Timestamping with Transmitter Skew Follow-up

MICROCHIP

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

Andras de Koos

October, 2020

SMART | CONNECTED | SECURE

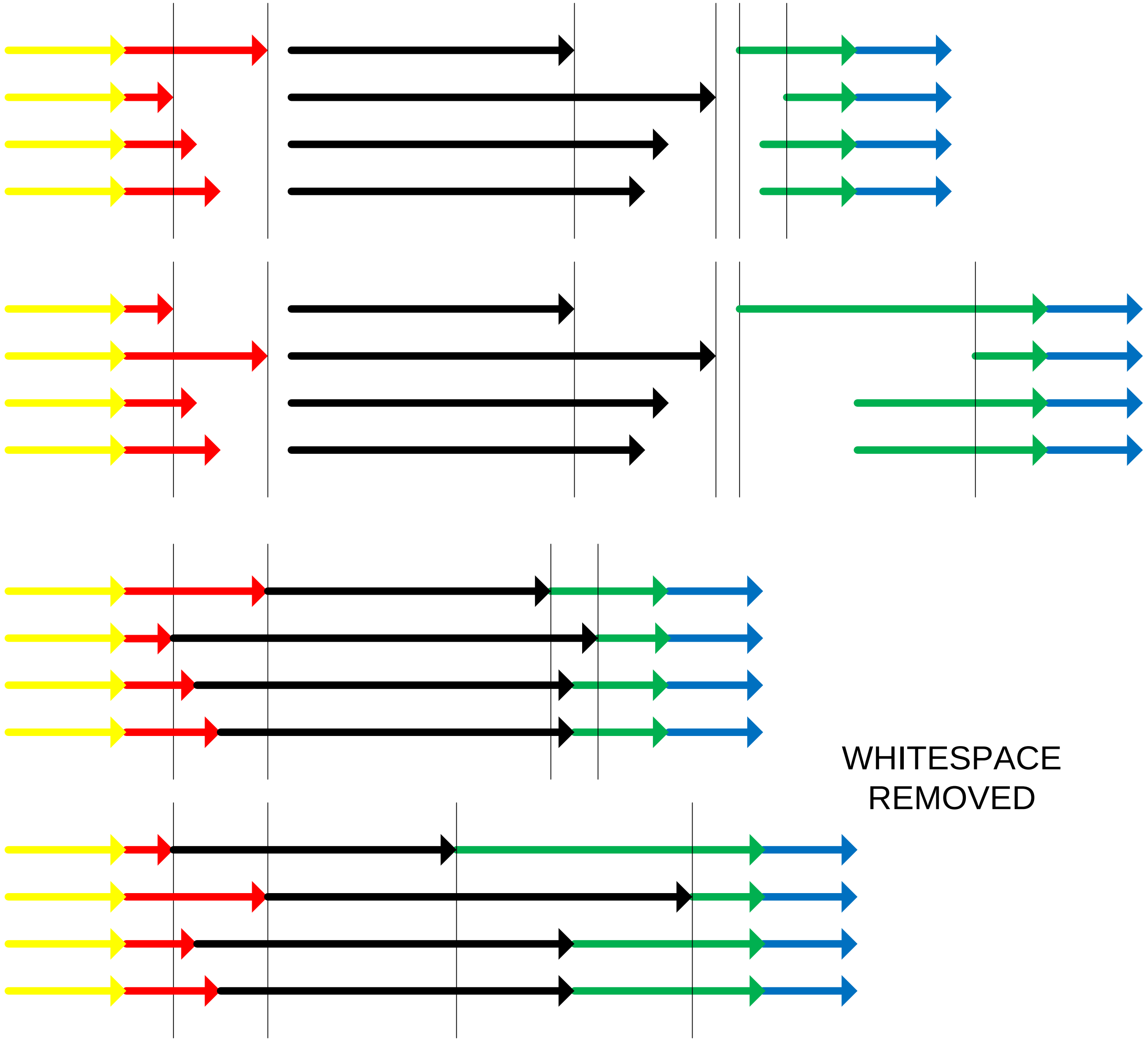
Recall (1):

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.

Recall (2):



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 associated with each lane, and associating 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.

Goals for updated text in 802.3

After discussion in September’s meeting, it was agreed that any update to 802.3 should specify that transmitter skew is not desirable for multilane interfaces where timestamping takes place. Any transmitter implementation should therefore aim for zero skew. When there is no transmitter skew, the problem disappears.

The more difficult subject to address is how to timestamp when the skew is not zero.

- Transmit timestamping with respect to the mid-point between first-departing and last-departing lanes is okay – centers the error.
- But there are cases where the mid-point is not optimal, and that last-departing would be better.

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.**

Proposed Update to Clause 90.7 – new Paragraph and Note

The following mirrors the receive skew paragraph:

Lane skew can be present on a multilane transmitter when PCS 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 exact 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.

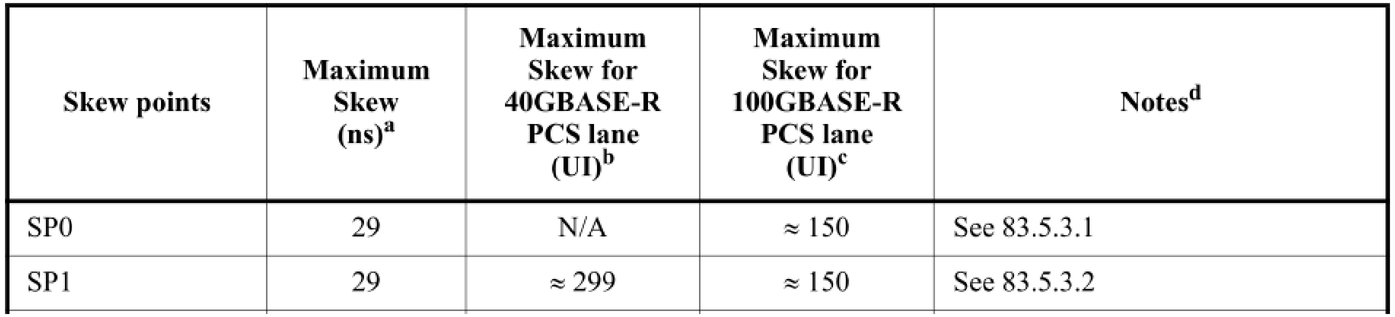
[NOTE 4] In certain cases, such as when the skew of the medium is known to be zero, it is appropriate to report the transmit path delay as if the <message timestamp point> departed the MDI output on the *last-departing lane*.

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 limited 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.

SMART | CONNECTED | SECURE

Thank You