PTP Timestamping Accuracy and 802.3

Richard Tse, Steve Gorshe

IEEE 802.3 NEA

Apr 2019

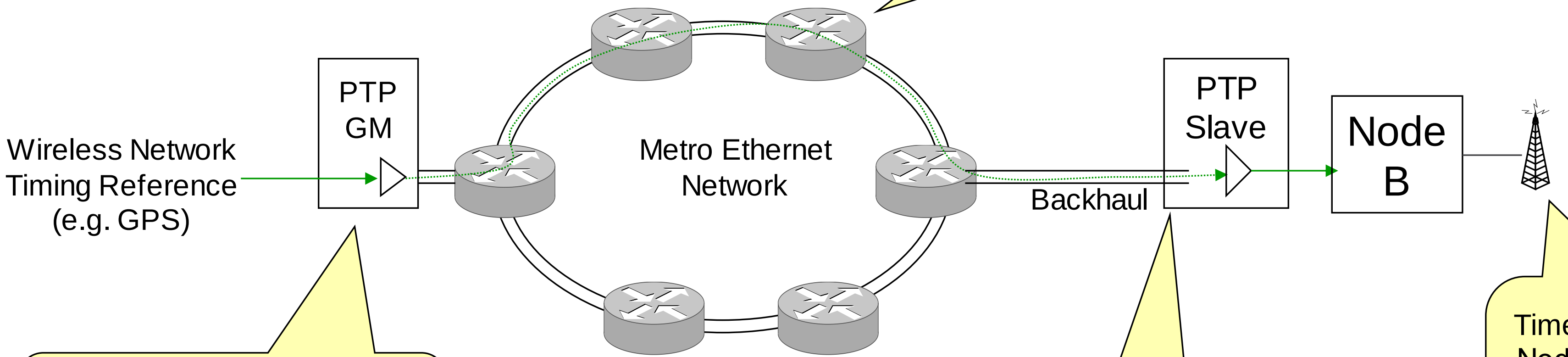
microchip.com

MICROCHIP®

Outline

1. PTP Application Example
2. PTP Fundamentals
   - Time Distribution Mechanism
   - Timestamp Generation Model
   - Time Error Measurement Model
3. Example Application Timing Requirements
4. Issues with 802.3 Timestamping
5. Performance vs Target
6. Example Actions for Improving 802.3 PTP Performance

PTP Application Example



PTP Boundary Clock: timing slave on one port, timing master on other ports

PTP Transparent Clock: measure and record residence time in the node

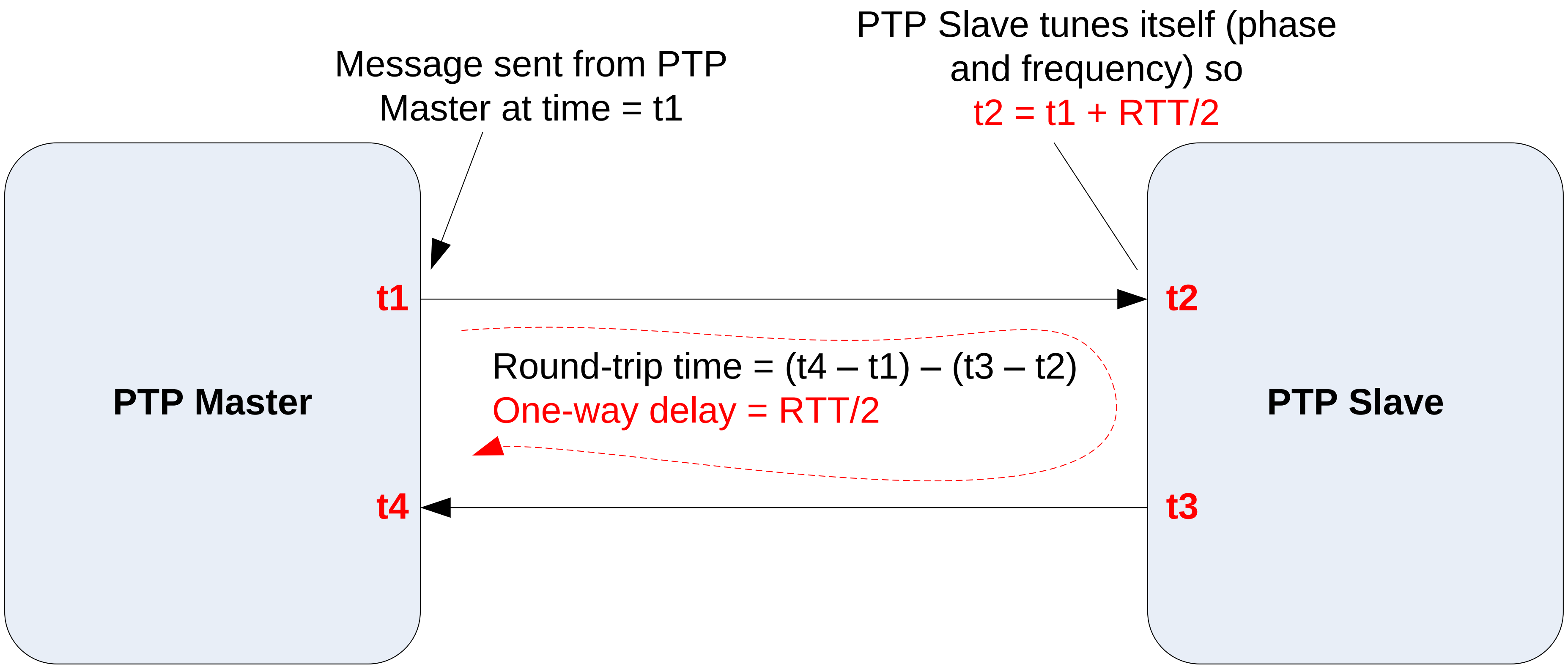
PTP Grandmaster:

primary source of PTP timing

PTP Slave: extracts timing from PTP

Time alignment allows Node B's radios to be time aligned with other Node B's radios, reducing interference

