# Summary of Contributed Solutions


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


Richard Tse

IEEE 802.3cx Teleconference July 22, 2020

# Main Identified Issues

- **Inconsistent Message Timestamp Point**
  - Start of SFD vs start of symbol after SFD
- **Path Delay Variance from Idle Insert/Delete for Alignment Markers**
- **Multi-PCS lane distribution**
  - Multi-lane delay architecture needs to be more tightly defined
  - Variable delays for xMII-to-multi-lane distribution and multi-lane-to-xMII recombination need to be accounted for
  -

# Summary: Message Timestamp Point Soln

| Proposed Solution | Pros | Cons | Comments |
|---|---|---|---|
| Use start of symbol after SFD for themessage timestamp point | Consistent with parent applications, IEEE 802.1AS and IEEE 1588v2 | Errors are introduced for some legacy implementations that use start of SFD as the message timestamp point | Possible workarounds for incompatible legacy implementations:<br><br>Applications without AM and FEC can be resolved by adding a 1-byte time offset to each timestamp.<br><br>AM functions can cause large but infrequent errors. These infrequent errors could be filtered away by time recovery algorithms, which have a low-pass filter.<br><br>FEC can cause frequent large timestamp errors (equal to one FEC block). Time recovery algorithms could detect and adapt for errors equal to the time of one FEC block. |

- We need to pick one message timestamp point or the other. The errors are the same for both.
- We should pick the one that is compatible with the parent applications for 802.3 timestamping - IEEE 802.1AS and IEEE 1588v2. These both use the start of the symbol after the SFD as their message timestamp point.

# Summary: Idle insert/delete for AMs Soln

| Proposed Solution | Pros | Cons | Comments |
|---|---|---|---|
| PHY data delay is adjusted to account for AM insertion/removal and its corresponding Idle rate adaption | Literally consistent with IEEE 1588 timestamping<br><br>Compatible with many existing implementations | | There seems to be general agreement on this solution.<br>• http://www.ieee802.org/3/maint/public/gorshe_1_0119.pdf<br>• http://www.ieee802.org/3/ad_hoc/ngrates/public/calls/19_0416/nicholl_nea_01_190416.pdf<br>• http://www.ieee802.org/3/ITSA/public/jan20/parkholm_itsa_01_0120.pdf<br>• http://www.ieee802.org/3/ITSA/public/jan20/tse_itsa_02_0120.pdf<br>• http://www.ieee802.org/3/ad_hoc/ngrates/public/calls/19_0416/nicholl_nea_01_190416.pdf<br>• http://www.ieee802.org/3/cx/public/april20/bordogna_3cx_01_0420.pdf |

- No opposing contributions have been received for this issue
- Several contributions for normative text have been received. We need to pick the best parts from all of these contributions.

# Summary: Multi-PCS Lane Distribution Soln #1

| # | Proposed Solution | Pros | Cons | Comments |
|---|---|---|---|---|
| 1 | "Method 1" and "Option A" from http://www.ieee802.org/3/cx/public/april20/tse_3cx_02a_0420.pdf<br><br>Define Tx lane distribution so PHY delay is constant and each lane transmits its block at a different time<br><br>Rx lane deskew function naturally compensates for intrinsic Tx skew<br><br>Rx lane multiplexing time is variable. Each Rx lane has a unique delay. | Literally congruent with IEEE 1588 timestamping rules<br><br>Tx lane distribution delay is constant | Rx lane multiplexing time must be accounted for on each message<br><br>Some deskew FIFO capacity is used to compensate for intrinsic Tx lane distribution delay | This solution's methodology cannot apply to other variable delay PHY functions. It is specific to this multi-lane PCS function.<br><br>Because each PHY measures its delay, should be compatible with split PHYs. |

# Summary: Multi-PCS Lane Distribution Soln #2

| # | Proposed Solution | Pros | Cons | Comments |
|---|---|---|---|---|
| 2 | "Method 1" and "Option B" from http://www.ieee802.org/3/cx/public/april20/tse_3cx_02a_0420.pdf<br><br>Define Tx lane distribution so all lanes transmit their blocks at the same time. Each Tx lane has a unique delay.<br><br>Rx lane multiplexing time is variable. Each Rx lane has a unique delay. | Literally congruent with IEEE 1588 timestamping rules | Tx lane distribution time must be accounted for on each message<br><br>Rx lane multiplexing time must be accounted for on each message | This solution's methodology cannot apply to other variable delay PHY functions. It is specific to this multi-lane PCS function.<br><br>Because each PHY measures its delay, should be compatible with split PHYs. |

# Summary: Multi-PCS Lane Distribution Soln #3

| # | Proposed Solution | Pros | Cons | Comments |
|---|---|---|---|---|
| 3 | "Method 2" and "Option C" from http://www.ieee802.org/3/cx/public/april20/tse_3cx_02a_0420.pdf<br><br>Define Tx lane distribution so all lanes transmit their blocks at the same time<br><br>Define Tx lane distribution time as a constant value, M<br><br>Define Rx lane multiplexing time as a constant value, N<br><br>M + N = intrinsic constant delay of both the Tx and Rx multi-PCS lane operations | Conceptually congruent with IEEE 1588 timestamping rules<br><br>Consistent with how FEC delays are dealt with in 802.3<br><br>Because the delays are treated as constants, 802.3 delay registers can be used to record their values (M and N) | Literally incongruent with IEEE 1588 timestamping rules | This generic solution works for all variable delay functions that have mirrored Tx and Rx delays that sum to a constant value. It simplifies the estimation of the delay for a single PHY function or for multiple cascaded PHY functions.<br><br>Should be compatible with split PHYs as long as each Rx PHY produces an output that is identical to its corresponding Tx PHY's input. |

# Summary: Multi-PCS Lane Distribution

- Soln #3
  - Might enable the simplest implementation because it eliminates the need to track every message's datapath and corresponding delay through the Tx and/or Rx PHY
  - Follows the methodology used by IEEE 802.3 for FEC and can be applied generically for one or many cascaded PHY functions that have variable intrinsic delays and should work for many (most?) new PHY functions that might be defined in the future

# Conclusions

- Can we make decisions to move ahead for any of these 3 issues?
  - Decide what normative text is needed
  - Decide what informative text is needed
  - Possibly select from or modify existing contributions
- Can we define the generic methodology that should be used for all PHY functions that have varying Tx and Rx intrinsic delays that always sum to a constant value?
  - Note that implementation-specific delays (e.g. start-up delays, process delays) must be accounted for separately, but, for PHY functions with this type of delay, it must be a constant value so they should be “easy” to determine

# Questions?

# Thanks!