# Timestamping with Transmitter Skew: Follow-Up and Proposed Amendments

A Leading Provider of Smart, Connected and Secure Embedded Control Solutions

**Andras de Koos**

October, 2020

SMART | CONNECTED | SECURE

# Recall (1):

The problem of transmitter skew was first brought up in the September 2020 802.3cx meeting in dekoos_3cx_01_0920.pdf

- Though transmitter skew is allowed by 802.3, there is no explicit mention of its effect on timestamping.
- An inaccurate transmit timestamp (or more precisely, inaccurate end-to-end accounting of the total latency) leads to PTP time synchronization error.

Any updates to 802.3 should make explicit recommendations about transmitter skew for multilane Ethernet interfaces in order to increase time synchronization accuracy.

# Recall (2):

There was general agreement that any multilane PMD implementation should aim for zero transmit skew, thus circumventing any timestamp inaccuracy problems.

- Updating the standards to this effect is straightforward.

Clarifying how to timestamp in the presence of transmit skew is not nearly as simple. In general, Tx skew may add to, or subtract from, the skew of the medium.

- Appropriate to timestamp with respect to the skew mid-point, although it is rarely optimal.

In specific cases, Tx skew is strictly additive to the skew of the medium.

- Optimal to timestamp with respect to the last-departing lane

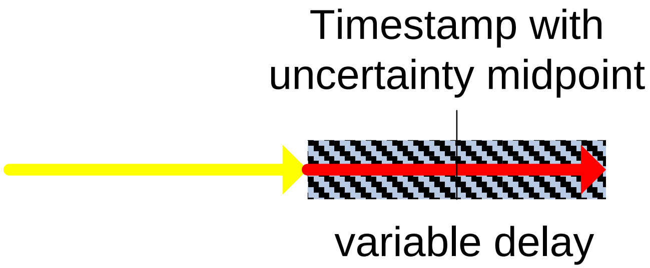
This phenomenon was discussed in the October 2020 ad-hoc meeting:
dekoos_3cx_01_1020.pdf

Updates to 802.3 should thus specify what to do in the general case, but open the door for implementations to optimize for their specific system context.

# Tx Skew Midpoint – Unmeasured path delay

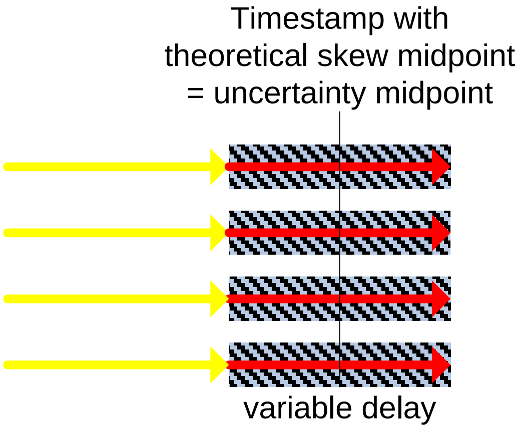
Using the Skew midpoint as the timestamp point is conceptually similar to timestamping with respect to the middle of the uncertainty window for a single-lane Ethernet PMD:

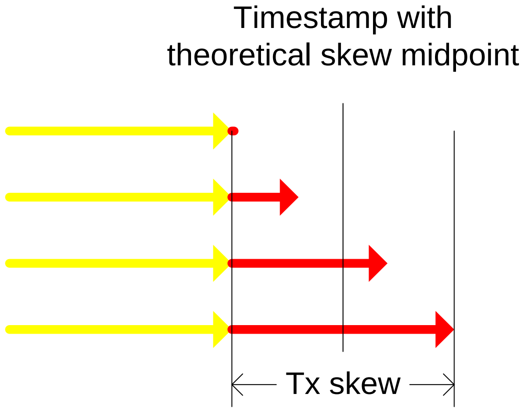
Assuming there is an unmeasured delay associated with a lane, the timestamp would generally be in the middle of the known uncertainty window, in order to minimize the total timestamp error.




For a multilane interface, the skew can be seen as the midpoint between the minimum and maximum transmit delays for each lane. *If there are no other skew contributors*, then the maximum skew between the lanes is exactly equivalent to the range of the single-lane uncertainty.

As such, if there is an *unmeasured* transmit path delay uncertainty for each lane, then the time synchronization error on a multi-lane PMD from timestamping at the skew midpoint is no worse than the time synchronization error on a single-lane PMD.






