Discussion of Comment R1-2 in p802.3cx

Alon Regev, Keysight Technologies

Nov 28, 2022

802.3cx Introduction

1588-2019 latency view

- IEEE Std 1588-2019 timestamps are supposed to be aligned to the reference plane (where the reference plane is "*the boundary between PTP Instance hardware and the PTP Network medium*")
  - I believe that the 1588 "reference plane" maps to the 802.3 "MDI" (Media Dependent Interface, effectively the connector where the Ethernet fiber/cable/wire in plugged-in)
- IEEE Std 1588-2019 allows the timestamp to be captured at a different point than the reference plane but recommends (using "should") that the time be corrected, specifically:

  The implementation-specific corrections of the captured timestamps are specified as follows:

  <egressProvidedTimestamp> = <egressCapturedTimestamp> + <implementation-specific correction of egressLatency and messageTimestampPointLatency>

  <ingressProvidedTimestamp> = <ingressCapturedTimestamp> – < implementation-specific correction of ingressLatency and messageTimestampPointLatency>

  (see IEEE Std 1588-2019 subclause 7.3.4.2)
- IEEE Std 1588-2019 Figure 26 (shown to the right) shows the relationship between these different points

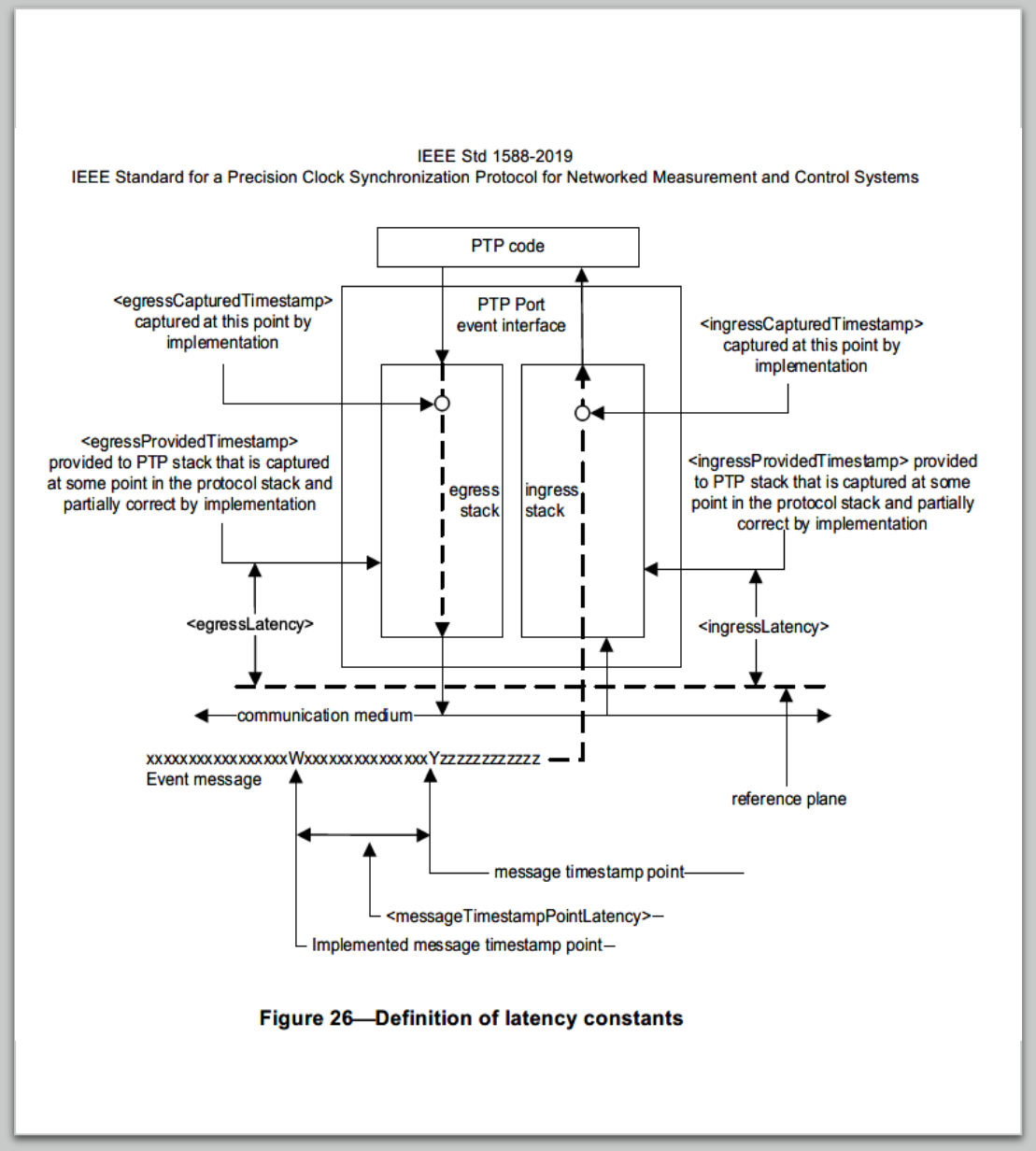


Figure 26—Definition of latency constants

IEEE 802.3-2018 / p802.3cx latency view

- IEEE Std 802.3-2018 clause 90 defines how this is done in IEEE Standard 802.3
  - subclause 90.5 defines a generic Reconciliation Sublayer (gRS) with functions to detect the SFD and timestamp packets

