

FREEDOM TO INNOVATE

## **Timestamp Inaccuracy Due to Idle Insert/Delete for AMs**

Richard Tse, Steve Gorshe

IEEE 802.3 Geneva, Switzerland January 2020

[microchip.com](http://microchip.com)



# Agenda

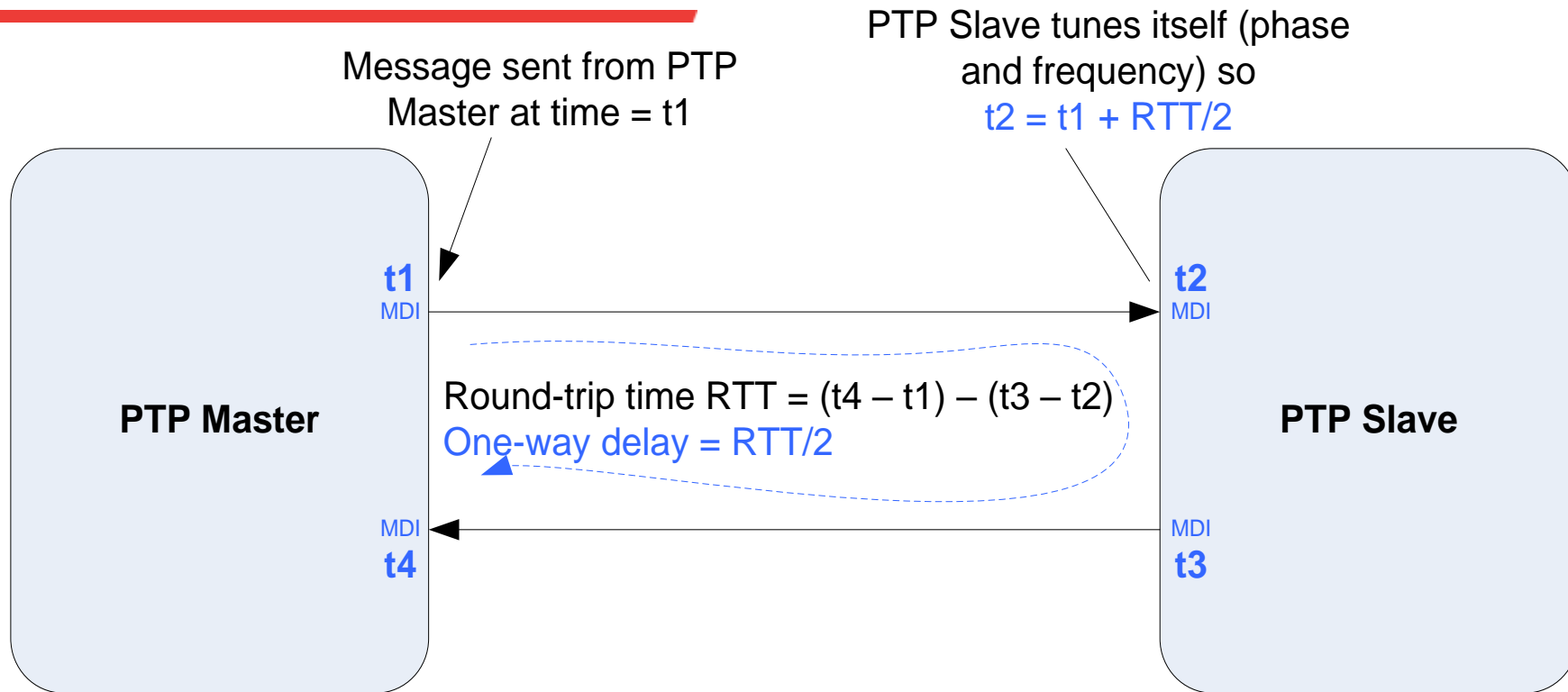
---

- Background
  - PTP Time Distribution Mechanism
  - Time Error Measurement Model
  - PTP Timestamp Generation Model
  - Current IEEE 802.3 Support for Time Synchronization
  - Why Can't High Accuracy Time Transport be Achieved Now?
- Problem: Idle insert/delete for AMs
  - Relevance of message timestamp point
  - Path data delay variation
- Potential Solution

