Proposed Annex for 802.3cx

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

Richard Tse

IEEE 802.3cx Teleconference Jan 19, 2021

Proposed Annex for 802.3cx

- **Annex would explain:**
  - The purpose and effects of the 802.3cx specifications
  - How to use the 802.3cx specifications
  - How repeating and mirrored variable delays can be accounted for
- **An outline and some details for this annex are given in this presentation**
- **Text for this annex to be developed after its contents are agreed upon**

MICROCHIP

Outline of Proposed Annex

1. **AM/CWM and its corresponding Idle insertions and deletions**
   - Time error effects
   - Use of *num_blk_change signals and path data delay register bits
2. **Multi-PCS lanes**
   - Time error effects between 802.3cx compliant implementation and example legacy implementations
   - Use of path data delay register bits
3. **Message timestamp points**
   - Time error effects
   - Compensation between the two options
4. **Tx skew**
   - Time error effects of non-zero Tx skew
5. **Generic path data delay mechanism for functions with variable delays**

MICROCHIP

Timing Error Summary

Ethernet Rate	Path Data Delay Variation per Tx/Rx Interface (ns)				Total TE per Tx or Rx Interface (ns)	**Path Data Delay Variation Contribution to Max|TE|, per PTP Boundary Clock (ns)**
	mismatched message timestamp point	Idle insert/remove (per Idle)	AM/CWM insert/remove	PCS Lane Distribution		
GE	8	16	N/A	N/A	24	48
10GE	0.8	3.2	N/A	N/A	4	8
25GE	0.32	1.28	2.56	N/A	4.16	8.32
40GE	0.2	1.6	6.4	4.8	13	26
100GE	0.08	0.64	12.8	12.16[1]	25.68	51.36
200GE	0.04	0.32	2.56	2.24[1]	5.16	10.32
400GE	0.02	0.16	2.56	2.4[1]	5.14	10.28

1. **Error given in table is for potential PCS lane distribution delay error. How FEC lane distribution delay variation is dealt with is already defined in 802.3bf.**

MICROCHIP

AM/CWM: Time Error

- Sizes of the timing errors that can result from non-802.3cx compliant implementations are shown on slide 3
- Time error occurs because path data delay change in Tx PHY and in Rx PHY are not mirrors of each other
- Timing errors occur only for PTP messages whose path data delay is modified by this function
  - Random probability of occurrence depends on:
    - Probability of a AM/CWM event occurring
    - Probability of corresponding Idle insertions or deletions occurring
    - Probability that any packet is vulnerable to having its path data delay affected by these occurrences in a particular implementation
  - If probability is sufficiently small, a time recovery algorithm might filter the erroneously timestamped messages
  - Some implementations do not transmit PTP messages in the region of AM/CWM insertions and its corresponding Idle deletions
    - Such an implementation avoids the error condition on its Tx port

MICROCHIP

AM/CWM: Example usage of Tx/Rx_num_blk_change and PCS path data delay register bits (1/2)

- Recall that the Tx_num_blk_change signal is used by TSSI model to make it "appear as if the AM or CWM insertion and the corresponding rate adaptation had been performed before the Tx xMII".
- This allows constant values in the 802.3 Tx PCS path data delay registers to continue to be used, with high accuracy, for the PCS delay.
- Same applies for Rx_num_blk_change signal

MICROCHIP

AM/CWM: Example usage of Tx/Rx_num_blk_change and PCS path data delay register bits (2/2)

The usage is illustrated by the scenarios below:

1. Without AM, CWM, or corresponding rate adaptation

- Word arrives at Tx xMII at time = T1
- PHY path data delay = PDD
  - The constant value, PDD, is programmed into the 802.3 Tx PCS path data delay registers
- Calculated Tx departure timestamp = T1 + PDD