- Figure 90-3 in IEEE Std 802.3 shows the relationship between this gRS (the timestamping capture point) and the MDI (the reference plane)
  - IEEE p802.3cx corrects “data delay” to “path data delay”

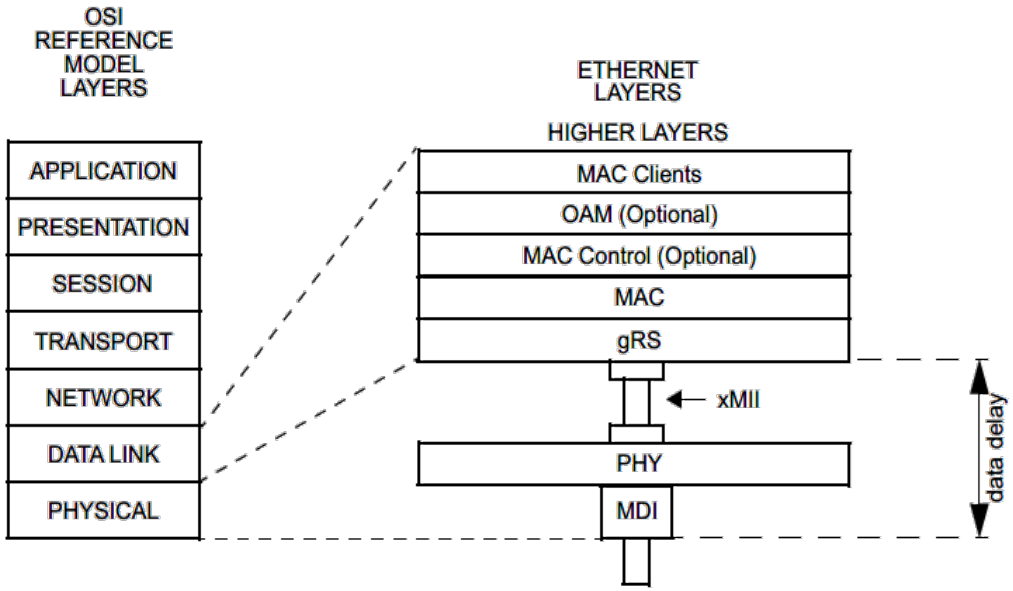
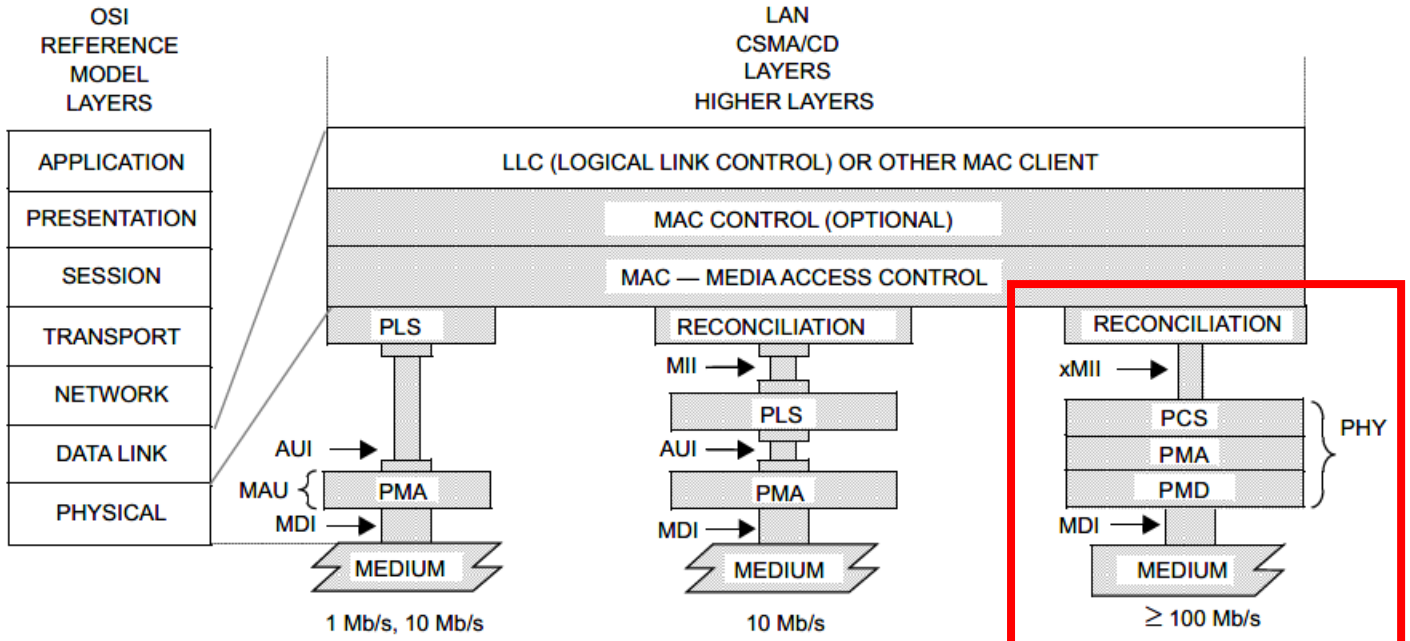


