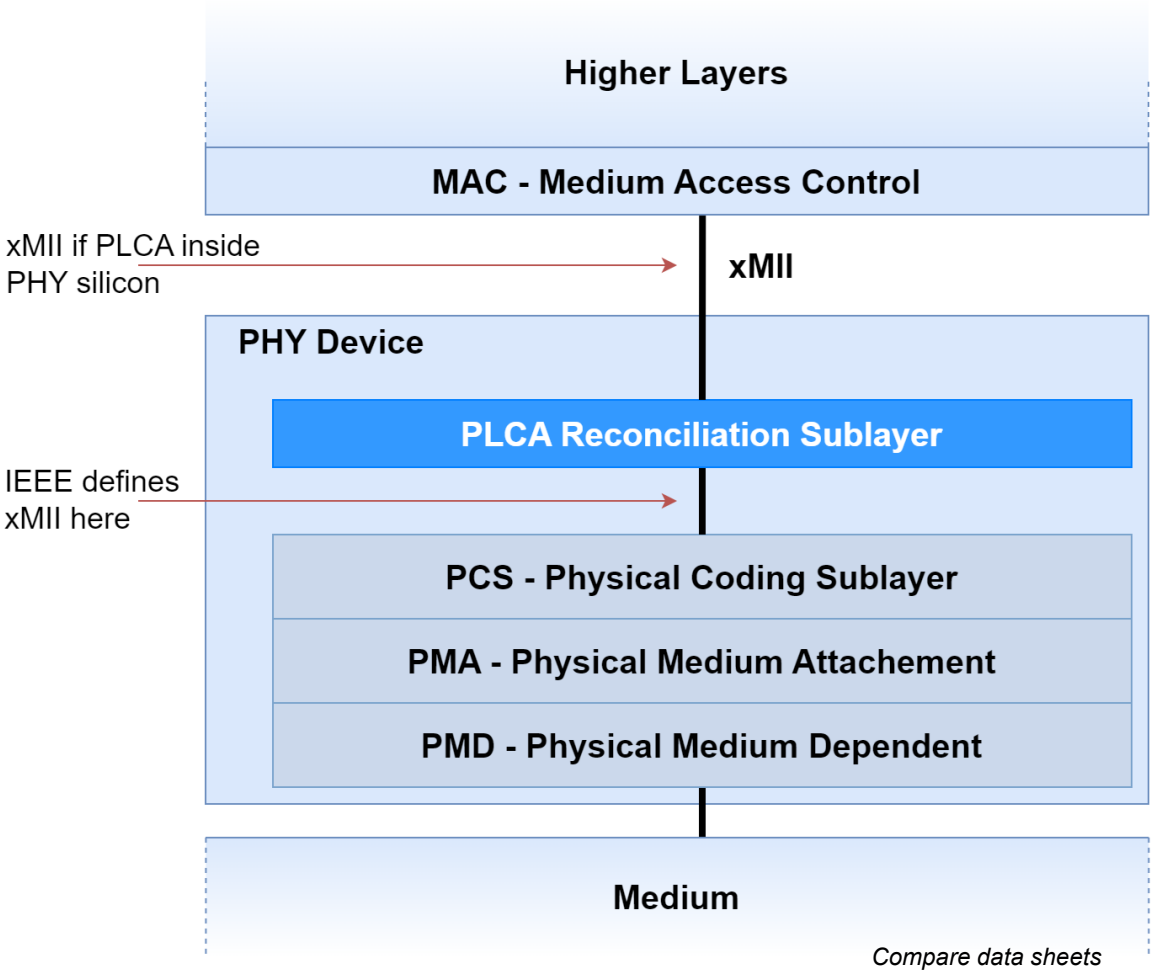
90. Ethernet support for time synchronization protocols

90.1 Introduction

Change the first sentence of the second paragraph of 90.1 as follows:

The TSSI is defined for 10BASE-T1S (see Clause 147) in ~~the~~ full-duplex and point-to-point half-duplex modes of operation, and for other PHY types in full-duplex mode~~only~~.

Backup – Where to implement PLCA and Time Stamping?



Compare data sheets MicroChip LAN867x or CanovaTech CT25205

- Need to support xMII when connecting to a standard MAC implementation
- PLCA timing cannot be made visible to MAC
  - If medium becomes idle (CS low), the MAC will wait for interPacketGap (96 bit times) before transmitting the preamble over the xMII
  - Transmit opportunities in 10BASE-T1S are 32 bit times only, CS will be high again before MAC starts to transmit
- Thus, PLCA is implemented “South of MII” and hidden from the MAC by suitable, transparent buffering
- **Timestamping needs to be done “South of PLCA”, i.e. within the PHY**
  - Else an error of (few µs) up to (few ms) will be introduced
- Combined MAC-PHY devices can do it this way
  - See e.g. LAN8650 using the OPEN SPI interface

References

[1] IEEE P802.1ASds d0.1

[2] IEEE P802.3da PAR

[3] D. Pannell et al, P802.1ASAds – Use Cases & Requirements, 2022-09, https://www.ieee802.org/1/files/public/docs2022/ds-pannell-Avnu-Automotive-UseCase-Requirements-0922-v01.pdf

[4] D. Pannell et al, Proposed Solutions for P802.1ASds, 2023-09, https://www.ieee802.org/1/files/public/docs2023/ds-pannell-proposed-solutions-for-ASds-0923-v02.pdf

[5] IEC/IEEE P60802 d2.1

[6] McCall & Stanton: 60802 Dynamic Time Sync Error – NRR Medians, Algorithms & Analysis Validation https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Stanton-Time-Sync-Error-Model-and-Analysis-0222-v03.pdf

[7] AVNU Automotive Ethernet AVB Functional specification v1.6, https://avnu.org/wp-content/uploads/2014/05/Auto-Ethernet-AVB-Func-Interop-Spec_v1.6.pdf

[8] IEEE Std. 802.3de-2022