Annex XX (informative) Timestamping Accuracy Considerations

XX.1 High Accuracy Timestamping - Introduction

This annex provides informative data on high accuracy timestamping for applications pertaining to clause 90. The following discussion presumes that the reader is already astute in the concepts used by IEEE Std 1588 and IEEE Std 802.1AS.

XX.2 Background on Timing Errors

Ambiguity in the specification prior to IEEE Std 802.3cx could lead to implementations that are not compatible with respect to timestamping, resulting in timing errors when such implementations interact with each other.

A timing error can result when implementations do not account for a varying PHY path data delay in the same manner. Examples of PHY functions that cause variation in the PHY path data delay are AM/CWM insertion/deletion, Idle insertion/deletion, and multi-PCS lane distribution/merging.

A timing error can also result when implementations do not use the same message timestamp point. As noted in 90.7, IEEE Std 802.3cx allows for two message timestamp points though it recommends just one of them.

Table XX-1 shows examples of potential timing errors that could be generated by each of the aforementioned causes.

Table XX-1 – Potential Timing Errors

Ethernet Rate	Potential Timing Error per Tx/Rx Interface (ns)			
	Mismatched Message Timestamp Point[1]	AM/CWM Insertion/Deletion	Idle Insertion/Deletion[2]	PCS Lane Distribution/Merging[3]
GE	8	N/A	16	N/A
10GE	0.8	N/A	3.2	N/A
25GE	0.32	2.56	1.28	N/A
40GE	0.2	6.4	1.6	4.8
100GE	0.08	12.8	0.64	12.16
200GE	0.04	2.56	0.32	N/A
400GE	0.02	2.56	0.16	N/A

1. The value shown only accounts for the time between the two message timestamp points when they are adjacent. See XX.3 for other factors that can affect these values.
2. The value shown corresponds to just one idle.
3. The value shown is only for PCS lane distribution/merging. FEC lane distribution/merging was already well specified before IEEE Std 802.3cx.

XX.3 Considerations for Use of Different Message Timestamp Points

The size of potential timing errors that can result from the use of different message timestamp points are given in Table XX-1. These values represent the timing error due to the time difference between the

message timestamp point options when they are adjacent. In a non-802.3cx-compliant implementation, additional error could result if the two message timestamp point options are separated due to:

- Insertion of bytes for AM or CWM functions
- Multi-PCS lane distribution

IEEE Std 802.3cx-compliant implementations would always only suffer a time error of one byte time between the two message timestamp point options because:

- The effect of AM, CWM, or Idle insertion and deletion are accounted for, using the Tx_num_blk_change and Rx_num_blk_change signals.
- The multi-PCS lane path data delay is modelled as a constant value for all lanes.

XX.4 Considerations for Multi-PCS Lane Functions

The size of potential timing errors that can result from implementations that are not compliant to IEEE Std 802.3cx for multi-PCS lane operations are given in Table-XX-1. This error might vary depending on how an implementation deals with its PCS lane-to-lane distribution and lane-to-lane merging delays.

The general concept used by IEEE Std 802.3cx to accommodate the delay variation of the multi-PCS lane distribution/merging operation is explained in XX.7. This concept, which was originally used to handle the varying delays of IEEE 802.3 multi-FEC lane distribution/merging operations, takes advantage of the fact that the sum of the intrinsic delay variation of the Tx multi-PCS lane distribution operation and of the intrinsic delay variation of the Rx multi-PCS lane merging operation is a predetermined constant for a given multi-PCS lane function.

The concept allows the intrinsic delay variations to instead be treated as a constant and thus, the static TimeSync PCS transmit path data delay register and TimeSync PCS receive path data delay register can continue to be used with high accuracy timestamping even when multi-PCS lane functions are present.

- The TimeSync PCS transmit path data delay register uses the greatest PCS lane distribution delay as its constant value (i.e., the Tx PCS lane distribution delay for lane 0).
- The TimeSync PCS receive path data delay register uses the smallest PCS lane merging delay as its constant value (i.e., the Rx PCS lane merging delay for lane 0).
- Because the PCS transmit path data delay is modelled as a constant value, the minimum and maximum TimeSync PCS transmit path data delay registers are, in an ideal implementation, the same value.
- Because the PCS receive path data delay is modelled as a constant value, the minimum and maximum TimeSync PCS receive path data delay registers are, in an ideal implementation, the same value.