Figure 90–3—Data delay measurement

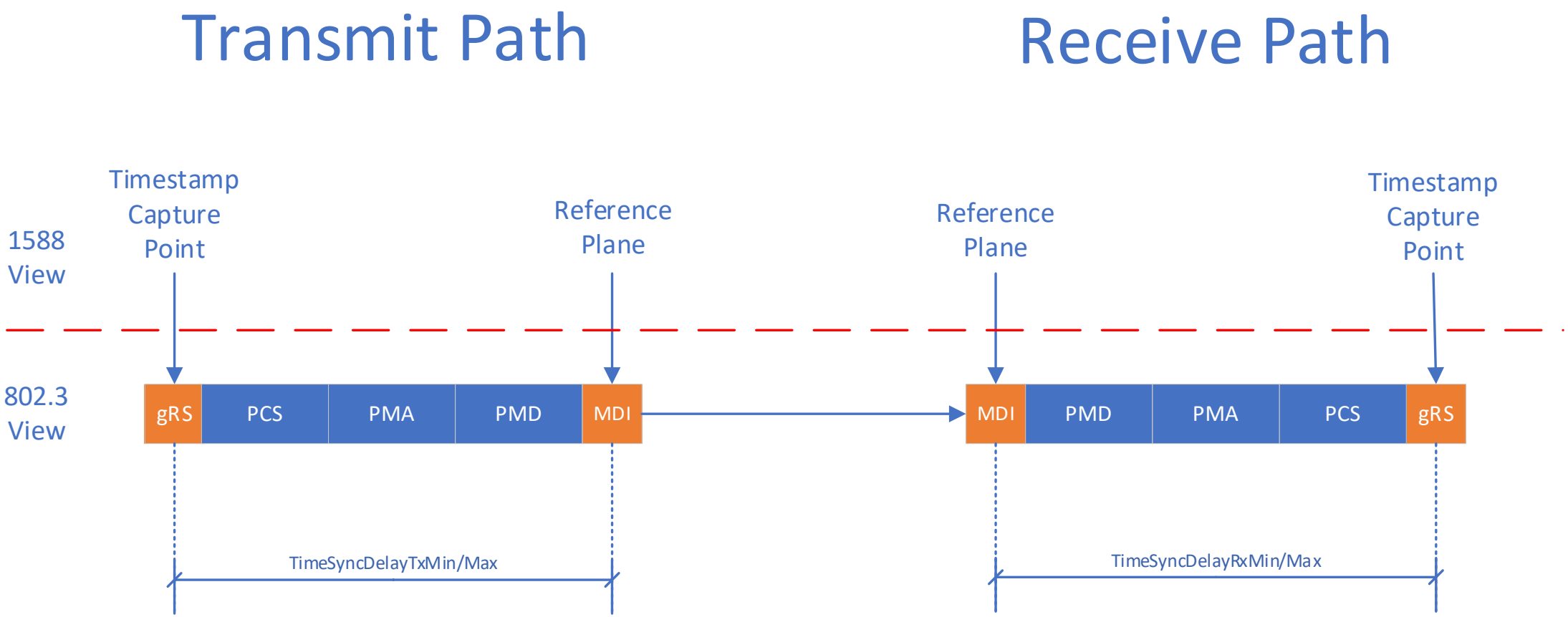
IEEE Std 802.3 shows some mode details of the layers between the Reconciliation Sublayer and the MDI



NOTE—In this figure, the xMII is used as a generic term for the Media Independent Interfaces for implementations of 100 Mb/s and above. For example: for 100 Mb/s implementations this interface is called MII; for 1 Gb/s implementations it is called GMII; for 10 Gb/s implementations it is called XGMII; etc.

Figure 1–1—IEEE 802.3 standard relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model

Relationship of 1588 terms to 802.3 terms



802.3: Different layers in different PHYs

- Some PHY types do not have a PMD layer (this is the case for most twisted-pair “copper” PHYs)
- An example of this is shown in IEEE Std 802.3 Figure 44-1 that shows the different layering for different 10GE PHY types
- Note that 10GBASE-T which is a copper PHY has “AN” (Auto-Negotiation) instead of a PMD
  - In my view, AN is not a layer in the PHY but rather a separate function within the PHY that is parallel to the PMA and PCS…