MICROCHIP

# Proposed updates to skew sections

- **In every skew sub-clause (e.g. 80.5, 116.5, 131.5), there should be a mention that the transmit skew should be kept to a minimum for the purposes of timestamp accuracy.**
  - It is already noted that in the case of discrepancies with the sub-clauses, the sub-clauses have precedence. But still worth pointing out for timestamping.
  - Should have a pointer to 90.7.

For instance, in section 80.5:

**Table 80–6—Summary of Skew constraints**

| Skew points | Maximum Skew (ns)[a] | Maximum Skew for 40GBASE-R PCS lane (UI)[b] | Maximum Skew for 100GBASE-R PCS lane (UI)[c] | Notes[d] |
|---|---|---|---|---|
| SP0 | 29 | N/A | ≈ 150 | See 83.5.3.1 |
| SP1 | 29 | ≈ 299 | ≈ 150 | See 83.5.3.2 |

The following note ([f]) should be added for SP0 and SP1:

[f] For Ethernet interfaces where timestamping is enabled, transmit skew can contribute to time synchronization error. For accurate time synchronization, the transmit skew should thus be minimized, ideally to zero. See 90.7.

# Proposed Update to Clause 90.7 – new Paragraph and Note

Note the existing paragraph for receive skew:

The receiver of a multi-lane PHY is expected to include a buffer to compensate for skew between the lanes. This buffer selectively delays each lane such that the lanes are aligned at the buffer output. The earliest arriving lane experiences the most delay through the buffer and the latest arriving lane experiences the least delay through the buffer. The receive path data delay for a multi-lane PHY is reported as if the beginning of the <message timestamp point> arrived at the MDI input on the lane with the smallest buffer delay.

The following proposed text for the transmit skew mirrors the receive paragraph above, and would be placed after the receive paragraph:

Lane skew can be present on a multilane transmitter when PCS/FEC lanes have different static latencies such that their alignment markers appear staggered as they depart the device at the MDI output. Since transmit skew in series with medium skew is not strictly additive, transmit skew can contribute to time synchronization error by obscuring the actual latency of the medium. Transmit skew should thus be minimized, ideally to zero. If the transmit skew is not zero, then it is recommended[4] that the transmit path delay for a multi-lane PHY be reported as if the message timestamp point departed the MDI output on an imaginary lane whose departure time is the *midpoint* between those of the first-departing and last-departing lanes. This has the effect of centering any timestamp error in the middle of the skew window.

[NOTE 4] For specific cases when transmit skew is strictly additive to any medium skew, it is appropriate to report the transmit path delay as if the message timestamp point departed the MDI output on the *last-departing lane*.

# Editorial Notes

1. The “imaginary” wording for the skew midpoint is awkward in the proposed 90.7 text.

“… departed the MDI output on an imaginary lane whose departure time is the *midpoint* between those of the first-departing and last-departing lanes.”

This was an attempt to mirror the wording of the Rx skew paragraph, which specified that the timestamp be done with respect to an actual lane:

“… arrived at the MDI input on the lane with the smallest buffer delay.”

Suggestions to improve the Tx skew midpoint description are welcome.

2. The proposed Tx Skew paragraph for 90.7 contains background information. Typically, 802.3 specifies what to do, but rarely the rationale behind the directives.

“Since transmit skew in series with medium skew is not strictly additive, transmit skew can contribute to time synchronization error by obscuring the actual latency of the medium.”

Though the background information is valuable, its inclusion may seem out-of-place in 802.3, and reflects this author’s preference for verbosity.