The above usage of the TimeSync PCS transmit/receive path data delay registers is consistent with that for the Tx multi-FEC lane distribution and Rx multi-FEC lane merging operations.

XX.5 Considerations for AM/CWM and Idle Rate Adaptation Functions

The size of potential timing errors that can result from implementations that are not compliant to IEEE Std 802.3cx for AM/CWM operations and Idle rate adaptation functions are shown in Table XX-1. This time error can occur because the AM, CWM, or Idle insertion and deletion operations cause an instant change in the PCS path data delay at the time of the event. Unlike other PHY functions like multi-PCS lane distribution/merging and multi-FEC lane distribution/merging, these events do not generate PHY path data delay variations that can be pre-determined and the Tx path data delay variation is not mirrored by the Rx path data delay variation.

The IEEE Std 802.3cx implementation accounts for each of these path data delay variations by using the Tx_num_blk_change and Rx_num_blk_change signals (see subclauses N and M). Because these signals allow the timestamp to compensate for the change in the path data delay, the static TimeSync PCS transmit path data delay register and TimeSync PCS receive path data delay register can continue to be used with high accuracy timestamping even when AM, CWM, and Idle rate adaptation functions are present.

Examples on the use of Tx_num_blk_change and Rx_num_blk_change are given in XX.5.1 and XX.5.2, respectively.

[Editorial note: add figures to XX.5.1 and XX.5.2 to help illustrate the examples]

XX.5.1 Example use of Tx_num_blk_change

1. Scenario without AM, CWM, or idle rate adaptation event:
    - Word arrives at Tx xMII at time = T1
    - Tx PCS path data delay = PDD1
        - The constant value, PDD1, is programmed into the TimeSync PCS transmit path data delay registers
    - Calculated Tx departure timestamp = T1 + PDD1
2. Scenario with AM, CWM, or idle rate adaptation event:
    - Word arrives at Tx xMII at time = T1
    - Tx PCS path data delay with AM, CWM, or Idle rate adaptation event = PDD1 + Tx_num_blk_change*(nanoseconds/block)
        - The PHY's delay changes due to AM, CWM, or Idle rate adaptation event
    - Calculated Tx departure timestamp = T1 + PDD1 + Tx_num_blk_change*(nanoseconds/block)
3. Scenario using Tx_num_blk_change to account for the path data 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 rate adaptation event (as if it happened before the Tx xMII, per 90.7)

- Tx PCS path data delay = PDD1
    - The constant value, PDD1, programmed into the TimeSync PCS transmit path data delay registers does not change.
- Calculated Tx departure timestamp = T1 + PDD1 + Tx_num_blk_change*(nanoseconds/block)

XX.5.2 Example use of Rx_num_blk_change

1. Without AM, CWM, or Idle rate adaptation event:

- Word arrives at Rx xMII at time = T1
- Rx PCS path data delay = PDD2
    - The constant value, PDD2, is programmed into the TimeSync PCS receive path data delay registers
- Calculated Rx arrival timestamp = T1 – PDD2

2. With AM, CWM, or Idle rate adaptation event:

- Word arrives at Rx xMII at time = T2
- Rx PCS path data delay with AM, CWM, or Idle rate adaptation event = PDD2 + Rx_num_blk_change*(nanoseconds/block)
    - The PHY's delay changes due to AM, CWM, or Idle rate adaptation event
- Calculated Rx arrive timestamp = T2 – (PDD2 + Rx_num_blk_change*(nanoseconds/block))

3. Scenario using Rx_num_blk_change to account for the path data delay variation:

- Adjusted word arrival time at Rx xMII = T2 - Rx_num_blk_change*(nanoseconds/block)
    - The arrival time at the Rx xMII is modified to reflect the AM, CWM, or Idle rate adaptation event (as if it happened after the Rx xMII, per 90.7)
- Rx PCS path data delay = PDD2
    - The constant value, PDD2, programmed into the TimeSync PCS receive path data delay registers does not change.
- Calculated Rx arrival timestamp = T2 – (PDD2 + Rx_num_blk_change*(nanoseconds/block))

XX.5.3 Considerations for Interoperation with Non-Compliant Implementations

For an implementation that is not compliant to IEEE Std 802.3cx and does not adjust the path data delay value for every AM, CWM, or corresponding idle rate adaptation event, the effect of the resulting timing error can be evaluated to determine if it causes significant degradation in a PTP system's performance. Some observations that might help this evaluation are given below:

- Typically, the probability of an AM/CWM insertion/deletion or a corresponding Idle insertion/deletion affecting the path data delay for a PTP message is random and is small.