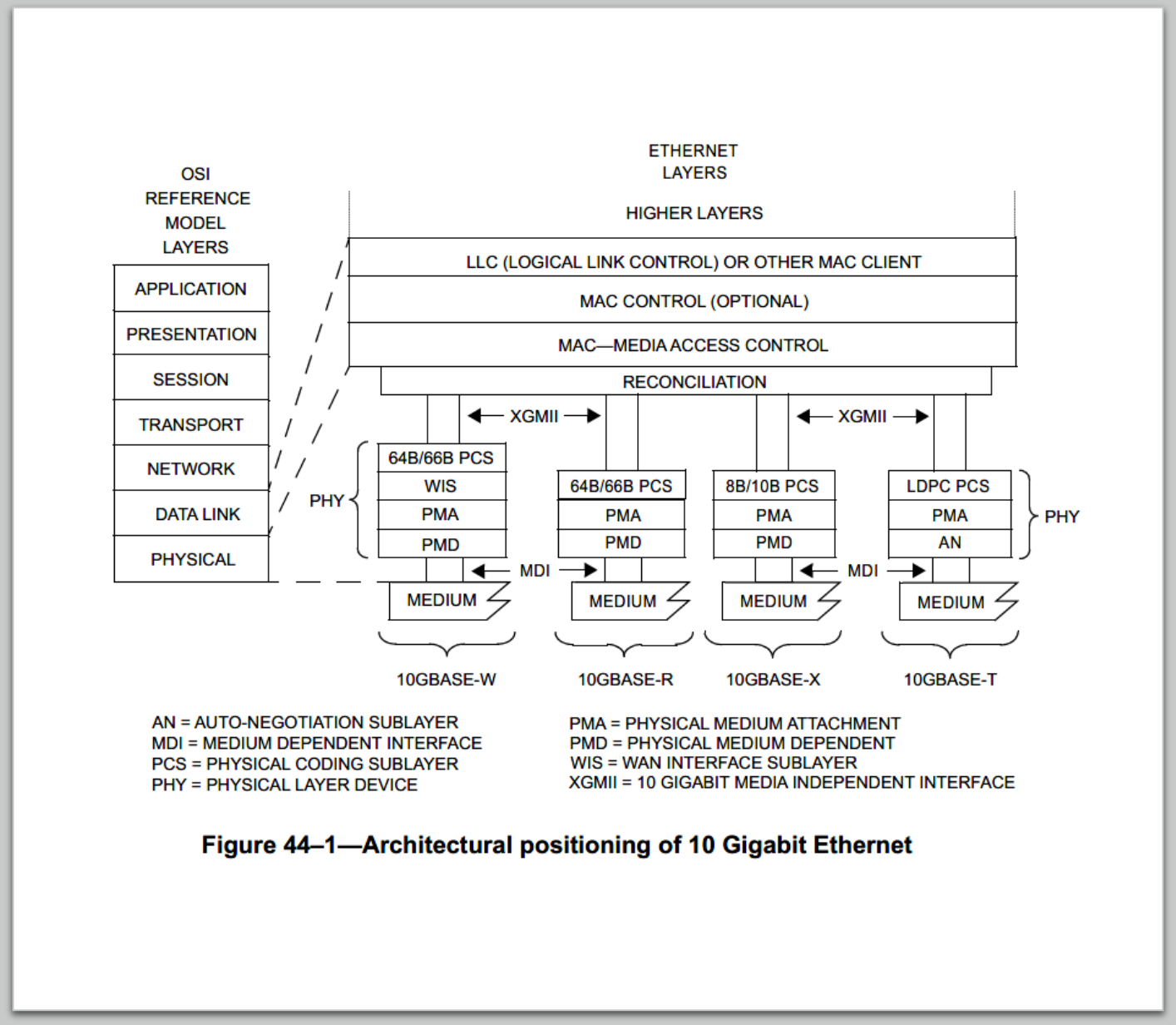
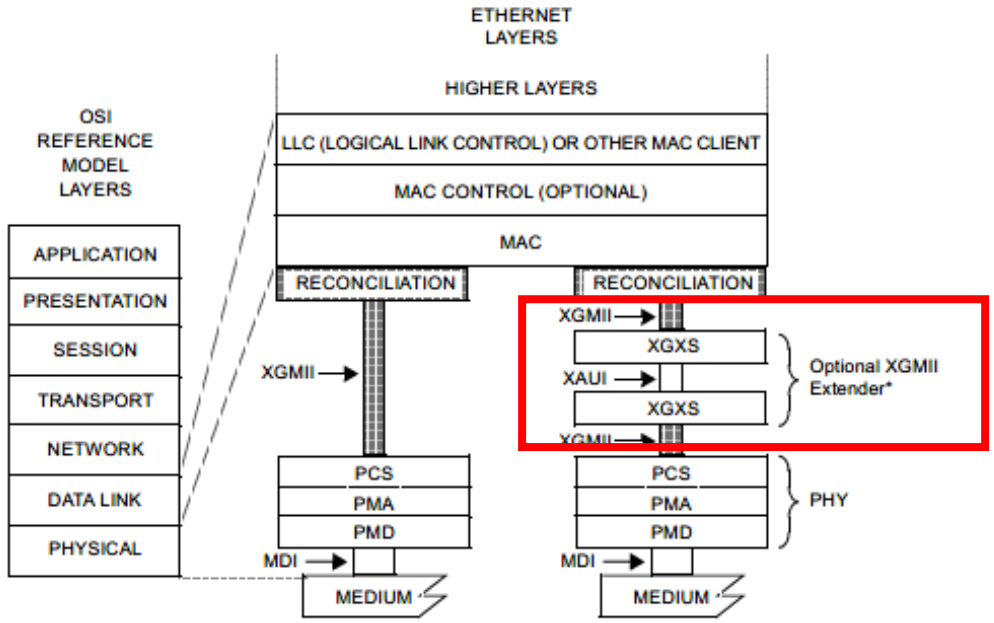