2. With AM, CWM, or corresponding rate adaptation

- Word arrives at Tx xMII at time = T1
- PHY path data delay with AM, CWM, or corresponding rate adaptation = PDD + Tx_num_blk_change*(nanoseconds/block)
  - The PHY's delay changes due to AM, CWM, or corresponding rate adaptation events
- Calculated Tx departure timestamp = T1 + PDD + Tx_num_blk_change*(nanoseconds/block)

3. This is how tx_num_blk_change is used in an 802.3cx compliant implementation to account for the delay variation

- Adjusted word arrival time at Tx xMII = T1 + Tx_num_blk_change*(nanoseconds/block)
  - The arrival time at the Tx xMII is modified to reflect the AM, CWM, or corresponding rate adaptation events (as if they happened before the Tx xMII)
- PHY path data delay = PDD
  - The constant value, PDD, programmed into the 802.3 Tx PCS path data delay registers does not change.
- Calculated Tx departure timestamp = T1 + Tx_num_blk_change*(nanoseconds/block) + PDD

MICROCHIP

Multi-PCS Lane: Time Error and PCS path data delay register bits

- The sizes of the potential timing errors that can result from non-802.3cx compliant implementations are given on slide 3
  - Example implementations are described
    - “Option A + method 1”
    - “Option B + method 1”
  - Timing errors from interaction of these different implementations are as per https://www.ieee802.org/3/cx/public/july20/tse_multilane_TE_analysis.xls
- Use of PCS path data delay register bit values
  - Tx uses max PCS lane distribution delay constant value (i.e., the Tx PCS lane distribution delay for lane 0)
  - Rx uses min PCS lane merging delay constant value (i.e., the Rx PCS lane merging delay for lane 0)
  - Usage is the same as for FEC lanes

MICROCHIP

Message Timestamp Point: Time Errors and Adaptations

- The sizes of the potential timing errors that can result from using different message timestamp points are given on slide 3
- What can be done for implementations that use a message timestamp point at “the beginning of SFD” to adapt them to using a message timestamp point at “the beginning of the symbol after the SFD”?
  - For single-lane interfaces
    - Add time offset of one byte time to the timestamp on a transmit interface
    - Add time offset of one byte time to the timestamp on a receive interface
  - See subsequent slides for multi-PCS lane interfaces

MICROCHIP

Message Timestamp Point: Multi-PCS Lane Interactions (1/3)

- 802.3cx compliant multi-PCS lane distribution delays:
  - Given “transmit and receive path data delays be reported as if the message timestamp point is at the start of the FEC block and multilane distribution sequence”, the multi-lane path data delay is modelled as a constant value
  - The timestamp for the SFD byte and for the symbol after SFD byte at the xMII are separated by one byte time
  - The timestamp difference between the two message timestamp points is one byte time (e.g., 0.08ns for 100GE) and can be compensated with a static timestamp offset
    - Add time offset of one byte time to the timestamp on a transmit interface
    - Add time offset of one byte time to the timestamp on a receive interface

MICROCHIP

Message Timestamp Point: Multi-PCS Lane Interactions (2/3)

- Non-802.3cx compliant multi-PCS lane distribution delays:
  - Assumes local Tx port and remote Rx port both use “option B + method 1” mechanism to account for PCS lane distribution delays
  - SFD byte and symbol after SFD byte could have timestamps that are:
    - The same, if both bytes belong to the same set of blocks at the Tx PCS output (or Rx PCS input)
    - Different by N blocks, where N = # of PCS lanes, if SFD byte belongs to an earlier set of blocks at the Tx PCS output (or Rx PCS input) than the symbol after the SFD byte
  - Compensation for this difference is implementation specific

MICROCHIP

Message Timestamp Point: Multi-PCS Lane Interactions (3/3)

- Mix of 802.3cx compliant PHY and non-802.3cx compliant PHY for multi-PCS lane distribution delays:
  - Time error is dynamic and has a variation magnitude equal to the sum of the PCS path data delay mismatch error and the message timestamp point error
  - Adaptation between these methods is implementation specific