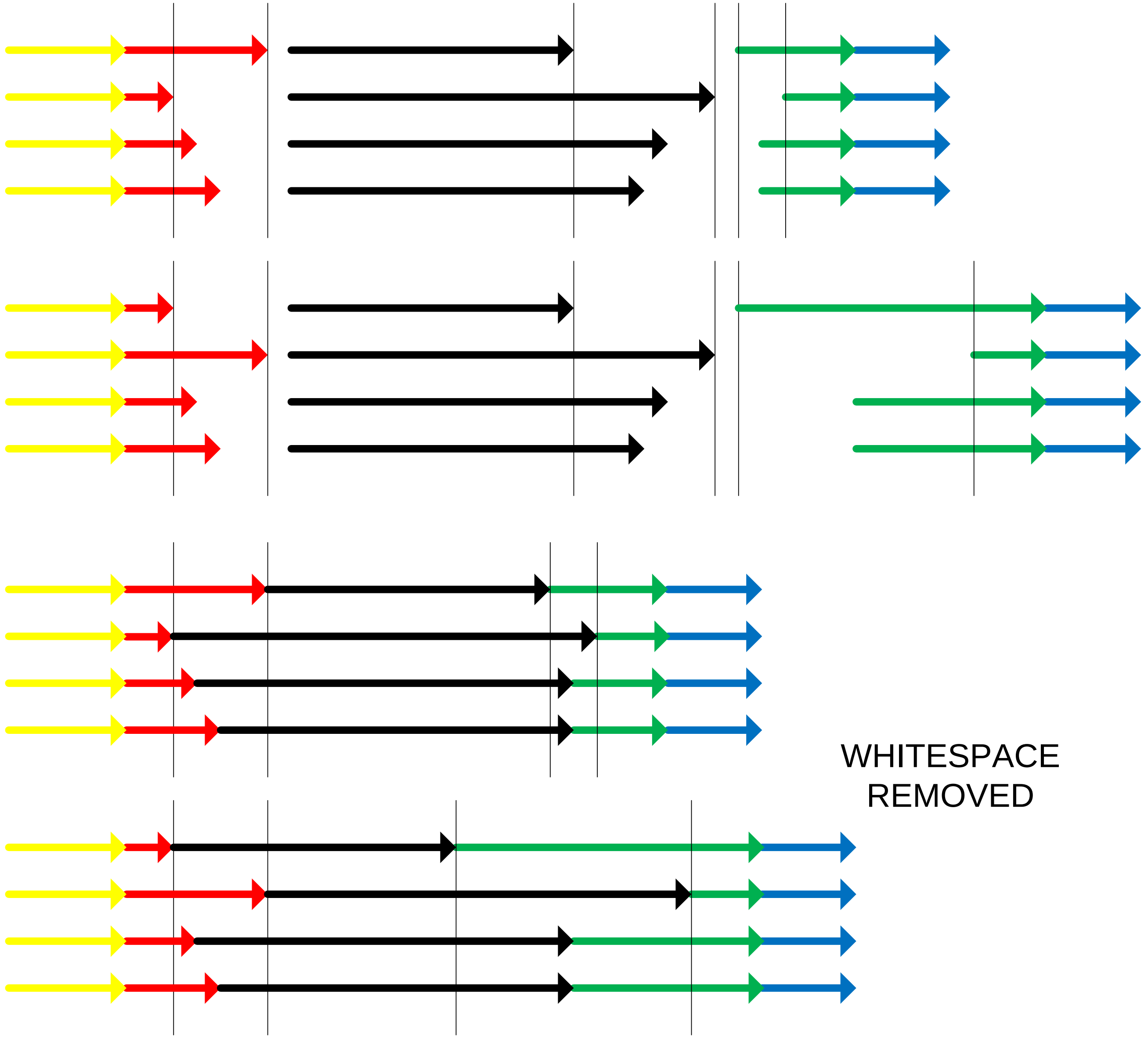
Any and all editorial comments are welcome.

SMART | CONNECTED | SECURE

# Thank You

MICROCHIP

# Appendix A: TxSkew in series with Medium Skew




Accounting for the transmitter skew is not simple.

- Transmitter skew in series with medium skew may cancel out, or may be additive.
- A full accounting is not possible without knowing the transmitter latency of each lane, and associating it with the latency of each lane of the medium.

- Timestamping on the *last departing lane* is optimal in specific cases, but not generally.
- In the general case, timestamping at the *midpoint of the first-departing and last-departing lanes* will yield the smallest maximum error.

# Appendix B : PCS/FEC lane skew vs PMA/PMD lane skew

- One such case where it is appropriate to timestamp on the last departing is where the same PCS skew exists on every PMA lane.
  - To use the example of a 100GE-R4: the 5 PCS lanes with each PMA lane can be skewed – i.e. the first bit of the 5 alignment markers within each PMA lane might not be adjacent to one another.
- In this case, the PCS skew will be *strictly additive* to any skew on the PMA/PMD lanes. As such, timestamping with respect to the last departing PCS lane is appropriate.
- Meanwhile, the transmitter PMA lane skew is not strictly additive to the skew of the medium.

- Explaining all this in 90.7 would be long, complicated, confusing, and ultimately would not give a clear directive as to how to deal with the Tx skew.
- Also, it is not clear how common the above example is, nor whether there are caveats, nor whether the examples presented for timestamping with respect to last-departing lane are exhaustive.
- **Better to *recommend* timestamping with respect to the transmit skew midpoint, leaving it to implementers to determine whether last-departing is more appropriate in their specific context.**