PTP Time Distribution Mechanism

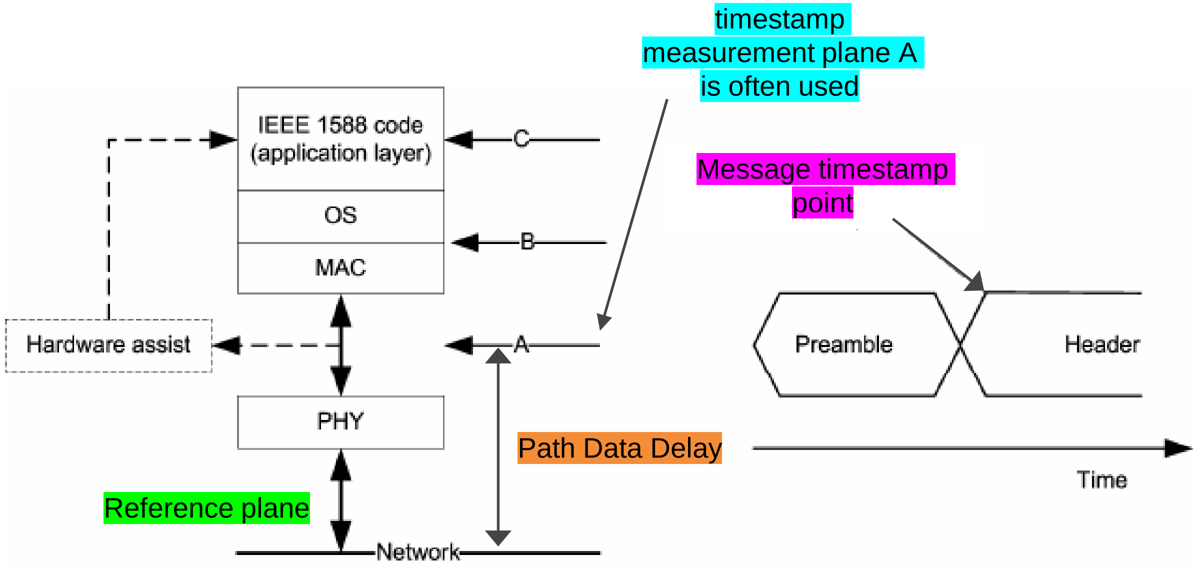


Message sent from PTP Master at time = t1
PTP Slave tunes itself (phase and frequency) so
t2 = t1 + RTT/2
PTP Master
t1
t4
PTP Slave
t2
t3
Round-trip time = (t4 – t1) – (t3 – t2)
One-way delay = RTT/2

-Timestamps **t1** and **t4** are captured at PTP Master
-Timestamps **t2** and **t3** are captured at PTP Slave
-All timestamps are given to PTP Slave to recover time