Figure 44–1—Architectural positioning of 10 Gigabit Ethernet

802.3: More Sublayers…

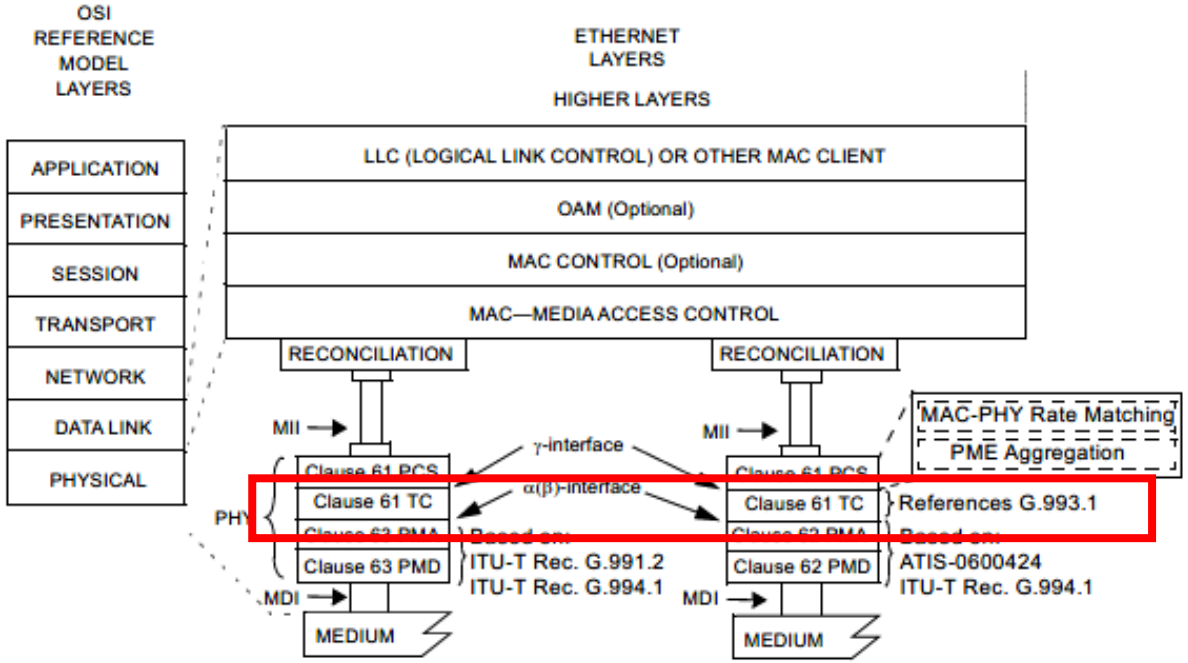


MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE

PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
XAUI = 10 GIGABIT ATTACHMENT UNIT INTERFACE
XGMII = 10 GIGABIT MEDIA INDEPENDENT INTERFACE
XGXS = XGMII EXTENDER SUBLAYER

*specified in Clause 47

Figure 46–1—XGMII relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3 Ethernet model



MDI = MEDIUM DEPENDENT INTERFACE
MII = MEDIA INDEPENDENT INTERFACE
OAM = OPERATIONS, ADMINISTRATION & MAINTENANCE
TC = TRANSMISSION CONVERGENCE

PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
PME = PHYSICAL MEDIUM ENTITY

Figure 61–1—Relation of this clause to other standards

1588 Message Timestamp Point vs. 802.3 DDMP

- 1588
  - In clause 3.1.33: **message timestamp point**: A point within a PTP event message serving as a reference point in the message. A timestamp is defined by the instant a message timestamp point passes the reference plane of a PTP Instance.
  - in clause 7.3.4.1: "Unless otherwise specified in a transport-specific annex to this standard, the message timestamp point for a PTP event message shall be the beginning of the first symbol after the start of frame delimiter.“
- 802.3cx
  - In clause 90.4.2: The path data delay of a packet is measured using a specific location in the packet called the DDMP. The DDMP is either the beginning of the start-of-frame delimiter (SFD) or the beginning of the first symbol after the SFD in the packet, as selected by registers 3.1813.13 and 5.1813.13. The term 'first symbol after the SFD' denotes the first octet after the SFD when referencing an xMII.

“symbol” in IEEE Std 1588-2019

- IEEE 1588-2019 does not define “symbol”

- In IEEE Std 1588-2019 "message timestamp point" is defined in clause 7.3.4.1 as: "Unless otherwise specified in a transport-specific annex to this standard, the message timestamp point for a PTP event message shall be the beginning of the first symbol after the start of frame delimiter.“
  - This seems to match the definition used in P802.3cx D3.1.
  - The term “symbol” is unclear in this clause in IEEE 1588: is it referring to a serial symbol (bit), octet, or PHY-type-specific symbol?

- As used elsewhere in IEEE 1588, symbol refers to UTF-8 characters, octets, bits, or 10-bit 8B10B symbol

“symbol” in IEEE Std 802.3-2018

- In clause 1.4.466, IEEE Std. 802.3-2018 defines "Symbol“ as "Within IEEE 802.3, the smallest unit of data transmission on the medium. Symbols are unique to the coding system employed. For example, 100BASE-T4 and 100BASE-T1 use ternary symbols; 10BASE-T uses Manchester symbols; 100BASE-X uses binary symbols or code-bits; 100BASE-T2 and 1000BASE-T uses quinary symbols. For 1000BASE-X PMDs operating at 1.25 GBd, a symbol corresponds to a code-bit after the 8B/10B encoding operation i.e., has the duration of 0.8 ns. For 10GBASE-R PMDs operating at 10.3125 GBd, a symbol corresponds to a code-bit after the 64B/66B encoding operation i.e., has the duration of approximately 0.097 ns.“