---

# Background

# PTP Time Distribution Mechanism

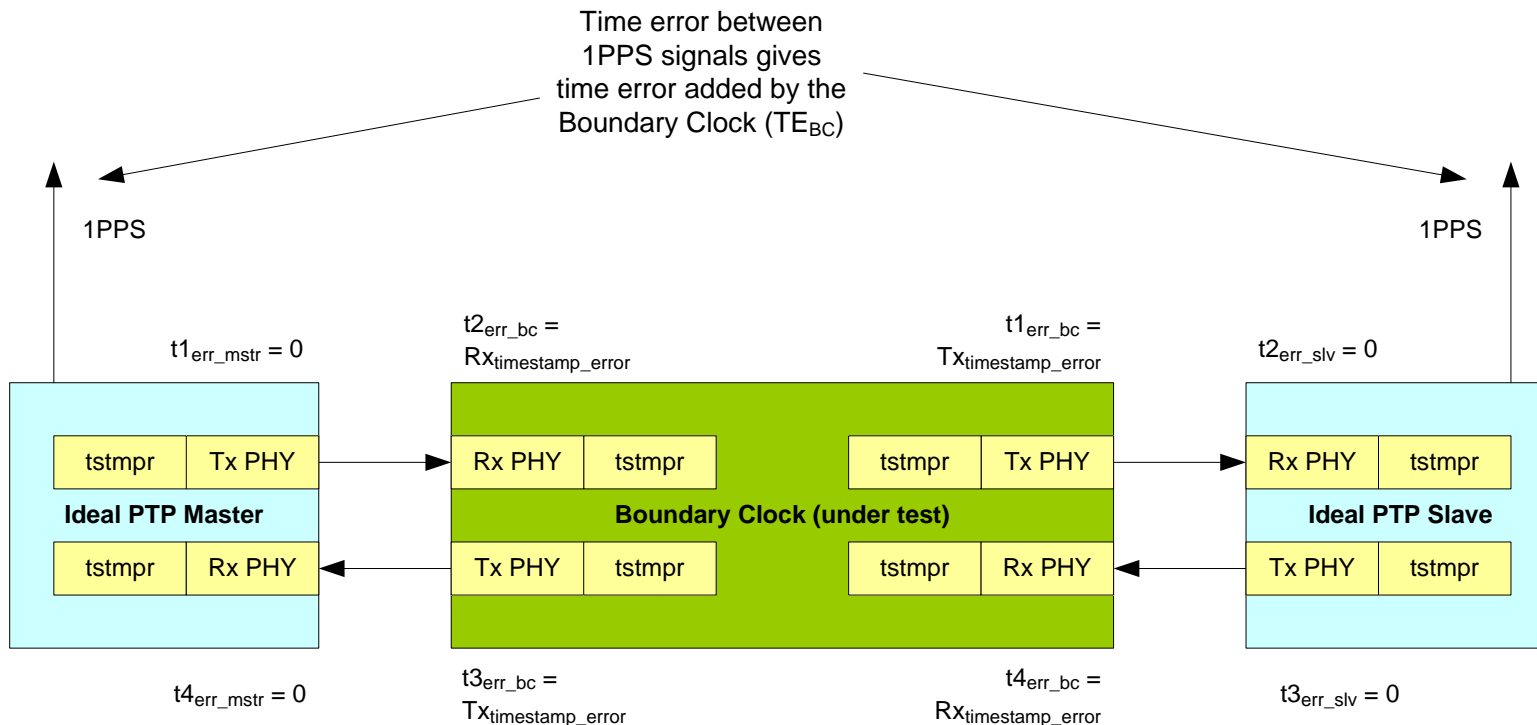


Because round-trip measurement is used, delay symmetry affects performance

- Timestamps  $t_1$  and  $t_4$  (corresponding to MDI) are captured at the PTP Master
- Timestamps  $t_2$  and  $t_3$  (corresponding to MDI) are captured at the PTP Slave
- All timestamps are given to the PTP Slave so it can:
  - calculate RTT
  - do adjustments to make  $t_2 = t_1 + RTT/2$

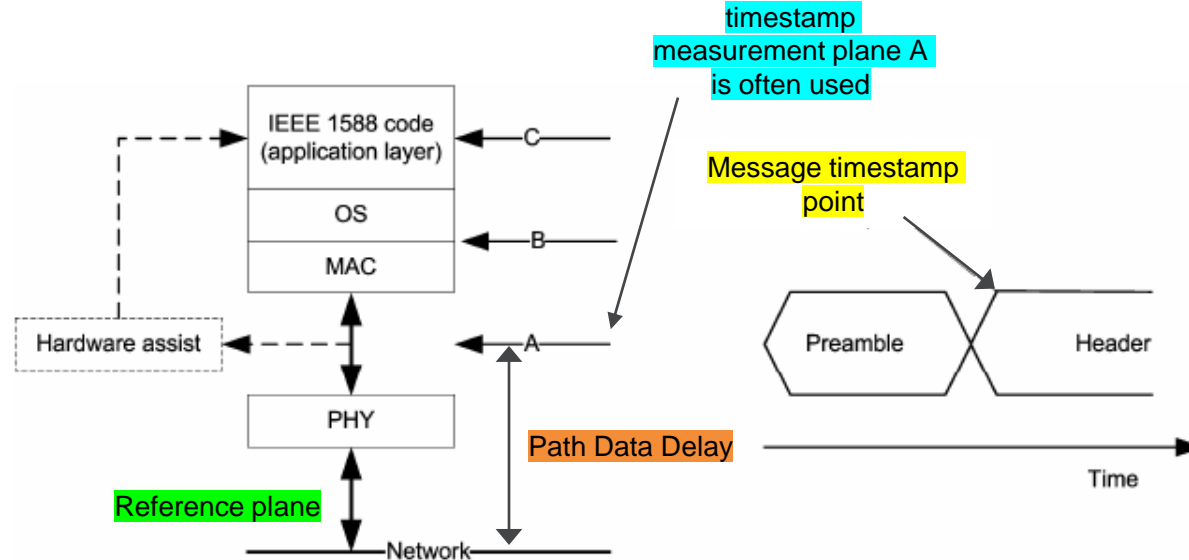
# Time Error Measurement Model (for Boundary Clock)