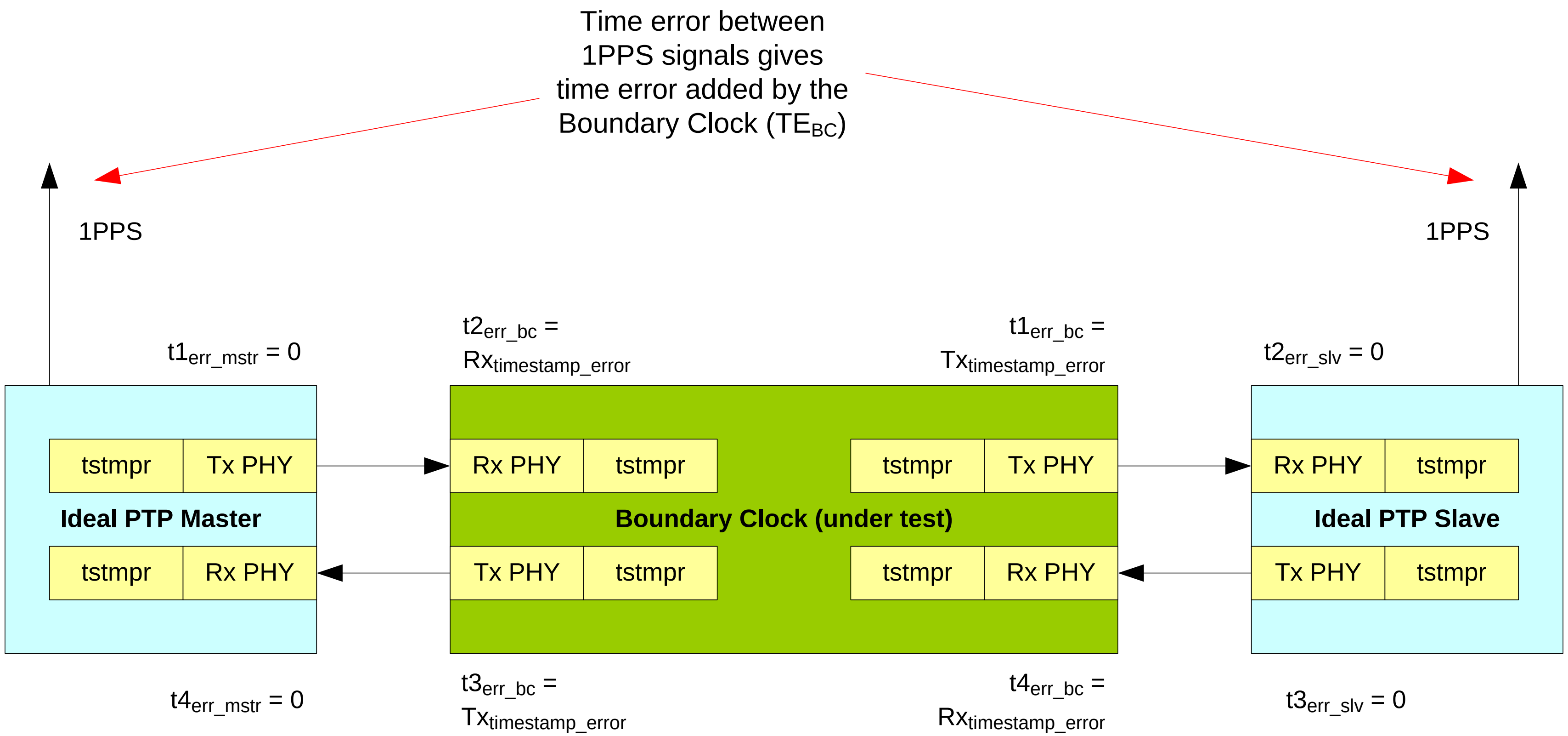
PTP Timestamp Generation Model

- A timestamp is generated at the time the "message timestamp point" crosses "reference plane", which is the intersection between the network (i.e. the medium) and the PHY
- Timestamp capture is implemented at the "timestamp measurement plane", which, in practice, occurs at point A and must be moved back to the reference plane
- *Good estimate of the PHY delay* ("path data delay", the time between the reference plane and the timestamp measurement plane) *is needed → varying delays should be compensated for*
- *Every endpoint needs to have the same understanding of these 4 concepts and how compensation is done*



Time Error Measurement Model (for Boundary Clock)

- PTP Master and PTP Slave are ideal (no timestamping errors, perfectly stable clocks)
- Boundary Clock's time error (TE) is affected by timestamping errors on messages to/from Master and to/from Slave
  - other sources of TE are ignored for this discussion
- $|TE_{BC}| = 0.5*(|t1_{err\_bc}| + |t2_{err\_bc}| + |t3_{err\_bc}| + |t4_{err\_bc}|) = (|Tx_{timestamp\_error}| + |Rx_{timestamp\_error}|)$


Time error between 1PPS signals gives time error added by the Boundary Clock (TE_{BC})
1PPS
1PPS
$t1_{err\_mstr} = 0$
$t2_{err\_bc} = Rx_{timestamp\_error}$
$t1_{err\_bc} = Tx_{timestamp\_error}$
$t2_{err\_slv} = 0$
tstmpr
Tx PHY
Rx PHY
tstmpr
tstmpr
Tx PHY
Rx PHY
tstmpr
Ideal PTP Master
Boundary Clock (under test)
Ideal PTP Slave
tstmpr
Rx PHY
Tx PHY
tstmpr
tstmpr
Rx PHY
Tx PHY
tstmpr
$t4_{err\_mstr} = 0$
$t3_{err\_bc} = Tx_{timestamp\_error}$
$t4_{err\_bc} = Rx_{timestamp\_error}$
$t3_{err\_slv} = 0$

Example Application Timing Requirements

- From ITU-T Recommendation G.8273.2, Timing characteristics of telecom boundary clocks and telecom slave clocks
  - Specifies the max timing errors that can be added by a telecom boundary clock
  - cTE is constant error
  - dTE$_L$ is low-passed dynamic error
  - TE$_L$ is constant error + low-passed dynamic error
  - TE is constant error + unfiltered dynamic error

Time Error Type	Class	Requirement (ns)
$\max\lvert TE\rvert$	A	100
	B	70
	C	30
	D	for further study
$\max\lvert TE_L\rvert$	A, B, C	not defined
	D	5

Class	cTE Requirement (ns)
A	±50
B	±20
C	±10
D	for further study

Time Error Type	Class	Requirement (ns)	Observation interval τ (s)
dTE$_L$	A and B	MTIE = 40	m < τ ≤ 1000 (for constant temp)
	A and B	MTIE = 40	m < τ ≤ 10000 (for variable temp)
	C	MTIE = 10	m < τ ≤ 1000 (for constant temp)
	D	MTIE = for further study	
	A and B	TDEV = 4	m < τ ≤ 1000 (for constant temp)
	C	TDEV = 2	
	D	TDEV = for further study	

Issues with 802.3 Timestamping

Improvements to Clause 90 are needed to enable better PTP performance

1. Message Timestamp Point and Tx/Rx Path Data Delay
2. Specify how delay variance from AM and Idle insertion/removal events are accounted for
3. Clarify timestamping for multi-lane PHYs
4. Specify how delay variance from multi-lane distribution mechanism is accounted for

Message Timestamp Point

Subclause 90.7 of IEEE 802.3 states

- “The transmit path data delay is measured from the input of the beginning of the SFD at the xMII to its presentation by the PHY to the MDI. The receive path data delay is measured from the input of the beginning of the SFD at the MDI to its presentation by the PHY to the xMII.”

however…

Subclause 7.3.4.1 of IEEE 1588v2 and subclause 11.3.9 of IEEE 802.1AS define the message timestamp point as follow:

- “the message timestamp point for an event message shall be the beginning of the first symbol after the Start of Frame (SOF) delimiter”
- “the message timestamp point for a PTP event message shall be the beginning of the first symbol following the start of frame delimiter”

Effect of Mismatched Message Timestamp Points

- Link delay measurement is affected by the message timestamp point
    - A timestamp at the beginning of SFD is earlier than a timestamp at the beginning of the first symbol after SFD
    - Examples:
        - Master and slave both use symbol after SFD:
            - Measured link delay = X
        - Master and slave both use beginning of SFD:
            - Measured link delay = X
        - Master uses symbol after SFD and Slave uses beginning of SFD:
            - Measured link delay = X – T_{SFD}
                - T_{SFD} is the time occupied by a SFD symbol
                - creates a constant time error cTE = T_{SFD}
- Alignment marker could also separate the SFD and the symbol after the SFD, creating an even greater discrepancy between their corresponding timestamps

AM and IDLE Insertion/Removal

Alignment Marker (AM) and Idle insertion/removal affect the path data delay:

- Insertion of AM or Idle momentarily increases the path data delay by T_{AM} or T_{Idle}, respectively
- Removal of AM or Idle momentarily decreases the path data delay by T_{AM} or T_{Idle}, respectively
- Idle insertion/removal operate independently at Rx and Tx so delay changes do not have deterministic relationship
- AM removal at Rx deterministically undoes the delay change caused by AM insertion at Tx
  - However, AM events cause many additional Idle insertion/removal events

Multi-Lane PHY Ambiguities

Ambiguities in 802.3 can affect path data delay values.