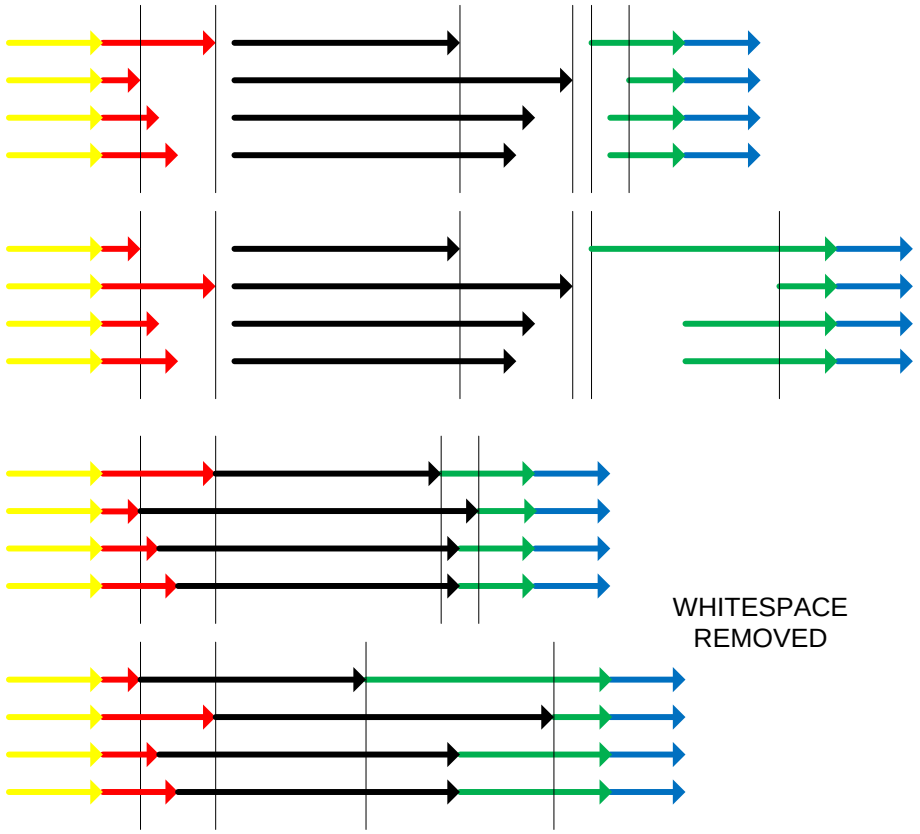
- A low-pass filter, which might be present in a PTP ToD recovery algorithm, could attenuate the effect of the resulting time error.
- Implementations that do not transmit PTP messages in the region of AM/CWM insertions and their corresponding Idle deletions avoid the time error generation at their Tx port but do not guarantee that the corresponding remote Rx port will not generate a time error due to its corresponding Idle insertions/deletions.

XX.6 Considerations for Tx Skew

On a multi-lane interface, the presence of skew at the transmit MDI is difficult to compensate because this skew is entwined with, but independent from the skew of the medium. As shown in Figure XX-1, the transmit skew in series with the medium skew can either be additive or subtractive. Because of this, the per-lane transmit skew values could only be used at the receiver, where the total skew of each lane can be observed at its deskew FIFOs. By using the observed per-lane total skew values at the receiver and the per-lane transmit skew values, the actual skew of each lane of medium can be determined.

To reduce the need for this type of processing, it is recommended that multi-lane transmitters minimize their lane skew at the MDI.

Figure XX-1 – Transmit PMA/PMD Skew in Series with Medium Skew



[Editorial note: This figure needs to be re-drawn to replace the colors with a different method for labeling the PCS lanes at each point, and the vertical lines should be labeled.]