- 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
- $|TE_{BC}| = 0.5 * (|t1_{err\_bc}| + |t2_{err\_bc}| + |t3_{err\_bc}| + |t4_{err\_bc}|) = (|Tx_{timestamp\_error}| + |Rx_{timestamp\_error}|)$



# 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 the above concepts and how compensation is done*



# Current IEEE 802.3 Support for Time Synchronization (1)

- IEEE 802.3 Clause 90 provides support for a TimeSync Client
  - The optional Time Synchronization Service Interface (TSSI) supports protocols that require knowledge of packet egress and ingress time
  - Timestamping is done in the gRS, where the timestamp is captured when the message timestamp point crosses the xMII

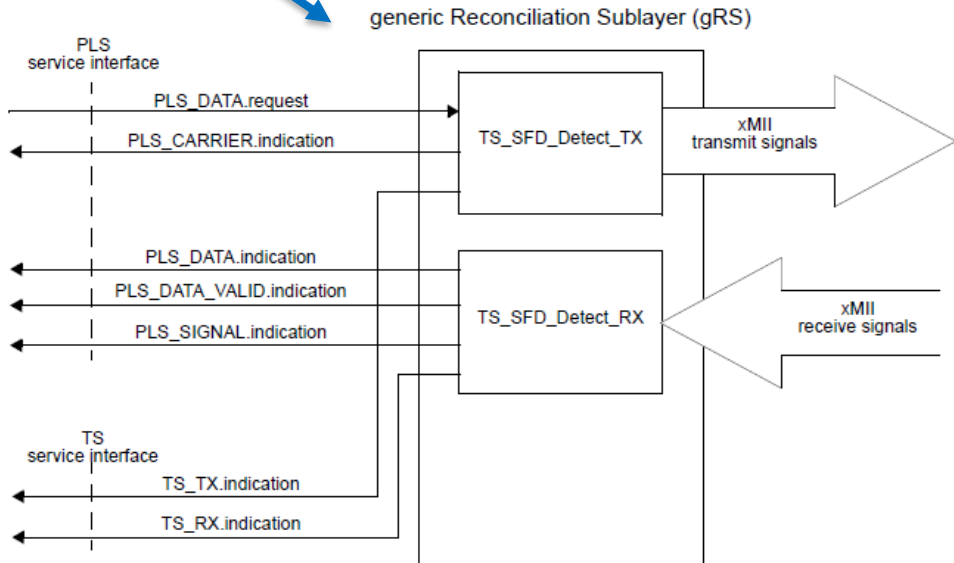
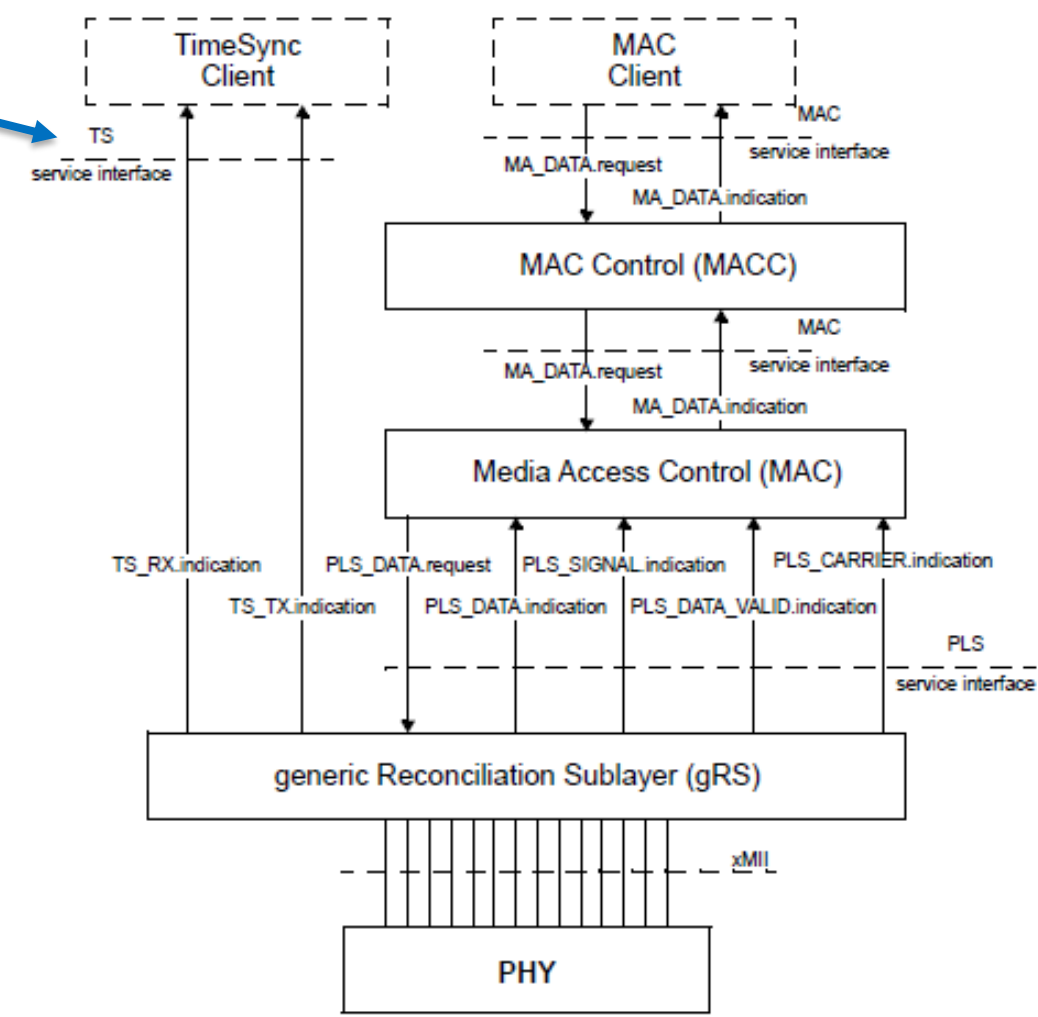