- Ambiguities for N-lane Transmitter implementation
  - A. Codewords and timestamps are not aligned at N-lane transmitter output
  - B. Codewords and timestamps are aligned at N-lane transmitter output
  - C. Codewords are aligned but timestamps are not aligned at N-lane transmitter output
- Path data delays for the lane distribution function can be different for each lane in Tx and Rx PHYs
  - Example: received lane 0 codeword goes to xMII first while received lane N goes to xMII last
  - No instructions are given on how to handle these deterministic but varying path data delays
- Interactions between implementations that interpret the specification differently will have additional time error
- See Appendix for details on the above items

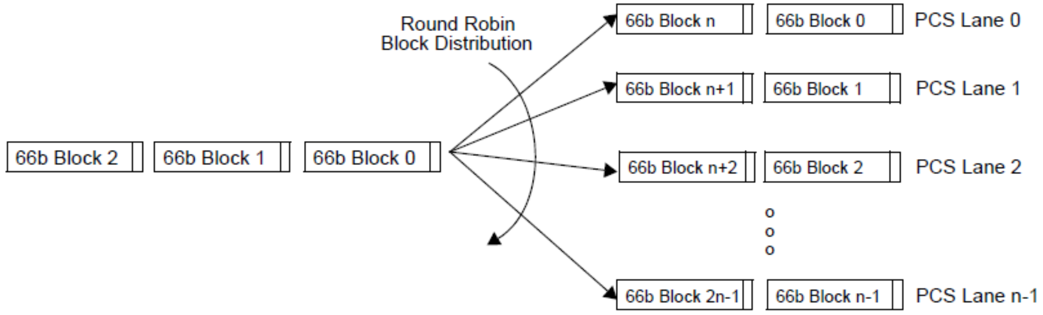


Figure 82–6—PCS Block distribution

Performance vs Target

- Max|TE| = 30ns for class C Telecom Boundary Clock (see slide 7)
  - There are other sources of TE in addition to those from timestamping

Ethernet Rate	Path Data Delay Variation per Tx/Rx Interface (ns)				Total TE per Tx or Rx Interface (ns)	Max|TE| contribution per PTP Boundary Clock (ns)
	mismatched SFD timestamp point	Idle insert/remove (per Idle)	AM insert/remove	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	25.68	51.36
200GE	0.04	0.32	2.56	2.24	5.16	10.32
400GE	0.02	0.16	2.56	2.4	5.14	10.28

Example Actions for Improving 802.3 PTP Performance

- Write new specifications for high performance implementations
  - Redefine message timestamping point to be “symbol after SFD” for high performance PTP
    - Consistent with IEEE 1588 and IEEE 802.1AS
    - Some legacy implementations could adapt by adding constant offset T_{SFD} to timestamps
  - Specify that the actual path data delay experienced by the PTP message, with any AM and Idle insertion/removal, is reflected in the timestamp
    - Many existing implementations already do this
  - Remove ambiguities for multi-lane PHYs
    - Clarify Tx lane alignment and timestamping
    - Specify how lane distribution delay variation is accounted for

Example Actions for Improving 802.3 PTP Performance

- Add Appendix that provides informative data on timestamp accuracy limits for implementations based on clause 90
  - E.g. table from slide 13

- Write white-paper that recommends how to implement these functions for high performance timestamping

MICROCHIP

Thank You

Appendix

Details on Lane Distribution Delay Issue

Lane Distribution Interpretation Option Details (1)

Ambiguities in 802.3 affect path data delays.

No instructions are given in 802.3 on how to handle the following deterministic but varying delays

- N-lane Transmitter Interpretation Options
  A. Codewords and timestamps are not aligned at N-lane transmitter
     - xMII to MDI has constant path data delay for every lane
       - Lane 0 arrives first at xMII and is transmitted first at MDI
       - Lane N arrives last at xMII and is transmitted last at MDI
     - Codewords on each lane have a different timestamp because they cross the reference plane at different times
       - Timestamper at Tx xMII uses the same xMII to MDI constant data path delay for every lane
     - Lane-to-lane skew of codewords at the transmitter is removed by Rx deskew buffers
  B. Codewords and timestamps are aligned at N-lane transmitter
     - xMII to MDI has different path data delay for each lane
       - Lane 0 arrives first at xMII and is transmitted at the same time as lane N at MDI, causing largest path data delay
       - Lane N arrives last at xMII and is transmitted at the same time as Lane 0 at MDI, causing smallest path data delay
     - Codewords on every lane have the same timestamp because they cross the reference plane at the same time
       - Timestamper at Tx xMII uses appropriate xMII to MDI path data delay for each lane
     - No lane-to-lane skew of codewords

Lane Distribution Interpretation Option Details (2)

