

Considerations for IEEE 802.3 PHY Delays in gPTP

Richard Tse

TSN meeting

Sept 28, 2026

Outline

- Reference presentations
- Conceptual considerations
- Practical considerations
- Summary of major points
- What can be done?

Reference presentations

- <https://www.ieee802.org/1/files/public/docs2026/as-Turner-PHYdelay-dot3-0826-V0.pdf>
- <https://www.ieee802.org/1/files/public/docs2026/as-Turner-PHYdelay-dot3-0526.pdf>
- <https://www.ieee802.org/1/files/public/docs2026/as-Turner-PHYdelay-dot3-0926-V0.pdf>

Conceptual considerations 1/2

- IEEE 802.3/802.3cx/802.3de/802.3df:
 - Use the MDI as the reference plane for timestamping, per IEEE 802.1AS/1588
- IEEE 802.3 PHY managed objects (e.g., registers, datasheet values) can enable functions outside of a PHY to compensate for the PHY's static delay
 - The PHY delay covers the functions between the xMII and the MDI
 - PHY functions that have mirrored Tx/Rx dynamic delays (e.g., interleaving/deinterleaving, multi-lane distribution/merging) are represented by static delay values, per IEEE 802.3cx
 - Tx side uses the maximum intrinsic delay of the function
 - Rx side uses the minimum intrinsic delay of the function
 - Instantaneous functional delays (e.g., idle insertion/removal) in a PHY are not given in any of its managed objects
 - Might be automatically compensated by the PHY and its corresponding TimeSync Client (90.4.1.2 of IEEE 802.3cx)
 - Might contribute to timestamping uncertainty
 - Implementation-specific PHY delays must be constant since the xMII to xMII delay across a link is constant

Conceptual considerations 2/2

- Other functions (higher layer H/W, S/W, etc) could exist between the 802.1AS reference plane and timestamp measurement plane, depending on where the timestamp capture is performed in an 802.1AS implementation
 - These delays are not covered by the corresponding 802.3cx managed objects (e.g., delay registers, datasheet values)
 - These delays could be covered by implementation-specific registers/datasheet values or could be automatically compensated for by an implementation

Practical considerations 1/4

- Types of delays in a PHY
 - Physical delays (e.g., wire delays, logic ramp-time delays) are static or pseudo-static
 - Intrinsic functional delays of some standardized functions (e.g., functions with mirrored Tx/Rx delays like interleaving/deinterleaving, lane distribution/merging) are static, per IEEE 802.3cx
 - Implementation-based functional delays (e.g., flip-flop pipeline stages, FIFOs) are static or pseudo-static
 - Functional delays of some standardized functions (e.g., idle insertion/removal) change instantaneously
- PHY delays are (typically) given in values that are referenced to an ideal timebase or assume the PHY's clock is ideal
 - 1.3ns logic ramp delay = 1.3ns TAI delay
 - 3 clock-cycles of delay with a 1GHz clock = 3ns TAI delay

Practical considerations 2/4

- Should PHY delays be included in link delay equations?
 - IEEE 802.3 considers the MDI as its reference plane for timestamping, per IEEE 802.1AS
 - The Tx+Rx PHY delays in the two directions are, often, significantly different
 - Different devices with different implementations at each PTP Port
 - Different start-up conditions at each PTP Port
 - The PHY delay asymmetry cannot be compensated for in link delay equations
 - The Tx and Rx delays of the PHY at one PTP Port are not known by the other PTP Port
 - Instantaneous delays (e.g., Idle insertion/removal) cannot be known outside a PHY and its associated TimeSync Client
 - The clock(s) used by a PHY might not be related to any gPTP clock
- **No, PHY delays should not be included in link delay equations. Each PTP Port should compensate for its own PHY delays.**

Practical considerations 3/4

- Should PHY delays be multiplied by some rateRatio value?
 - Estimated PHY delay is (typically) based on an ideal clock (e.g., TAI)
 - Actual Tx PHY delay is (typically) based on its line clock
 - Line clock might not be related to any gPTP local clock
 - Actual Rx PHY delay is (typically) based on its recovered line clock (i.e., same as the line clock of the Tx PHY at the other end of the link)
 - A PHY implementation could use multiple clocks
 - **To determine the actual PHY delay in the gPTP timebase, each PHY could multiply its own TAI-based delay estimate by the offset ratio between its line clock rate (and other clock rates, if present) and the GM's clock rate**
 - A PHY's line clock is not part of the gPTP specification but could be described as a non-normative concept

Practical considerations 4/4

- What timing accuracy is required for an application?
 - A PHY with a (very large) $10\mu\text{s}$ TAI delay estimate with a line clock running at 100ppm offset has a delay estimation error of 1ns
 - Is this sufficiently accurate for the given application?
 - A PHY with a (more typical) 100ns TAI delay estimate with a line clock running at 100ppm offset has a delay estimation error of 10ps
 - This should be sufficiently accurate for (most/all?) current applications
- Example line clock frequency offset limits (relative to TAI):
 - $\pm 100\text{ppm}$ for general Ethernet applications
 - $\pm 4.6\text{ppm}$ for telecom applications
 - $\pm 50\text{ppb}$ for mobile radio applications

Summary of major points

- IEEE 802.3 and IEEE 802.1AS are consistent with the MDI being “the reference plane”
- IEEE 802.3 PHY delay covers the time between the xMII and the MDI
- PTP Ports should handle their own PHY delays to get accurate MDI-based timestamps that are then used in gPTP link delay equations
 - PHY delay estimates are typically referenced to an ideal timebase (e.g., TAI)
 - PHY delays at one PTP Port are not known by any other PTP Port
 - Even if static PHY delay values could be transferred between PTP Ports, instantaneous PHY delay changes are still problematic
 - PHY implementations could use one or more clocks with timebases that are not related to gPTP
- PHY delay estimation errors
 - Could be reduced by migrating the PHY delay estimation values to the GM’s timebase
 - Might be insignificant for the application and, in these cases, could be ignored

What can be done?

- Since there are many functional specifications and implementation styles for PHYs, there does not appear to be any universal solution that resolves PHY delay estimation errors for IEEE 802.1AS
- **An informative annex could be added to IEEE 802.1AS:**
 - Refer to relevant IEEE 802.3 specifications on PHY delays
 - Explain how PHY delay estimates could add error to timestamps
 - Show the relevant clocks and timebases that are present in example PHY implementations
 - Give examples of the potential magnitude of these errors (size of PHY delay, ppm offsets of relevant clocks)
 - Give examples of how these errors could be reduced

Questions?