Figure 90-2—TS\_SFD\_Detect\_TX and TS\_SFD\_Detect\_RX functions within the generic Reconciliation Sublayer (gRS)



# Current IEEE 802.3 Support for Time Synchronization (2)

- TSSI allows for “PHY” delay measurement to be done by TimeSync Client(s)
  - 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.
- The obtained path data delay measurement is reported in the form of a quartet of values as defined for the TimeSync managed object class.
  - maximum transmit path data delay
  - minimum transmit path data delay
  - maximum receive path data delay
  - minimum receive path data delay

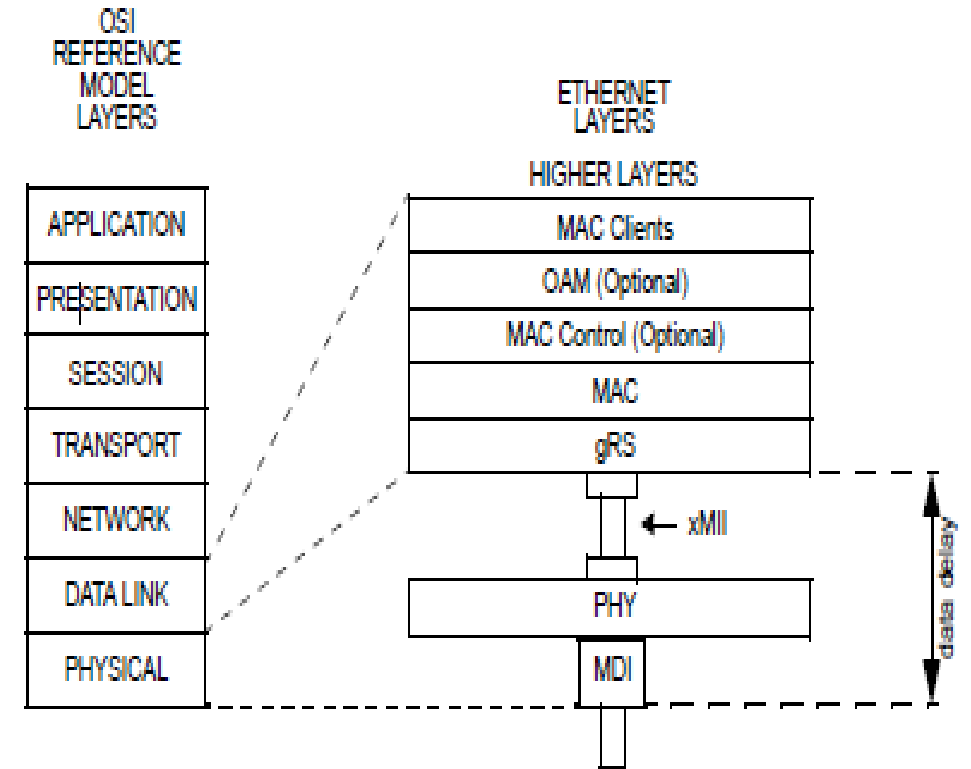


Figure 90-3—Data delay measurement



# Current IEEE 802.3 Support for Time Synchronization (3)

---

- **Multi-Lane – clause 90.7 (added in 2016):**
  - “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 SFD arrived at the MDI input on the lane with the smallest buffer delay.”
- **FEC – clause 90.7 (added in 2018):**
  - “For a PHY that includes an FEC function, the transmit and receive path data delays may show significant variation depending upon the position of the SFD within the FEC block. However, since the variation due to this effect in the transmit path is expected to be compensated by the inverse variation in the receive path, it is recommended that the transmit and receive path data delays be reported as if the SFD is at the start of the FEC block.”