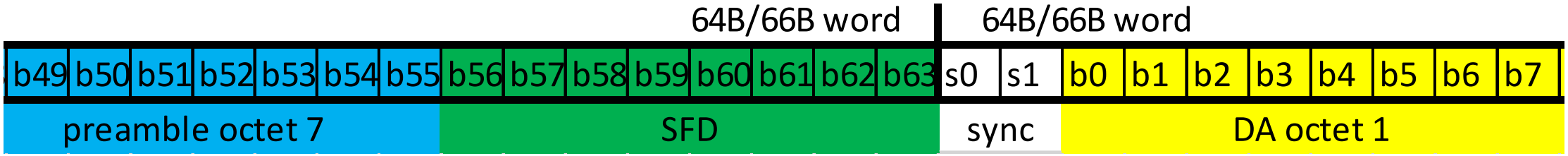
DDMP at the xMII has no issue

- IEEE 802.3 defines the following xMII layers: MII, GMII, XGMII, 25GMII, XLGMII, CGMII, 200GMII, 400GMII
- The xMII groups the serial MAC data in 4 bit (MII), 8 bit (GMII), 32-bit (XGMII, 25GMII), or 64-bit (XLGMII, CGMII, 200GMII, 400GMII) words
- As the xMII is a parallel interface, my interpretation of “symbol” at the 802.3 xMII layer is the entire 4 / 8 / 32 / 64 line (plus control lines) during one clock cycle (i.e. 64-bits on CGMII)
- On an originating transmitter, the SFD will always end on a symbol boundary (as the preamble plus SFD take 8 bytes), so there is no ambiguity in the term “first symbol after the SFD”
- On the receiving side
  - due to “preamble shrinkage”, I thought the SFD and first octet after SFD could be in the same symbol
  - I have convinced myself that preamble shrinking could only occur with 1G and lower speeds (i.e. MII and GMII), and as such the SFD and first octet after SFD would always be in 2 distinct symbols

DDMP at the MDI

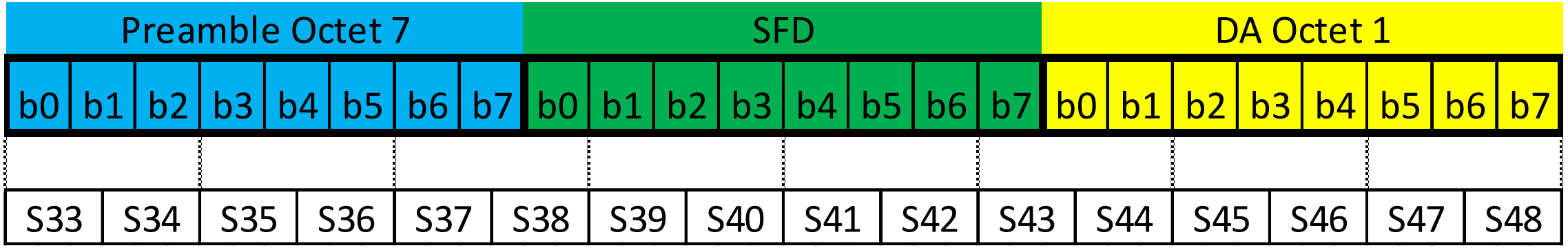
- The symbol following the MDI can be a control symbol (not a data symbol). It is not clear if this control symbol should be used or the symbol that corresponds to the first data bit should be used
  - Example: 10GBASE-R uses 64B/66B encoding. If SFD is at the end of a 64B66B block, it will be followed by a 2-symbol sync header before the next “data” symbol
- At the MDI, a single symbol or set of symbols may represent a plurality of bits. Examples:
  - Clause 96 (100BASE-T1) PAM3
    - 3 bits of data are converted to 2 symbols of PAM3
    - First data bit following SFD will always be in a symbol covering the last bit of SFD and the fist two bits of data. First bit of SFD will always be in a symbol covering the last 2 bits of preamble prior to SFD as well as the first bit of SFD.
  - Clause 126 (2.5GBASE-T and 5GBASE-T) PAM16
    - After 64B65B and LDPC each 4 bits of data are mapped to a PAM16 symbol
    - Due to the use of 64B65B encoding, SFD and data can start at any bit offset within the PAM16 symbol
    - LDPC blocks are aligned to symbol boundaries, so if the recommendation to use start of FEC block is used, the issues are avoided

Example: 10GBASE-R 64B/66B



- No ambiguity as to start of SFD symbol (b56 in first 64B/66B word)
- “First symbol after the SFD” is ambiguous
  - Could be s0 symbol in second 64B/66B word
  - Could be b0 symbol in second 64B/66B word

Example: 100BASE-T1 PAM3



Current text is ambiguous as to which symbols to use

- Potential symbols for SFD
  - S37 – first symbol of 2-symbol set that contains the first bit of the SFD
  - S38 – second symbol of 2-symbol set that contains the first bit of the SFD
  - S39 – first symbol that contains only SFD bits
- Potential “first symbol after the SFD”
  - S43 – first symbol of 2-symbol set containing the first bit of DA
  - S44 – second symbol of 2-symbol set containing the first bit of the DA
  - S45 – first symbol not containing any part of the SFD

Why is this an issue

- If both link-partners select the same symbol for the measurement of the Tx and Rx path delays then there is no problem
- If the two link partners (or if the Tx and Rx of the same device) use different symbols for the measurement of the Tx and Rx path delays then this will cause a time shift
- Which symbol is chosen by the spec doesn’t matter so much; what matters is the precise definition of the DDMP

Does this only apply to MDI

- As the spec provides path delay measurements for the PMA/PMD, WIS, PCS, XS, and TC, these issues can occur at any point along the path where symbols are not always aligned to power of 2 data bit boundaries and where a symbol can represent control as well as data.