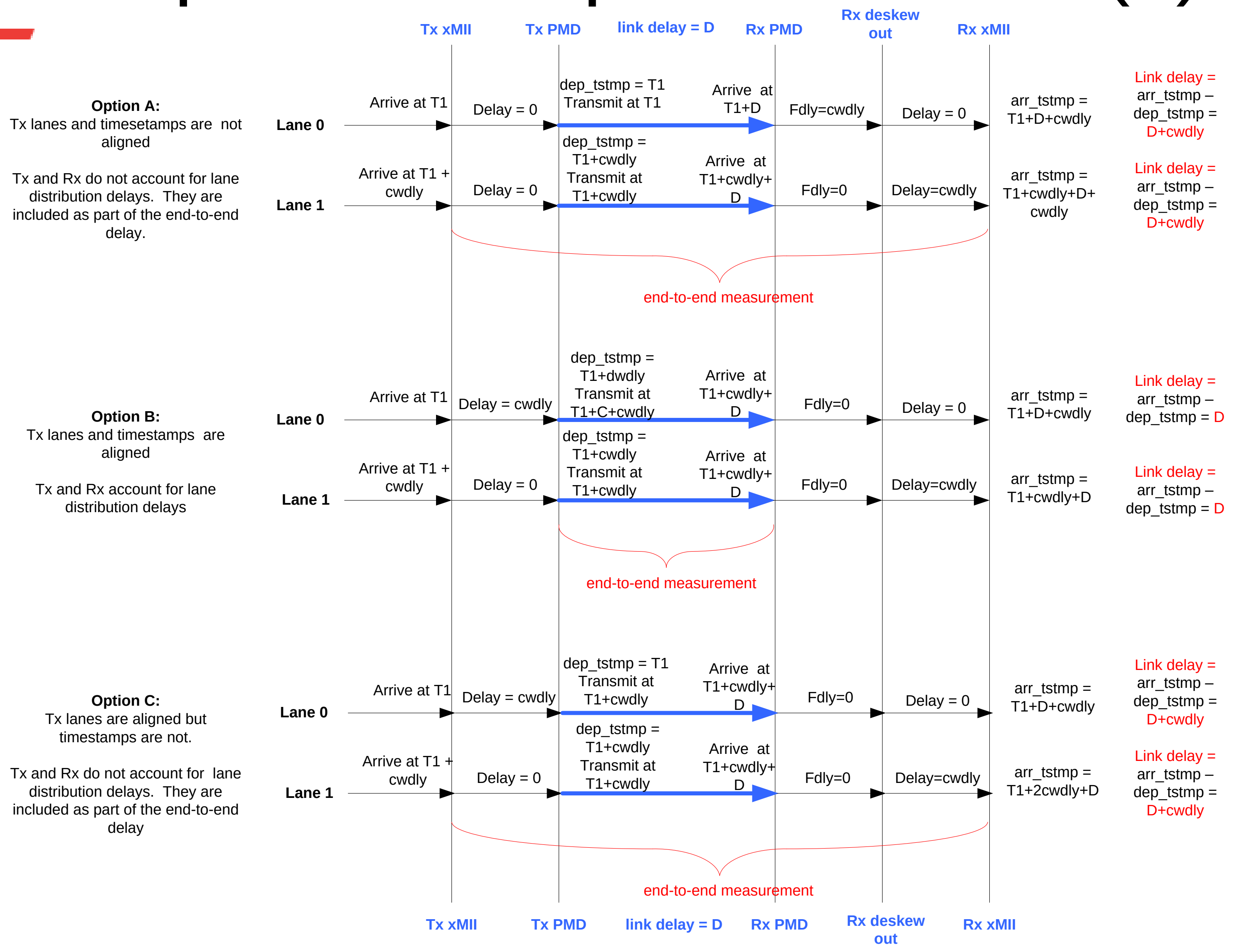
- N-lane Transmitter Options (continued)
  C. Codewords are aligned but timestamps are not aligned at N-lane transmitter
    - xMII to MDI has different path data delay for each lane
      - Lane 0 arrives first at xMII and is transmitted at the same time as lane N at MDI, causing largest path data delay
      - Lane N arrives last at xMII and is transmitted at the same time as Lane 0 at MDI, causing smallest path data delay
    - Timestamps assume a constant data path delay for all lanes
      - Timestamper at Tx xMII uses the same xMII to MDI constant path data delay for every lane
    - No lane-to-lane skew of codewords

Lane Distribution Interpretation Option Details (3)

- N-lane Receiver Options:
  - After deskew buffers, all lanes are aligned
    - For N-lane transmitter type “A”, intrinsic lane-to-lane skew of codewords is “moved into the medium” by the deskew function
    - For N-lane transmitter types “B” and “C”, there is no skew of codewords between lanes
  - MDI to xMII multiplexer causes varying path data delay
    - All lanes are deskewed and are ready to go to xMII
    - Lane 0 goes to xMII first and has smallest path data delay
    - Lane N goes to xMII last and has largest
  - How is this lane-to-lane varying delay handled?

Lane Distribution Interpretation Options Details (4)

- Figure shows examples of the 3 Options
- Arrival times at each stage are shown (Arrive at, Transmit at)
- The delays through each functional stage are shown (Delay, Fdly, link delay)
    - Constant delays are assumed to be 0 where the actual values don't matter
- The departure timestamps at Tx (dep_tstmp) and arrival timestamps at Rx (arr_tstmp) are shown
- The calculated link delay (Link_delay) is shown for the span (end-to-end measurement)



Tx xMII | Tx PMD | link delay = D | Rx PMD | Rx deskew out | Rx xMII

Option A:
Tx lanes and timesetamps are not aligned

Tx and Rx do not account for lane distribution delays. They are included as part of the end-to-end delay.

Lane 0: Arrive at T1 → Delay = 0 → dep_tstmp = T1, Transmit at T1 → Arrive at T1+D → Fdly=cwdly → Delay = 0 → arr_tstmp = T1+D+cwdly → Link delay = arr_tstmp – dep_tstmp = D+cwdly

Lane 1: Arrive at T1 + cwdly → Delay = 0 → dep_tstmp = T1+cwdly, Transmit at T1+cwdly → Arrive at T1+cwdly+D → Fdly=0 → Delay=cwdly → arr_tstmp = T1+cwdly+D+cwdly → Link delay = arr_tstmp – dep_tstmp = D+cwdly

end-to-end measurement

Option B:
Tx lanes and timestamps are aligned

Tx and Rx account for lane distribution delays

Lane 0: Arrive at T1 → Delay = cwdly → dep_tstmp = T1+dwdly, Transmit at T1+C+cwdly → Arrive at T1+cwdly+D → Fdly=0 → Delay = 0 → arr_tstmp = T1+D+cwdly → Link delay = arr_tstmp – dep_tstmp = D

Lane 1: Arrive at T1 + cwdly → Delay = 0 → dep_tstmp = T1+cwdly, Transmit at T1+cwdly → Arrive at T1+cwdly+D → Fdly=0 → Delay=cwdly → arr_tstmp = T1+cwdly+D → Link delay = arr_tstmp – dep_tstmp = D

end-to-end measurement

Option C:
Tx lanes are aligned but timestamps are not.