MICROCHIP

Message Timestamp Point: AM/CWM Interactions

- Additional message timestamp point error occurs when AM/CWM is placed between the SFD byte and the symbol after the SFD byte
  - Timestamp error increases from one byte time to (one byte time + duration of the AM/CWM)
- Probability of this additional error
  - AM/CWM event occurs once every 16384 blocks
  - Probability of timestamp event is once per Ethernet frame + IFG (i.e., once every 10 blocks for min sized frame and small IFG, less often for larger frames or larger IFGs)
  - Thus, probability of AM/CWM corrupting the timestamp of *any* frame is no greater than:

$$P \leq (1/16384) * (1/10) \cong 6E\text{-}6$$

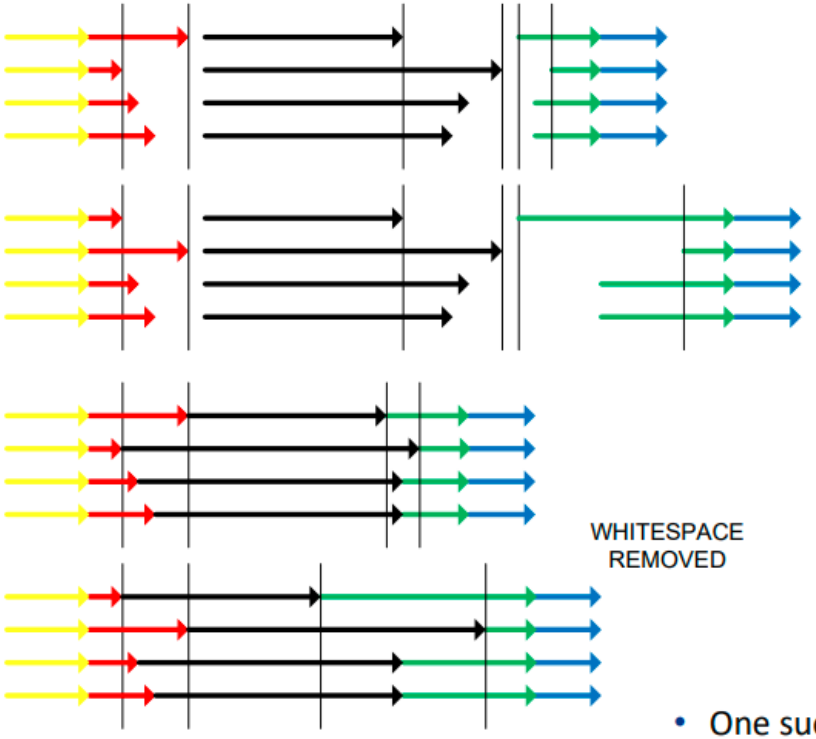
- Timestamp errors with this low probability of occurrence could be filtered by time recovery algorithms, which have low-pass filters

MICROCHIP

Tx Skew

- **Copied from** https://www.ieee802.org/3/cx/public/nov20/dekoos_3cx_01_1120.pdf

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.

- 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.

MICROCHIP

General path data delay mechanism for PHY functions with variable delays (1/6)