Potential fix #1

- change every instance of "first symbol after SFD" to "the symbol representing the first data bit after the SFD".
  - Note that this is different than the text originally proposed in the comment: “the symbol containing the first data bit after the SFD”
- The comment proposed adding text to annex 90A explaining how to interpret this for different types of symbols, but I think this would be better done in clause 90.7. Here is my proposed text (needs work):
  - “For a PHY that atomically maps N bits to M symbols (such as 100BASE-T1 in Clause 96 that uses 3B2T to convert 3 bits to 2 symbols), the path data delay measurement should use the start of the first symbol as the time event used for any of the N bits encoded in the M symbols.”

Potential fix #2

- Add text to section 90.7 explaining how to interpret “first Symbol after SFD” as well as how to handle different types of symbols; here is my proposed text (needs work):
  - The “first symbol after SFD” should be interpreted to mean the symbol representing the first data bit after the SFD rather than any symbols representing sync or control information. When applying to a PHY that atomically maps N bits to M symbols (such as 100BASE-T1 in Clause 96 that uses 3B2T to convert 3 bits to 2 symbols), the path data delay measurement should use the start of the first symbol as the time event used for any of the N bits encoded in the M symbols .”

Potential fix #3 (from Richard Tse)

- EgressLatency, ingressLatency (Tx/Rx path data delays)
  - Path data delays are constant values:
    - Equals a constant value for every xMII bit for PHY functions that have constant delays
    - Equals max potential Tx path data delay of *any* xMII bit for Tx PHY functions with varying mirrored-delays
    - Equals min potential Rx path data delay of *any* xMII bit for Rx PHY functions with varying mirrored-delays
  - TX/RX_NUM_BIT_CHANGE used for non-deterministic changes (affects PCS sublayer)
- Timestamp generation at measurement plane (DDMP at xMII)
  - DDMP is the first bit of SFD or first bit of octet after SFD on the xMII
    - Timestamp for TS_TX/RX.indication corresponds to the time of the first bit of the xMII
    - Each subsequent xMII bit has an additional bit time's worth of delay
    - Do we need another parameter in the TS_TX/RX.indication parameter to identify which bit in the xMII word is the DDMP?
- Timestamp generation at reference plane (DDMP at MDI???)
  - 802.3cx models the timestamp generation at the reference plane with a bit-level timestamp at the xMII and a PHY path data delay so identification of DDMP at MDI is not needed
    - Timestamp at reference plane = xMII timestamp +/- path data delay

Consideration for 8B/10B

- This is a case of a PCS sublayer with a delay that is constant
- 8-bits at GMII are transformed “instantly” to/from a 10-bit code-group
- For path data delay measurement, “beginning of DDMP” corresponds to bit <7> at GMII and to bit *a* at MDI