Tx and Rx do not account for lane distribution delays. They are included as part of the end-to-end delay

Lane 0: Arrive at T1 → Delay = cwdly → dep_tstmp = T1, Transmit at T1+cwdly → Arrive at T1+cwdly+D → Fdly=0 → Delay = 0 → arr_tstmp = T1+D+cwdly → Link delay = arr_tstmp – dep_tstmp = D+cwdly

Lane 1: Arrive at T1 + cwdly → Delay = 0 → dep_tstmp = T1+cwdly, Transmit at T1+cwdly → Arrive at T1+cwdly+D → Fdly=0 → Delay=cwdly → arr_tstmp = T1+2cwdly+D → Link delay = arr_tstmp – dep_tstmp = D+cwdly

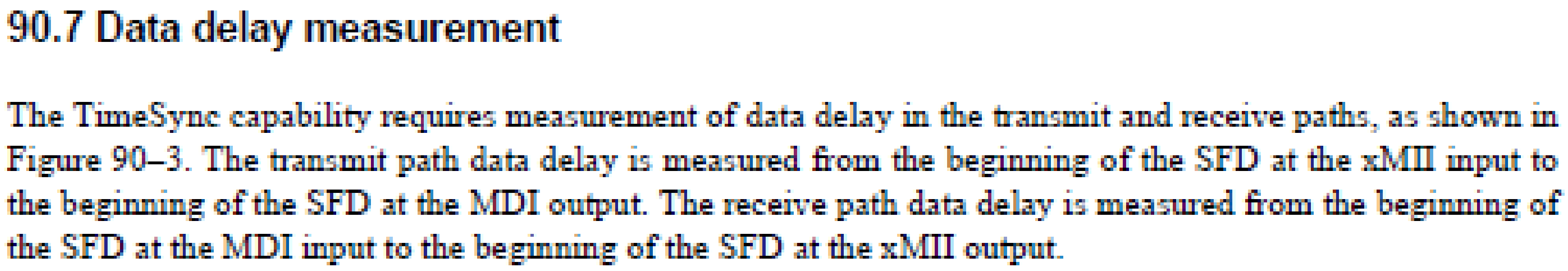
end-to-end measurement

Tx xMII | Tx PMD | link delay = D | Rx PMD | Rx deskew out | Rx xMII

Lane Distribution Delays – Constant vs per-Lane

- There are two inherent approaches for determining the xMII-to-MDI delay on multi-lane PHYs
  1. Method 1 – Account for the delay between the MII and the lane that carries the message timestamp point of the PTP message.
  2. Method 2 – Because the Tx + Rx lane distribution delay is a constant for every lane, use this constant delay regardless of which lane carries the message timestamp point.
     - This is like how 802.3 handles FEC delays

Lane Distribution Delays: Method 1



90.7 Data delay measurement

The TimeSync capability requires measurement of data delay in the transmit and receive paths, as shown in Figure 90–3. The transmit path data delay is measured from the beginning of the SFD at the xMII input to the beginning of the SFD at the MDI output. The receive path data delay is measured from the beginning of the SFD at the MDI input to the beginning of the SFD at the xMII output.

- For a multilane PHY, after deskew delays are accounted for appropriately and since timestamping is at the MDI, would the timestamps be the same regardless of which lane the message’s timestamp reference point is transmitted on (or received on)?
  - Since all lanes are transmitted at the same time and received at the same time (after deskew) at the MDI, it would seem this is a valid conclusion.

Lane Distribution Delays: Method 1 (continued)