# Why Can't High Accuracy Time Transport be Achieved Now with IEEE 802.3?

- PTP timestamping is done at the MDI
- IEEE 802.3's timestamping is done at the xMII (per clause 90 of IEEE 802.3)
- PHY path data delay must be known for the PTP message to move the timestamp from xMII to MDI
- Many newer 802.3 PHYs have fundamental dynamic variations in their path data delay
- But
  - Path data delay variations in the PHY are not inherently visible at the xMII
- Thus
  - IEEE 802.3's current timestamping mechanism does not inherently support high accuracy on PHYs with path data delay variations
  - Specifications are needed on how to deal with each path data delay variation

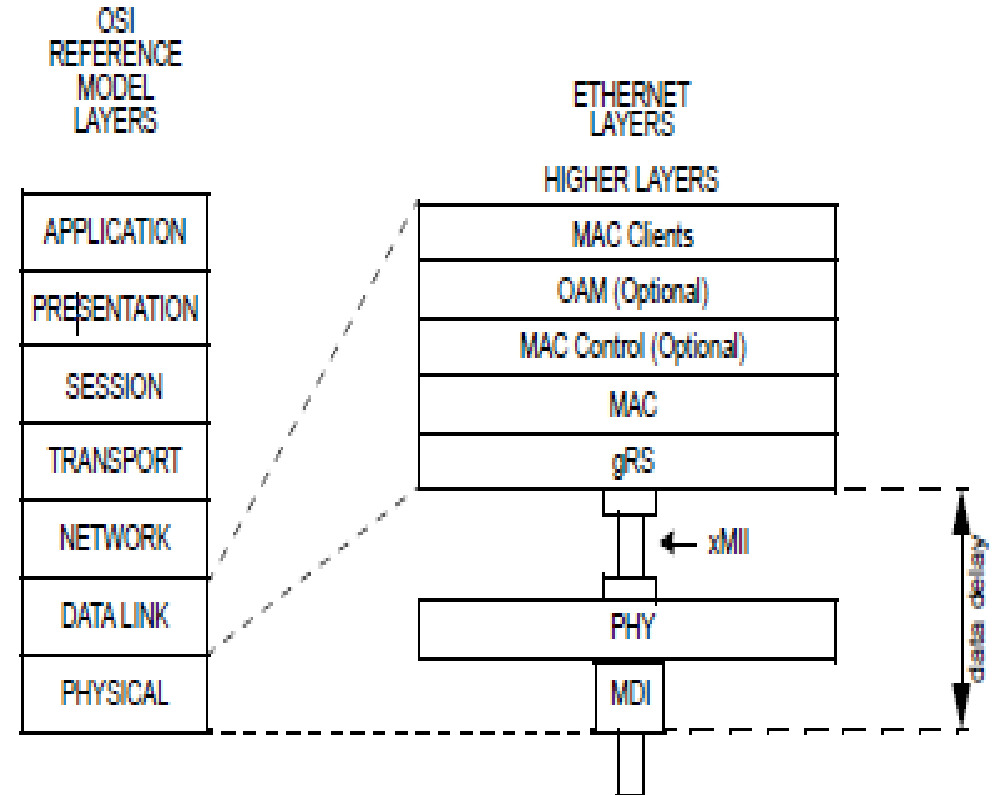


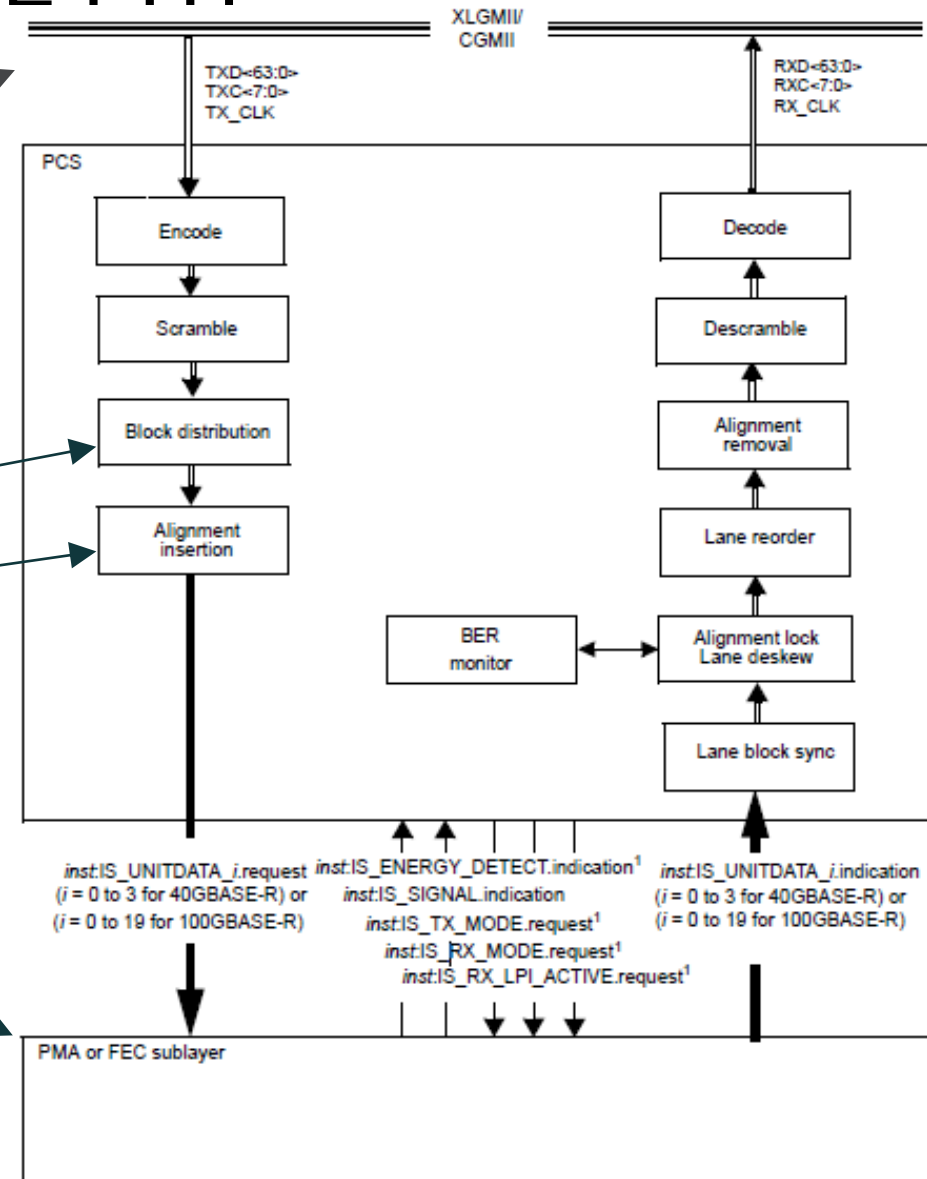
Figure 90-3—Data delay measurement

# Path Data Delay Variations in 100GE PHY

Timestamps are captured at xMII

Block distribution to multi-PCS lanes, Alignment Marker insertion/removal (and their corresponding Idles), and FEC all inherently cause dynamic path data delay variation

Timestamps should correspond to the time at MDI



NOTE 1—FOR OPTIONAL EEE DEEP SLEEP CAPABILITY

Figure 82-2—Functional block diagram

---

Problem: Path Data Delay (PDD) variance caused by Alignment Marker and Idle insertion/removal events needs to be accounted for in a standardized manner

- PHY path data delays,  $PDD_x$ ,  $(PDD_1 + PDD_2)$ , and  $(PDD_3 + PDD_4)$  values might change because these events insert or extract data within a PHY
- These changes must be detected and handled consistently in all PHYs so an accurate RTT can be measured

# Key Relevance of AMs to Timestamp Accuracy

---

- Due to the different 802.3 and 1588/802.1AS message timestamp points, an alignment marker (AM) could also separate the SFD and the symbol after the SFD, creating an even greater discrepancy between their corresponding timestamps.

# Message Timestamp Point (1)

---

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.
- “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 SFD** arrived at the MDI input on the lane with the smallest buffer delay.”
- “For a PHY that includes an FEC function, the transmit and receive path data delays may show significant variation depending upon **the position of the SFD** within the FEC block. However, since the variation due to this effect in the transmit path is expected to be compensated by the inverse variation in the receive path, it is recommended that the transmit and receive path data delays be reported as if **the SFD** is at the start of the FEC block.”

# Message Timestamp Point (2)

---

However...