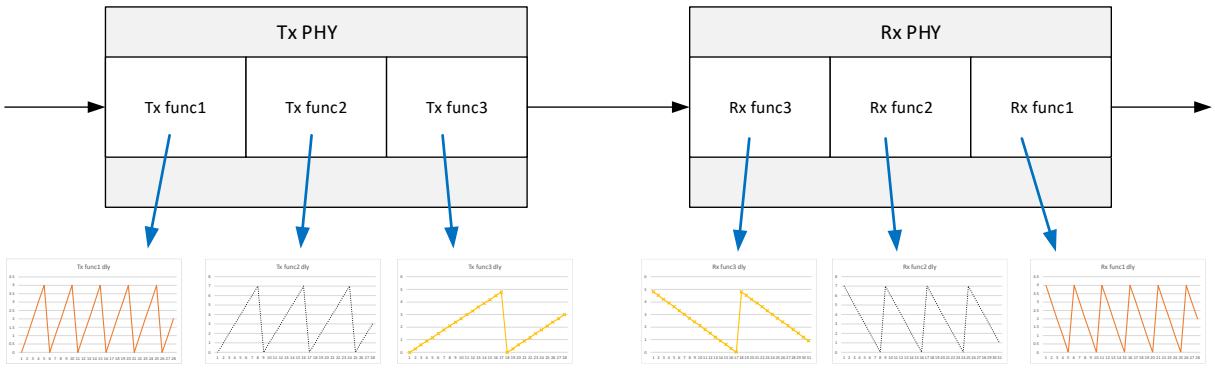
XX.7 General Method for Dealing with Repeating Delay Variation Patterns

Many PHY functions have varying intrinsic delays with the following characteristics:

- The Tx and the Rx intrinsic delay variations follow a known repeating pattern.
- The intrinsic delay variation pattern on Tx is a mirror of the intrinsic delay variation pattern on Rx and the sum of the two intrinsic delays is a known constant value. This is true because the data stream before the Tx function and the Rx stream after the Rx function are identical.

One can take advantage of the above characteristics to simplify an implementation. For example, if a PHY has multiple functions with these characteristics, as shown in Figure XX-2, it can model its path data delay as a constant value instead of the dynamically varying sum of multiple varying delays.

Figure XX-2 – PHY with Cascaded Functions with Varying Delays



For the example shown in Figure XX-2, the sum of the Tx PHY's varying delays is shown in Figure XX-3 and the sum of the Rx PHY's varying delays is shown in Figure XX-4. These sums have no easily discernable pattern and might require a complex implementation to determine the instantaneous path data delay for any chosen bit that corresponds to the message timestamp point of a PTP message in the Ethernet data steam.