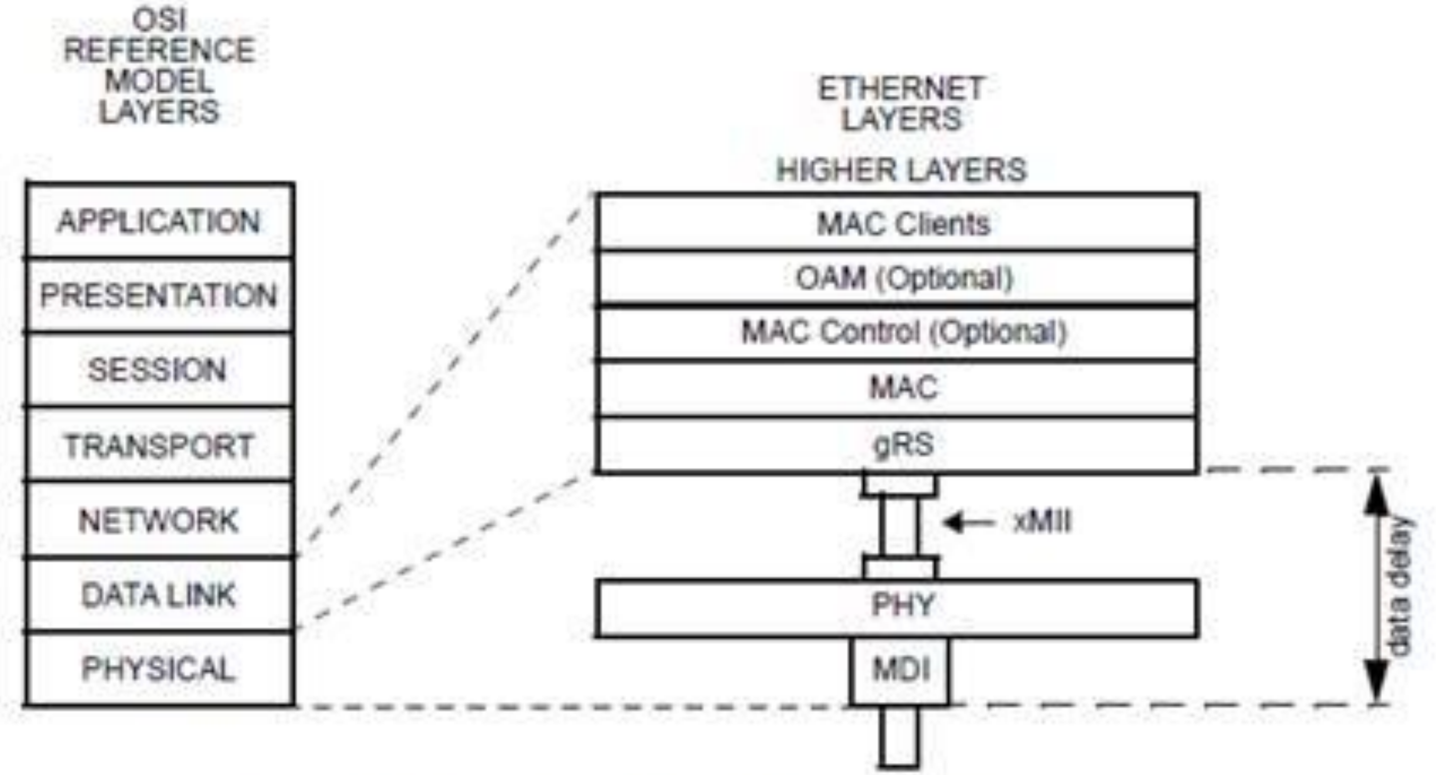
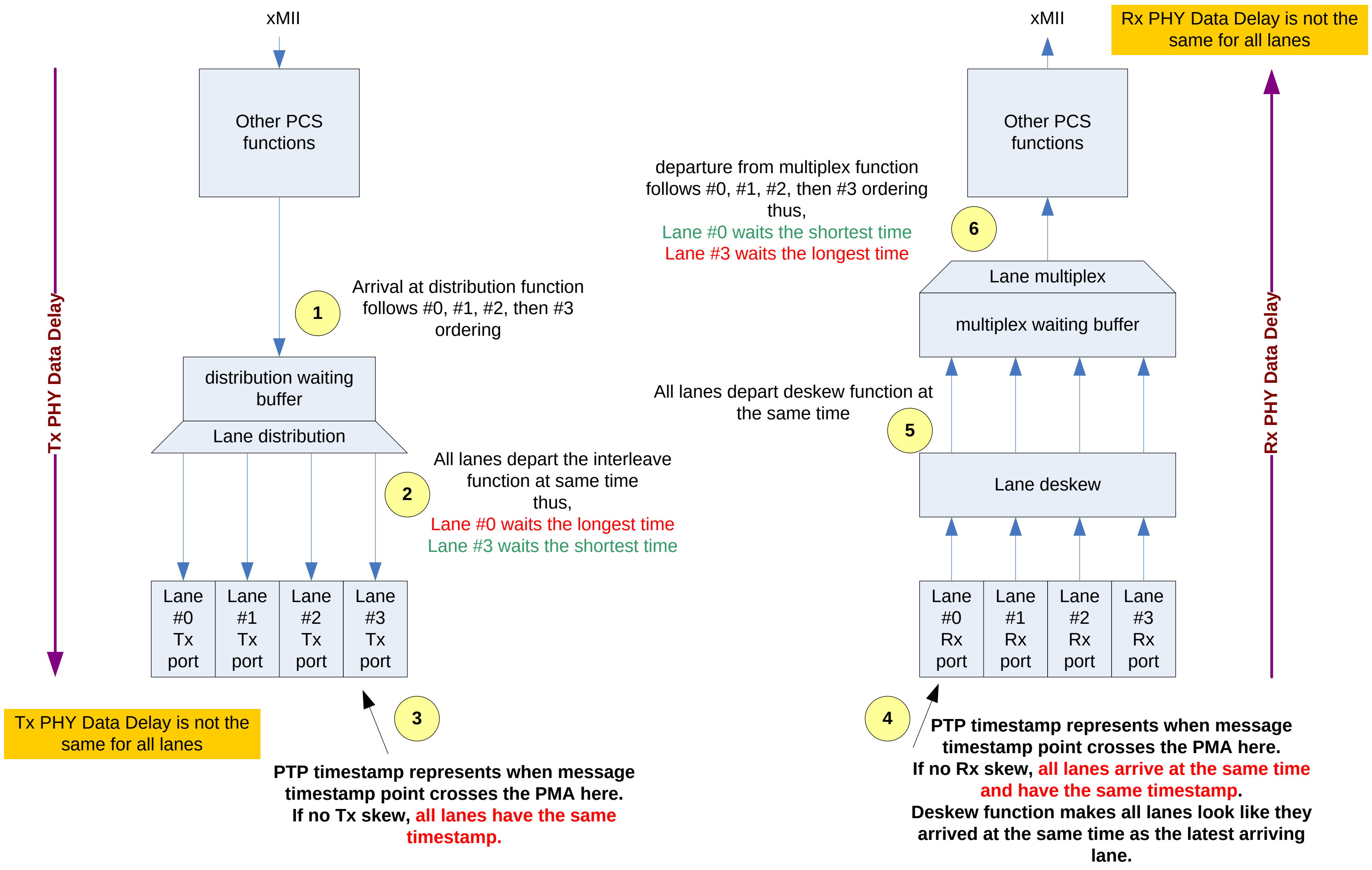


Figure 90–3—Data delay measurement

Lane Distribution Delays: Method 1 (continued)

- However, this means that PHY data delay (between xMII and MDI, as per Figure 90-3 above) is not the same for every lane because the MDI-to-xMII multiplexing delay (for Rx) and xMII-to-MDI demultiplexing delay (for Tx) is different for each lane (as shown in Figures 82-3 and 82-4 below). In the Tx direction, codewords going to lane 0 have the most delay and codewords going to lane 3 have the least delay. In the Rx direction, the opposite is true. To capture an accurate timestamp at the xMII (as per the 802.3 model), the lane-based intrinsic delay must be included as part of the PHY data delay.
  - Was this the intent?

Lane Distribution Delays: Method 1 (continued)



Arrival at distribution function follows #0, #1, #2, then #3 ordering

All lanes depart the interleave function at same time thus,
Lane #0 waits the longest time
Lane #3 waits the shortest time

Tx PHY Data Delay is not the same for all lanes

PTP timestamp represents when message timestamp point crosses the PMA here.
If no Tx skew, all lanes have the same timestamp.