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

- “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 Different 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 =  $X - T_{\text{SFD}}$ 
        - $T_{\text{SFD}}$  is the time occupied by a SFD symbol
        - creates a constant time error  $c\text{TE} = T_{\text{SFD}}$
- Alignment marker could also separate the SFD and the symbol after the SFD, creating an even greater discrepancy between their corresponding timestamps



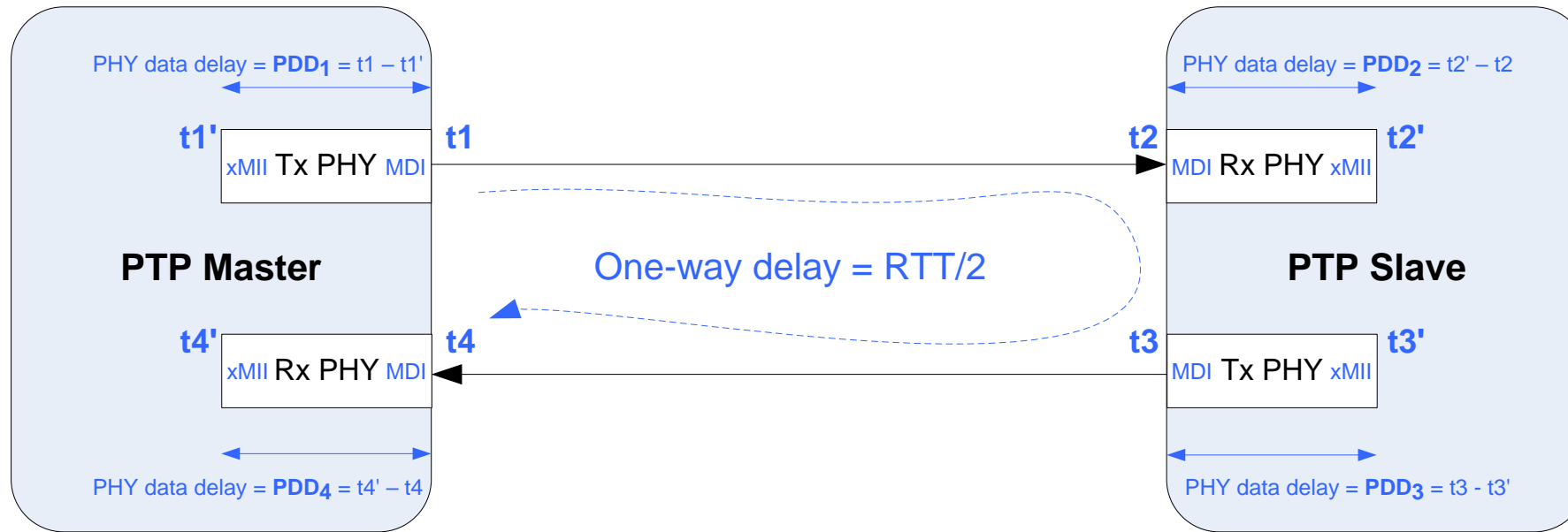
# AM and IDLE Insertion/Removal

---

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

# PTP Time Distribution with IEEE 802.3



$t_1', t_2', t_3',$  and  $t_4'$  are captured at the IEEE 802.3 xMII interfaces

$t_1, t_2, t_3,$  and  $t_4$  are derived from  $t_1', t_2', t_3',$  and  $t_4'$  using the corresponding PHY **path data delay** ( $PDD_x$ )

$$\begin{aligned} \text{Round-trip time RTT} &= (t_4 - t_1) - (t_3 - t_2) \\ &= ((t_4' - PDD_4) - (t_1' + PDD_1)) - ((t_3' + PDD_3) - (t_2' - PDD_2)) \end{aligned}$$

To get an accurate RTT value, the following PHY **path data delays must be known** for each PTP event message:

- All corresponding  $PDD_x$  or
- $(PDD_1 + PDD_2)$  and  $(PDD_3 + PDD_4)$

# Potential Solution

---

- Specify how AMs are to be handled wrt timestamping
  - If the insertion or removal of AMs and/or Idles in these PCSs affects the transmit or receive data path delay, this effect must be accounted for in the timestamp. In this way, the timestamp operation is performed as if alignment markers are present at the xMII (i.e., as if AM insertion and Idle insertion/removal is performed ahead of the Tx xMII and AM deletion and Idle insertion/removal is performed after the Rx xMII).
  - See more dialogue and text that proposes a similar solution at:  
[http://www.ieee802.org/3/ad\\_hoc/ngrates/public/calls/19\\_0416/nicholl\\_nea\\_01\\_190416.pdf](http://www.ieee802.org/3/ad_hoc/ngrates/public/calls/19_0416/nicholl_nea_01_190416.pdf).



**MICROCHIP**

Questions?