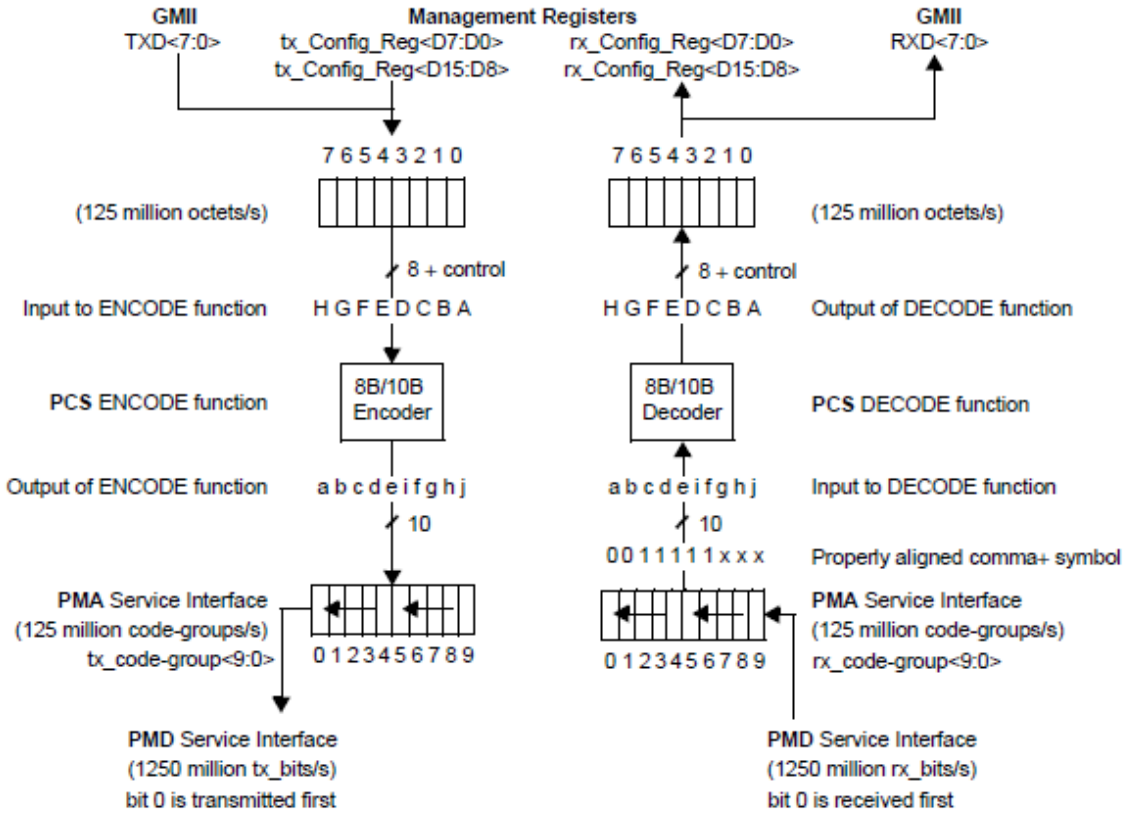
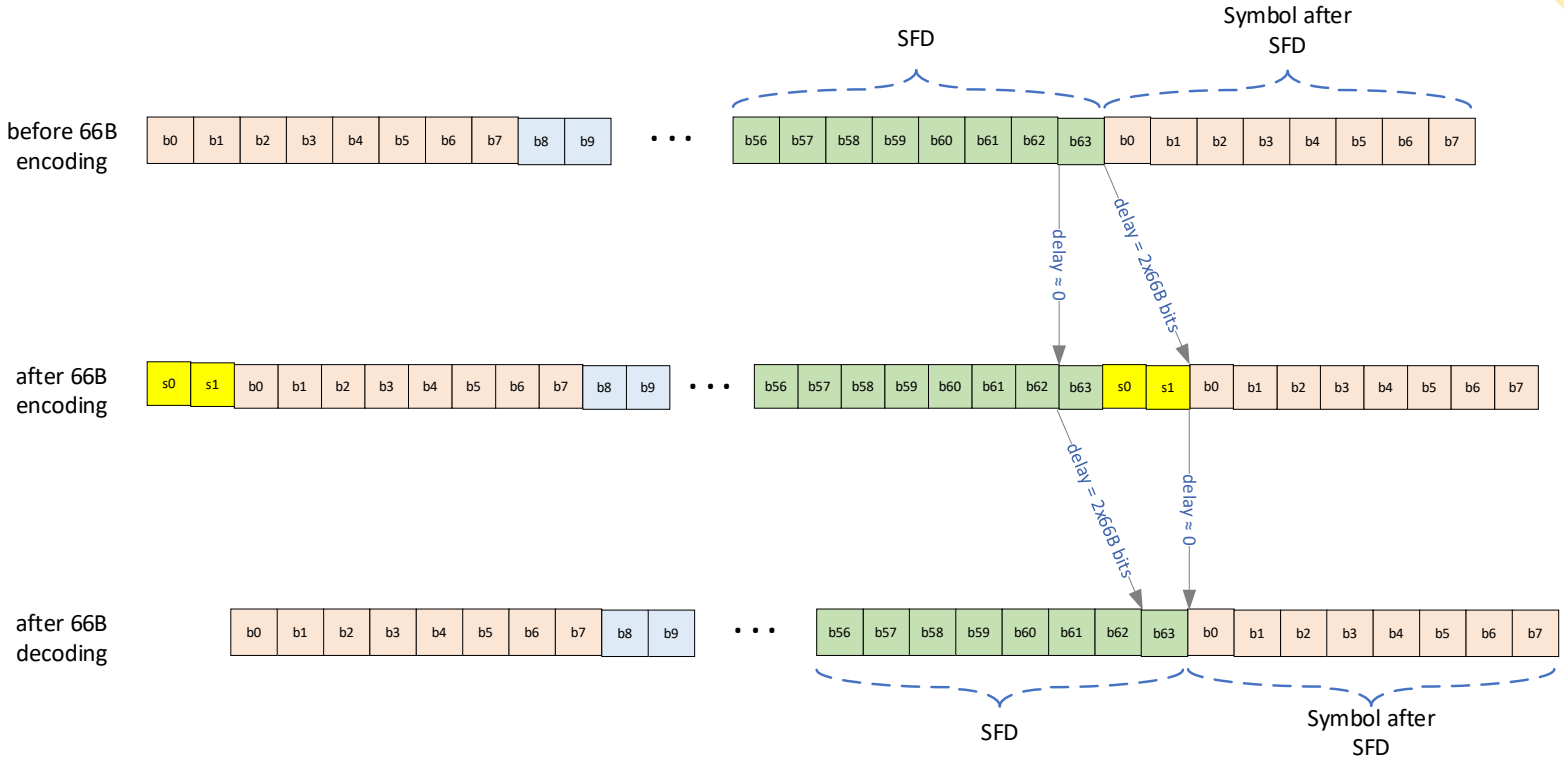


Figure 36–3—PCS reference diagram

Consideration for 64B/66B

- Insertion of 66B sync bits is like insertion of FEC overhead
  - bit b0, placed immediately after the sync bits, experiences the most Tx path data delay
  - bit b63, placed immediately before the sync bits, experiences the least Tx path data delay
  - Tx path data delay and Rx path data delay are mirrors of each other, resulting in equal delay across encoder + decoder for all bits
- Using the same method as FEC (i.e., allocate max delay to Tx and min delay to Rx), we allocate 2 × 66B bit delay to the Tx encoder and 0 to the Rx decoder
  - This insinuates that a data bit, not a sync bit, is timestamped at the MDI
  - However, because 802.3 timestamps DDMP at the xMII and models the PHY delays as constants, the definition of the DDMP at the MDI is not important



Consideration for 100BASE-T1 PAM3

- Total PAM3 encoding + decoding delay = 1.5 bits for every bit
- Max potential Tx delay = 1.5 bits
- Min potential Rx delay = 0 bits
- With bit-level timestamp at xMII and Tx/Rx path data delays, the identification of which bit is the DDMP at the MDI is not needed