Figure XX-3 –Total Delay of Tx PHY with Cascaded Varying Delays

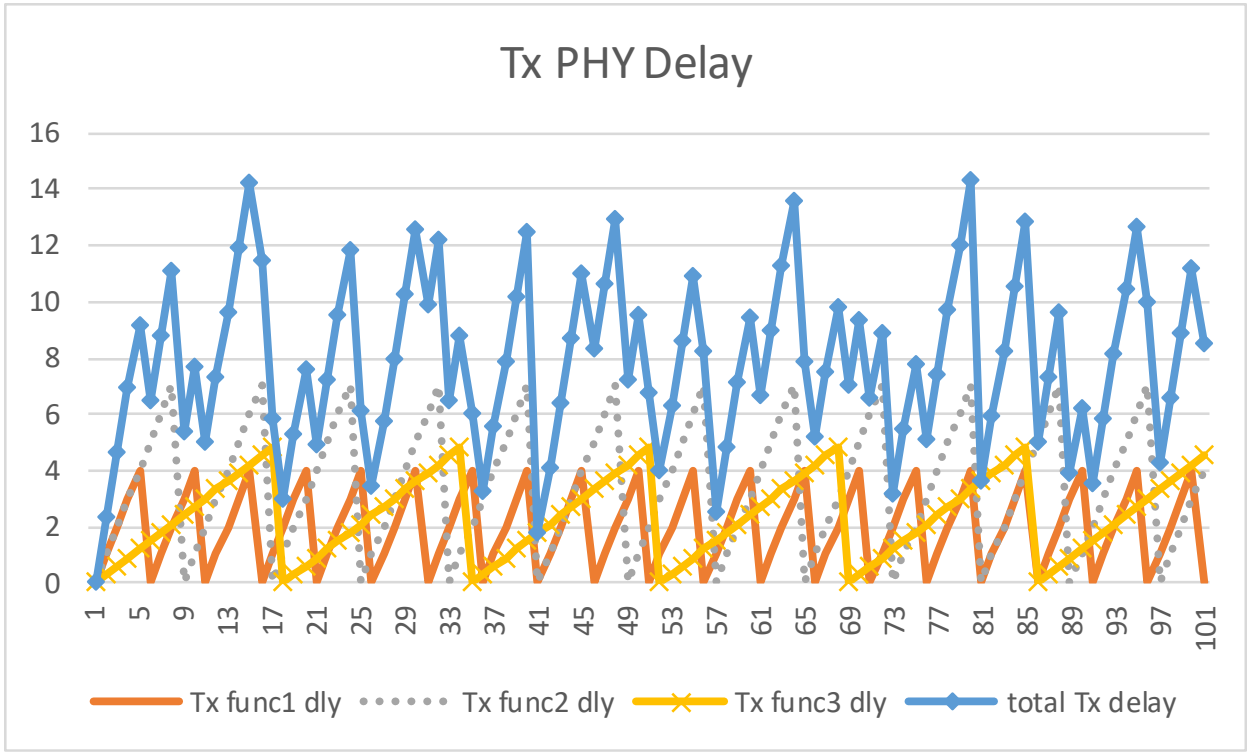
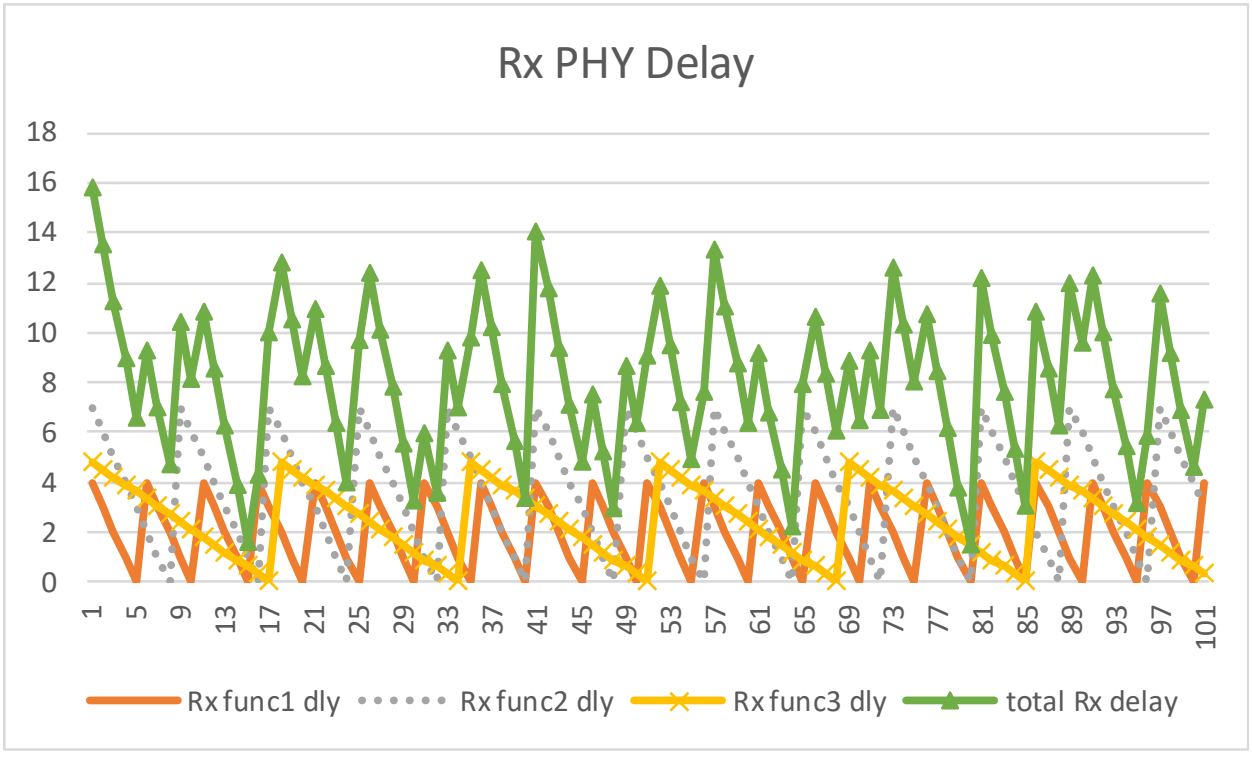