- **Copied from** https://www.ieee802.org/3/cx/public/may20/tse_3cx_03_0520.pdf

Characteristic Intrinsic Delays

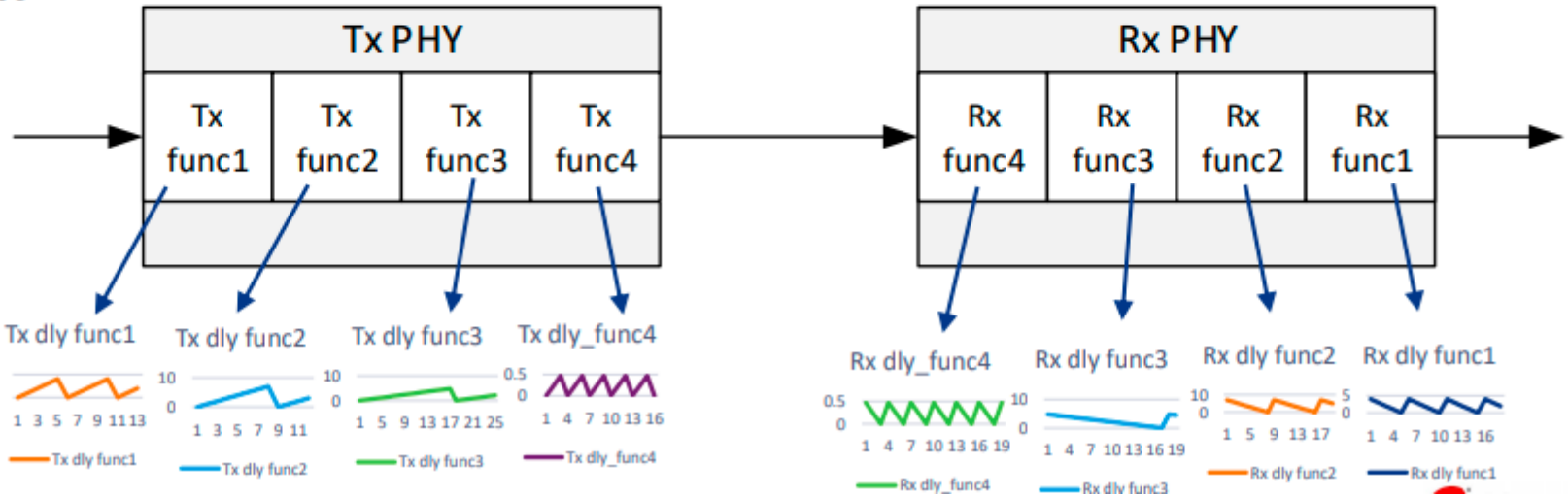
- **Some functions in PHYs have varying intrinsic delays**
  - The intrinsic delay variation often follows a repeating pattern
  - The intrinsic delay variation pattern on Tx is often a mirror of the intrinsic delay variation pattern on Rx, and the sum of the two intrinsic delays is a constant value
    - This must be true if the Tx stream before the Tx function and the Rx stream after the Rx function are identical

MICROCHIP

General path data delay mechanism for PHY functions with variable delays (2/6)

Concatenated Functions

- **Example PHY with 4 concatenated functions**
  - Multi-lane distribution
  - FEC
  - etc
  - etc

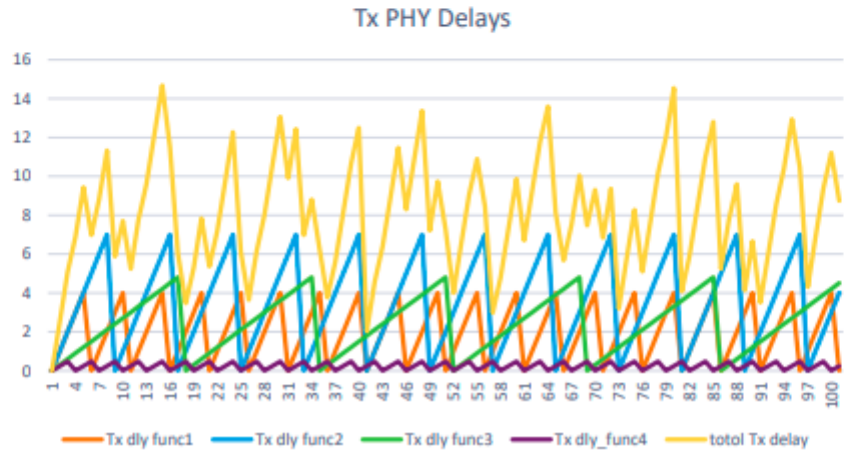


MICROCHIP

General path data delay mechanism for PHY functions with variable delays (3/6)