Rx PHY Data Delay is not the same for all lanes

departure from multiplex function follows #0, #1, #2, then #3 ordering thus,
Lane #0 waits the shortest time
Lane #3 waits the longest time

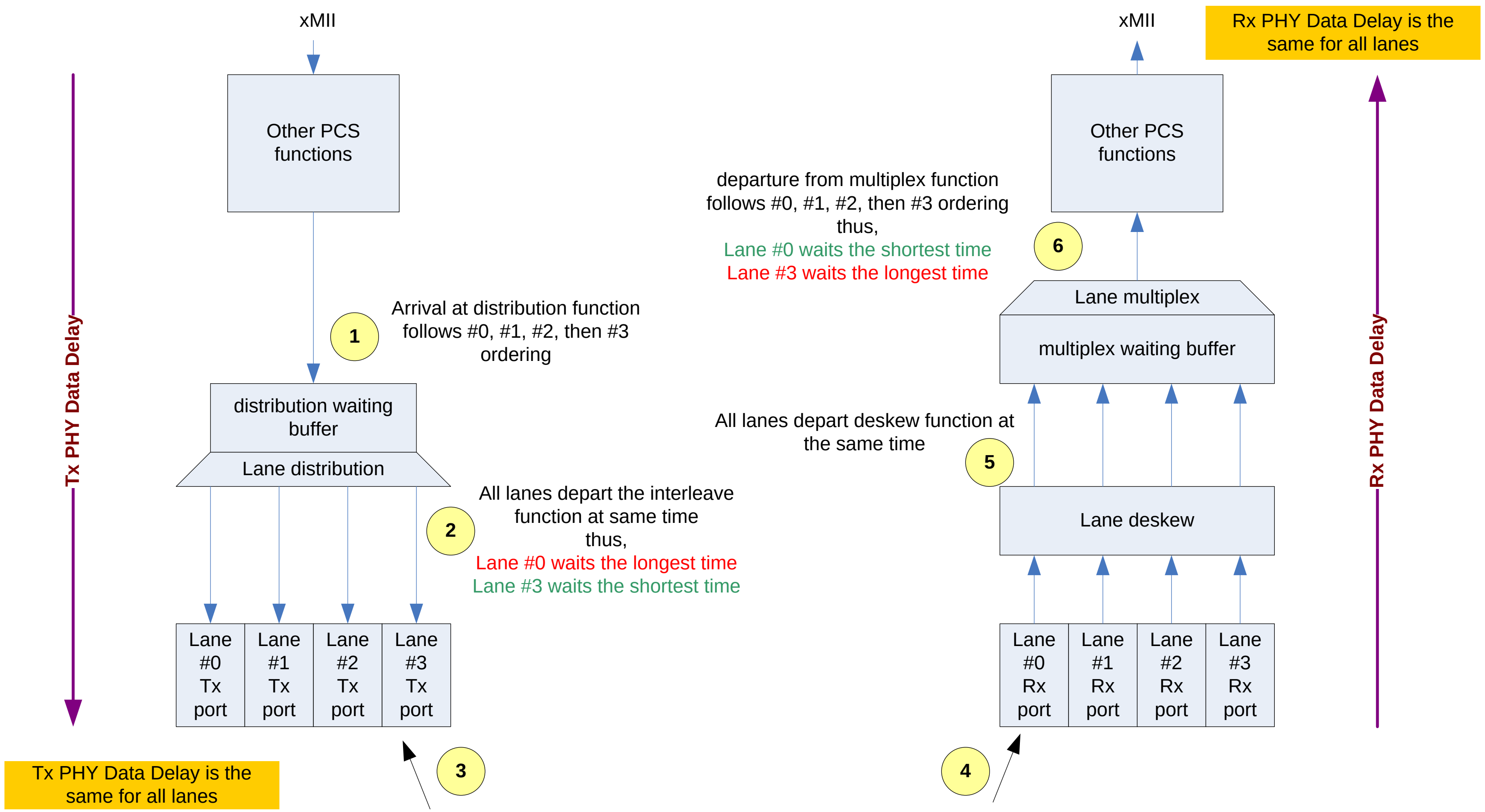
All lanes depart deskew function at the same time

PTP timestamp represents when message timestamp point crosses the PMA here.
If no Rx skew, all lanes arrive at the same time and have the same timestamp.
Deskew function makes all lanes look like they arrived at the same time as the latest arriving lane.

Lane Distribution Delays: Method 2

- These multilane PHY data delays could also be designated to be a constant value for all lanes if the principle that is used for FEC’s varying intrinsic delays is applied for multilane’s multiplexing/demultiplexing varying intrinsic delays.
    - i.e., the Tx intrinsic demultiplexing delay is balanced by the Rx multiplexing intrinsic delay, making the aggregated demux/mux delay a constant.
    - Was this principle on anyone’s mind when the multiplane PHY function was defined?

Lane Distribution Delays: Method 2 (continued)



xMII

Other PCS functions

Tx PHY Data Delay

1 Arrival at distribution function follows #0, #1, #2, then #3 ordering

distribution waiting buffer

Lane distribution

2 All lanes depart the interleave function at same time thus,
Lane #0 waits the longest time
Lane #3 waits the shortest time

Lane #0 Tx port	Lane #1 Tx port	Lane #2 Tx port	Lane #3 Tx port

Tx PHY Data Delay is the same for all lanes

3

xMII

Rx PHY Data Delay is the same for all lanes

Other PCS functions

departure from multiplex function follows #0, #1, #2, then #3 ordering thus,
Lane #0 waits the shortest time
Lane #3 waits the longest time

6

Lane multiplex

multiplex waiting buffer

All lanes depart deskew function at the same time

5

Lane deskew

Rx PHY Data Delay

Lane #0 Rx port	Lane #1 Rx port	Lane #2 Rx port	Lane #3 Rx port

4

Distribution function's delay variance is a defined to be a constant (actual variance is cancelled out by the peer Rx multiplex function).
Departure timestamps are defined to have a constant offset relative to timestamp at xMII.
Despite departing at the same time, all lanes have different timestamps.

Multiplex function's delay variance is defined to be a constant, and undoes the delay variance added by the peer Tx distribution function.
Arrival timestamps are defined to have a constant offset relative to timestamp at xMII.
After deskew, all lanes arrive at the same time but have different timestamps.