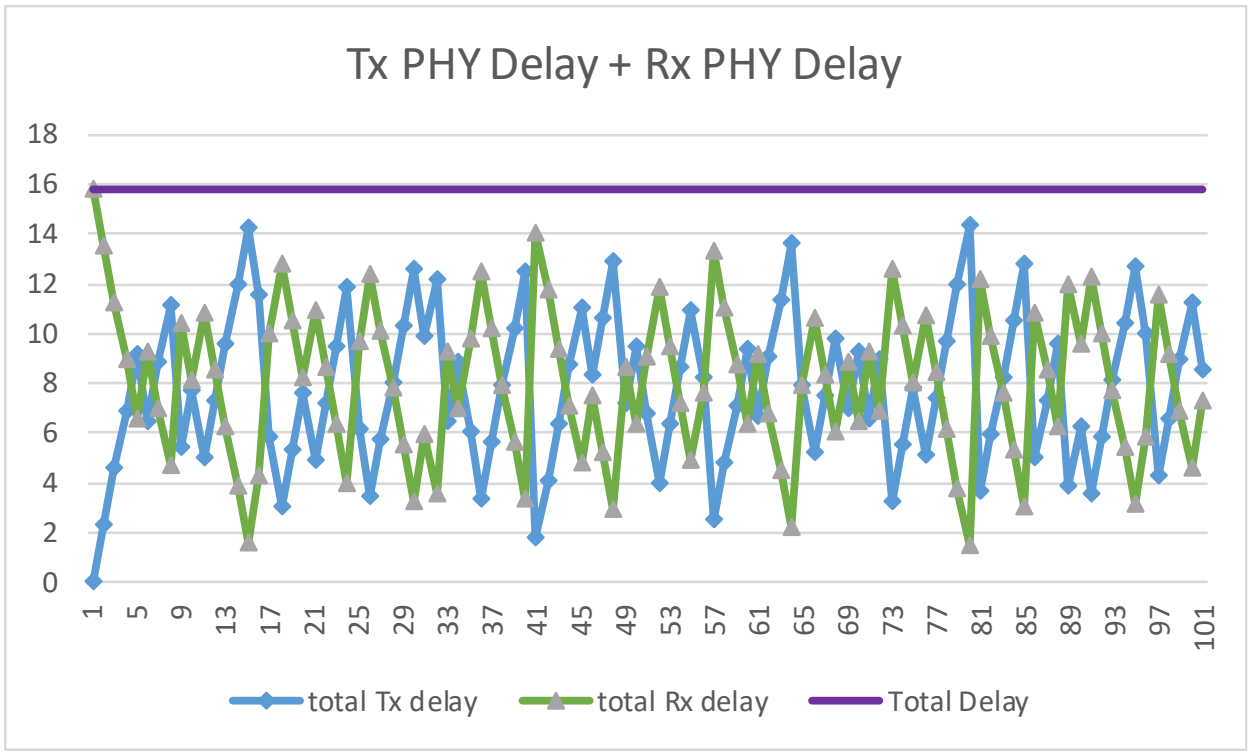
Figure XX-4 – Total Delay of Rx PHY with Cascaded Varying Delays



Because the intrinsic varying delay in the Rx PHY is a mirror of the intrinsic varying delay in the Tx PHY, the total intrinsic delay through both PHYs is a constant, as illustrated in Figure XX-5. This eliminates the need to track the varying delay of the message timestamp point of a PTP message through the Tx PHY and the Rx PHY. Instead, one can standardize what portion of the constant total intrinsic delay is allocated to each of the Tx PHY and the Rx PHY. The allocated portion of the constant total intrinsic delay value is then added to the implementation-specific delay of the corresponding PHY, which is also a constant value, to get the PHY's total delay.

It is recommended to use this method to deal with all varying PHY delays of this nature.

Figure XX-5 – Tx PHY Delay, Rx PHY Delay, and Total Delay



[Editorial note: Provide examples of how this method can be used for existing basic functions such as 64B/66B encoding/decoding, 2x32B to 66B encoding/decoding, 256B/257B transcoding.]