**MICROCHIP**

**Thank You**

---

# Backup Information

# Application Timing Requirements

Classes C and D were added in 2018 for 5G transport applications

- 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: constant time error
  - dTE<sub>L</sub>: low-passed dynamic time error
    - MTIE: Maximum Time Interval Error
    - TDEV: Time Deviation
  - TE<sub>L</sub>: constant time error + low-passed dynamic time error
  - TE: constant time error + unfiltered dynamic time error

Time Error Type	Class	Requirement (ns)
max TE	A	100
	B	70
	C	30
	D	for further study
max TE <sub>L</sub>	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 $\tau$ (s)
dTE <sub>L</sub>	A and B	MTIE = 40	$m < \tau \leq 1000$ (for constant temp)
	A and B	MTIE = 40	$m < \tau \leq 10000$ (for variable temp)
	C	MTIE = 10	$m < \tau \leq 1000$ (for constant temp)
	D	MTIE = for further study	
	A and B	TDEV = 4	$m < \tau \leq 1000$ (for constant temp)
	C	TDEV = 2	
D	TDEV = for further study		

# Resulting Performance vs Target Performance

- Target Max|TE| = 30ns for class C Telecom Boundary Clock
  - In a system, there are other sources of TE, in addition to those from timestamping, that use up the allowance

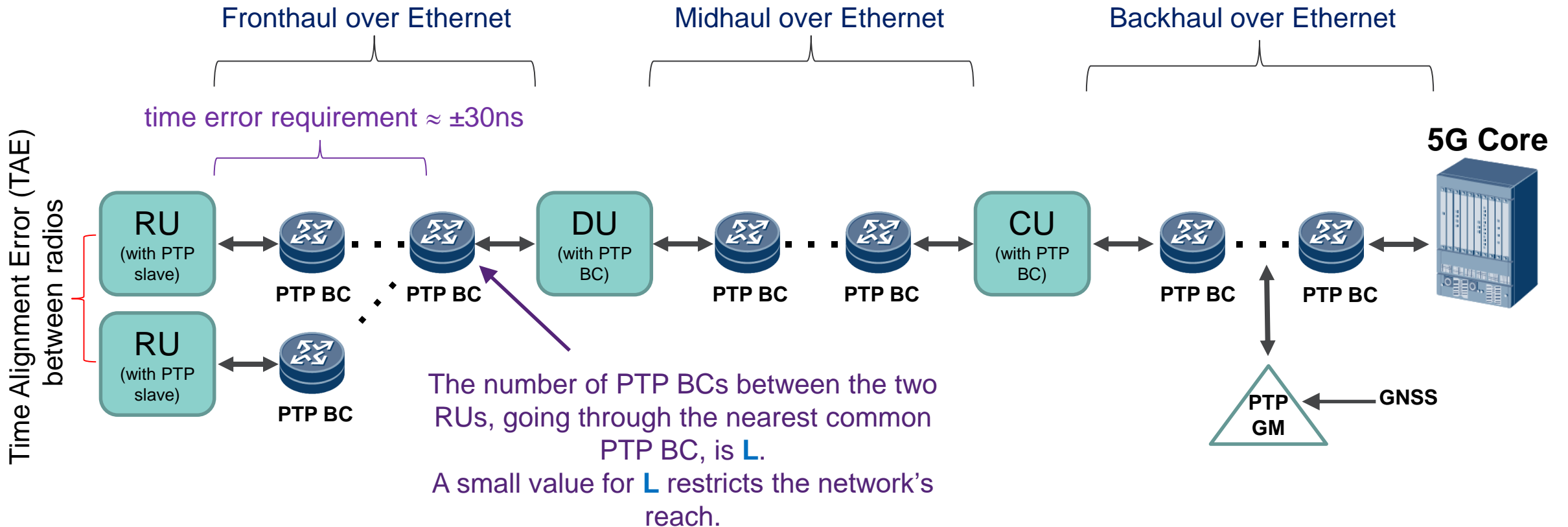
Ethernet Rate	Path Data Delay Variation per Tx/Rx Interface (ns)				Total TE per Tx or Rx Interface (ns)	Path Data Delay Variation Contribution to Max TE , per PTP Boundary Clock (ns)
	mismatched SFD timestamp point	Idle insert/remove (per Idle)	AM insert/remove	Lane Distribution		
GE	8	16	N/A	N/A	24	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

100GE is very important for C-RAN



# Transport Timing for 5G Centralized-RAN (C-RAN)

- C-RAN separates the BBU into “centralized” elements (Distributed Units (DUs) and Central Units (CUs)), allowing their resources to be efficiently shared between the Remote Units (RUs, radios)
- 5G mmWave NR (New Radio) has short reach (i.e. are densely packed) and high capacity
  - These qualities cause a need for a substantial fronthaul network (i.e. more timing hops) to connect RUs to their DUs



# Application Timing Consequences

---

- ITU Q13/SG15 WD13-25 shows why improved PTP performance is needed:
  - For radio time alignment error (TAE) of 260ns (see “TAE” in the figure on slide 9):
    - With all Class B Boundary Clocks everywhere, including in the RUs,  
 $L = 1$  (only direct connect can satisfy requirements!)
    - With all Class C Boundary Clocks in network and class B Slave Clocks in the RUs,  
 $L = 5$
    - With all Class C Boundary Clocks in network and “class C-like” Slave Clocks in the RUs,  
 $L = 7$
    - If results were expanded to use class D Boundary Clocks in network and “class C-like” Slave Clocks in the RUs,  $L > 17$
- To build a practical C-RAN network for 5G applications, PTP Clock performance should be Class C or better