Total PHY Delay of Concatenated Tx Functions

- For 1-step Sync timestamping, the total Tx PHY delay must be predicted in advance so the originTimestamp can be inserted before the PTP message enters the Tx PHY
- Predicting the total Tx PHY delay of the message timestamp point through a series of concatenated Tx PHY functions can be difficult

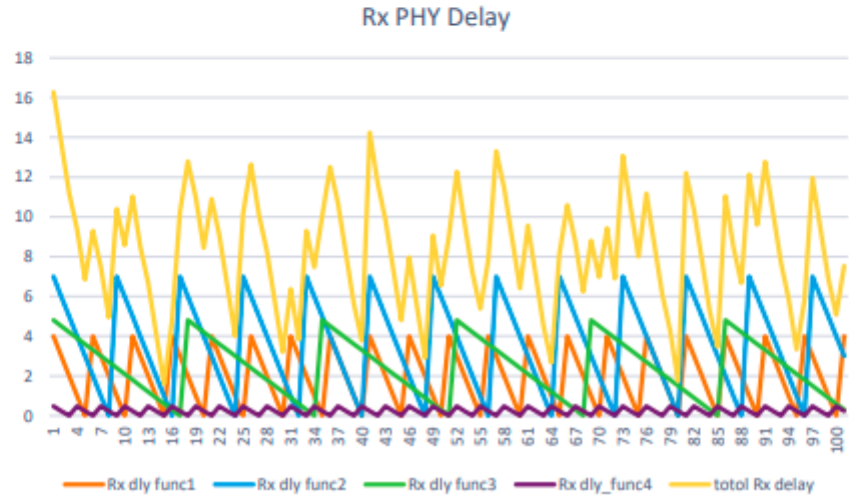


5

General path data delay mechanism for PHY functions with variable delays (4/6)

Total PHY Delay of Concatenated Rx Functions

- Rx PTP messages are most easily detected after it exits the Rx PHY
- Tracing back the message timestamp point through a series of concatenated Rx PHY functions can be difficult

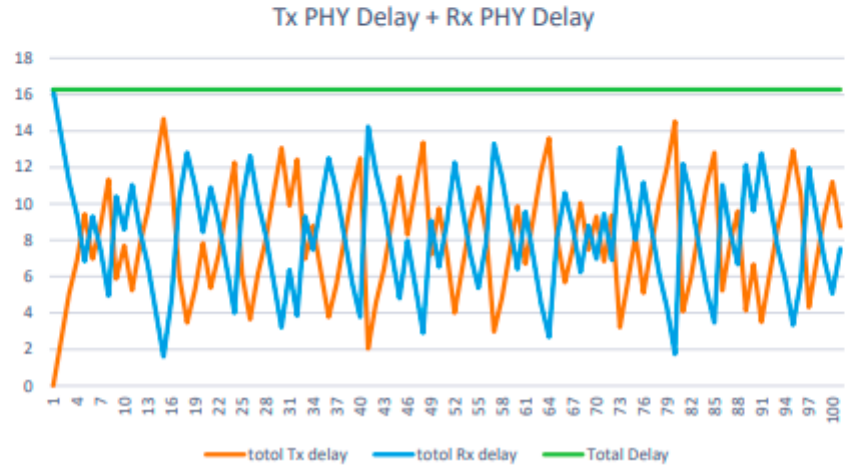


6

General path data delay mechanism for PHY functions with variable delays (5/6)

Total Delay of Tx + Rx PHYs

- **If each of the intrinsic Tx and Rx functional delays are mirrors of each other, then the end-to-end delay is a constant value**
  - One doesn't need to track the delay of the message timestamp point through either the Tx or Rx PHYs
  - One can simply standardize the allocation of a portion of the total constant delay to the Tx side and the rest to the Rx side



7

General path data delay mechanism for PHY functions with variable delays (6/6)

- It is recommended to deal with PHY delays of this nature by using the constant sum of Tx and Rx delays and allocating a specified portion of the sum to the Tx side and the remaining portion of the sum to the Rx side.

MICROCHIP

Thank You