Time Synchronization Clarifications for 802.3dj

Andras de Koos – Microchip Technology

IEEE 802.3dj Task Force

Introduction / Goal

There are a number of unresolved questions/comments against D1.0 involving TimeSync.

This presentation aims to:

1. Give a quick overview of the existing TimeSync content in 802.3.

- Follow-on to David Law's presentation to the ad-hoc in April

  https://www.ieee802.org/3/dj/public/adhoc/optics/0424_OPTX/law_3dj_optx_01a_240425.pdf
- General Timestamping model with the gRS
- Focus on the definition of the path data delay

2. Drive discussion on whether extra content is needed in 802.3dj for Time Synchronization

- List Clause 90 and Annex 90A as optional clauses
  - in the rate introductory clauses
  - in the Physical Layer clauses associated with each PHY
- Are any specific instructions/examples needed for path data delay calculation through the new PHY functions?
  - Debatable!
  - Where to put them?

PTP/Timestamping Background

- Synchronize Time-of-Day across a network by exchanging messages and their recording their arrival/departure times.
- With the round-trip delay, the time difference between the TimeTransmitter and TimeReceiver can be calculated. The Time-of-Day (ToD) at the TimeReceiver can thus be synchronized to that of the TimeTransmitter.

Example with Ottawa (TimeTransmitter) and Montreal (TimeReceiver).
(Imagine carrier pigeons with a 1-hr transit time)

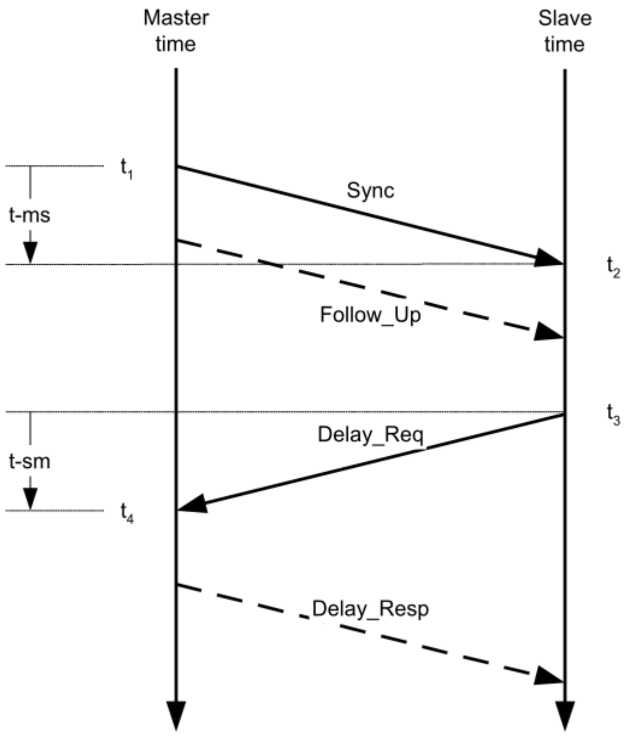
- Initially, Montreal's Time-of-Day is 5 minutes fast.

	Time in Ottawa	Time in Montreal	
Sync departure	**12:00**	12:05	T1 = 12:00
Sync arrival	13:00	**13:05**	T2 = 13:05
Delay_Req departure	13:55	**14:00**	T3 = 14:00
Delay_Req arrival	**14:55**	15:00	T4 = 14:55

Once T1,T2,T3,T4 are known, the necessary Time-of-Day adjustment in Montreal can be calculated:

Adjustment = [(T4-T3) - (T2-T1)]/2

= [55min - 65min]/2 = **-5 minutes**



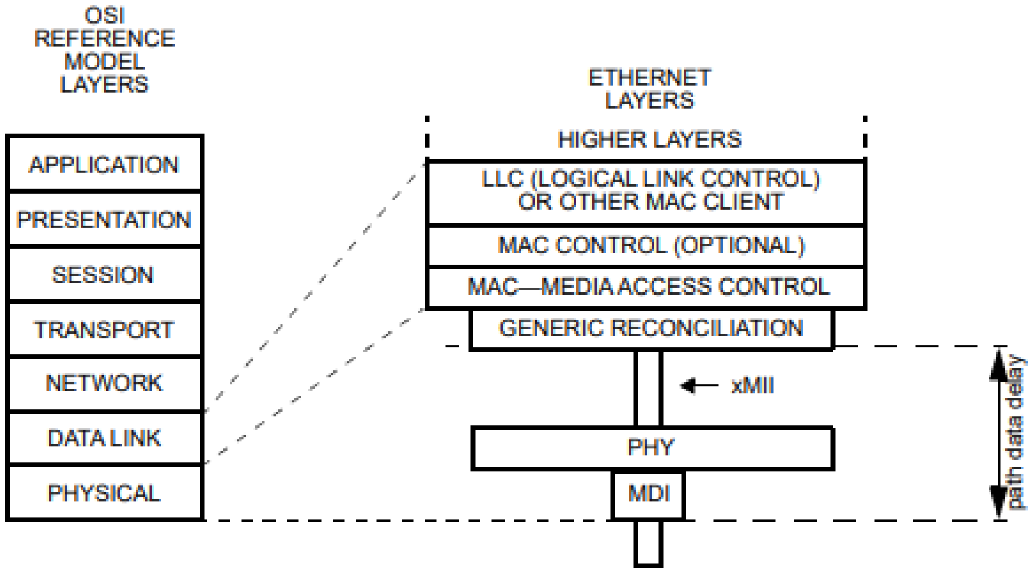
Important Notes:

- Latency of the medium does not matter, BUT
  - **Must be symmetric** (or have known asymmetry)
  - Unaccounted-for asymmetry leads to Constant Time Error (cTE) i.e. the error remaining after the ToD adjustment at the TimeReceiver.
- **The Timestamps are at the MDI** – i.e. at the PHY-medium interface.
  - The sync mechanism isolates the *perceived* difference in medium latency due ToD difference.
- The more precise the timestamps, the smaller the end-to-end cTE.
  - A consistent error on an arrival or departure time would look just like a link asymmetry.
- Requirements for time synchronization accuracy are defined in ITU-T Recommendation G.8273.2
  - "Class C" targets end-to-end cTE to within **+/- 10ns**.
  - "Class D" target is not yet official, but is projected to be **+/- 4ns**

Timestamping Model for Ethernet

- Determining the timestamp at the MDI is a challenge, especially with the complicated PHYs that exist in 802.3dj
- For many PHYs, the time when a start-of-frame crosses the MDI is a messy concept due to:
  - Distribution to lanes
  - Lane skew
  - Re-ordered symbols (e.g. convolutional interleaver)
  - Added/removed bits (e.g. FEC parity bits)

  => All these cause discontinuities in the MII-MDI latency

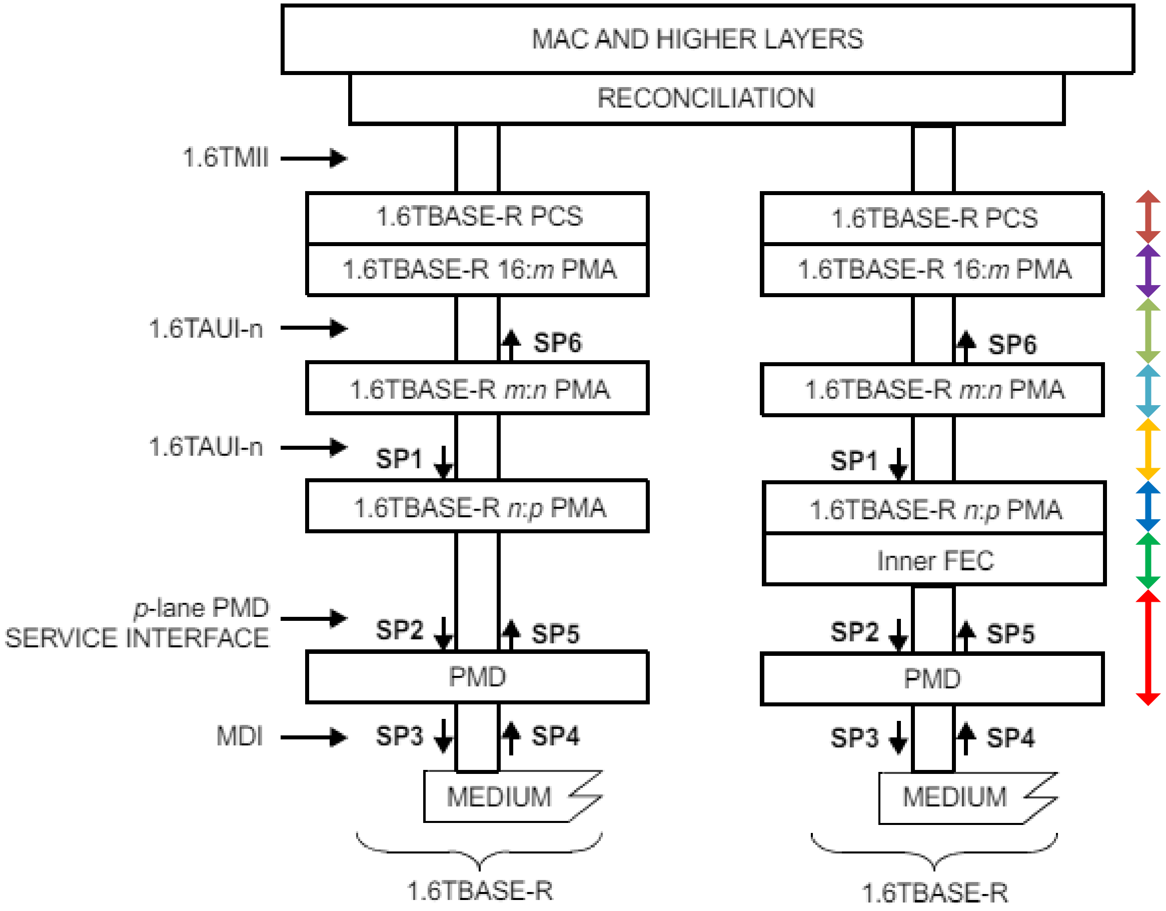


- To avoid these problems, Clause 90 presents a simpler method:
  - calculate timestamps at the MII, which is a smooth, continuous interface.
  - apply a **constant** offset (the **path data delay**) to adjust the time to that of the MDI.
- The path data delay takes into account BOTH:
  - Implementation delay (implementation-specific, may vary per startup)
  - Variable delay, if any (cyclical, deterministic, “intrinsic”, dictated by the standard).
- Clause 90, and Annex 90A give general instructions to calculate and allocate the variable, cyclical delays

Path data delay Example with a PHY stack

- The overall path data delay can be calculated piecemeal.
- The total path data delay would be the sum of:

1. PCS path data delay
2. 16:m PMA path data delay
3. AUI latency
4. m:n PMA path data delay
5. AUI latency
6. n:p PMA path data delay
7. Inner FEC path data delay
8. PMD path data delay



Any layer that has multiple functions can calculate its path data delay in the same manner:
Layer path data delay = func1 path data delay + func2 path data delay + …

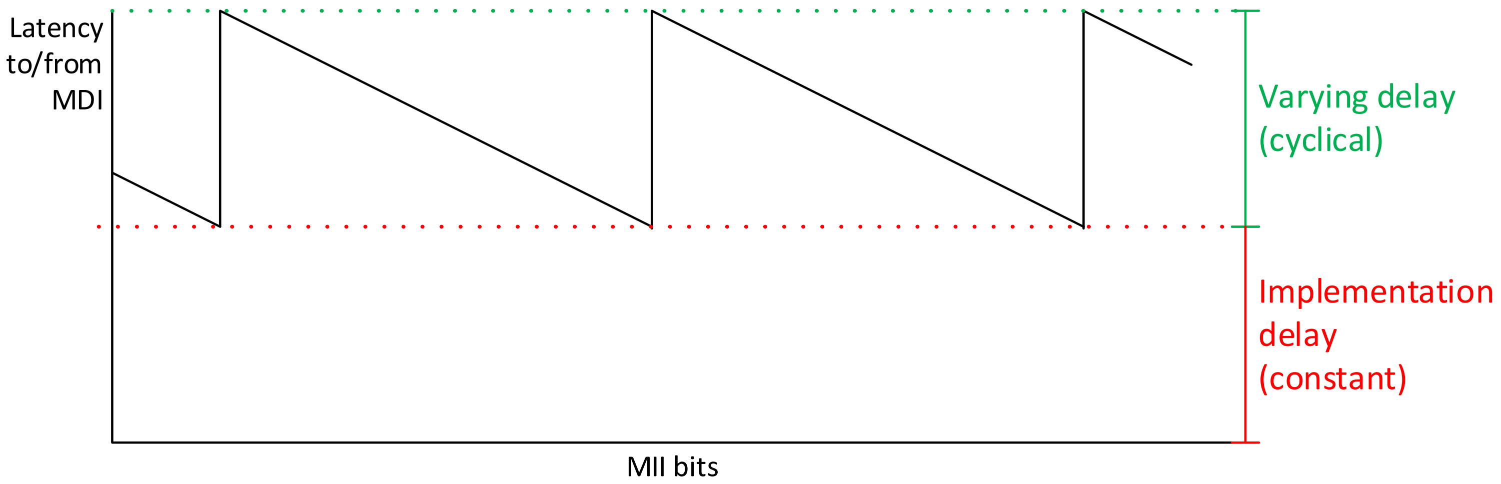
ASIDE: Implementation Delay

Driven by the implementation

- Pipeline delays
- FIFO delays
- wire delays
- SerDes delays
- etc

Constant value per startup,

but may vary between startups.



To determine the overall path data delay, an accurate measure of the implementation latency is necessary. The implementation's latency can either be:

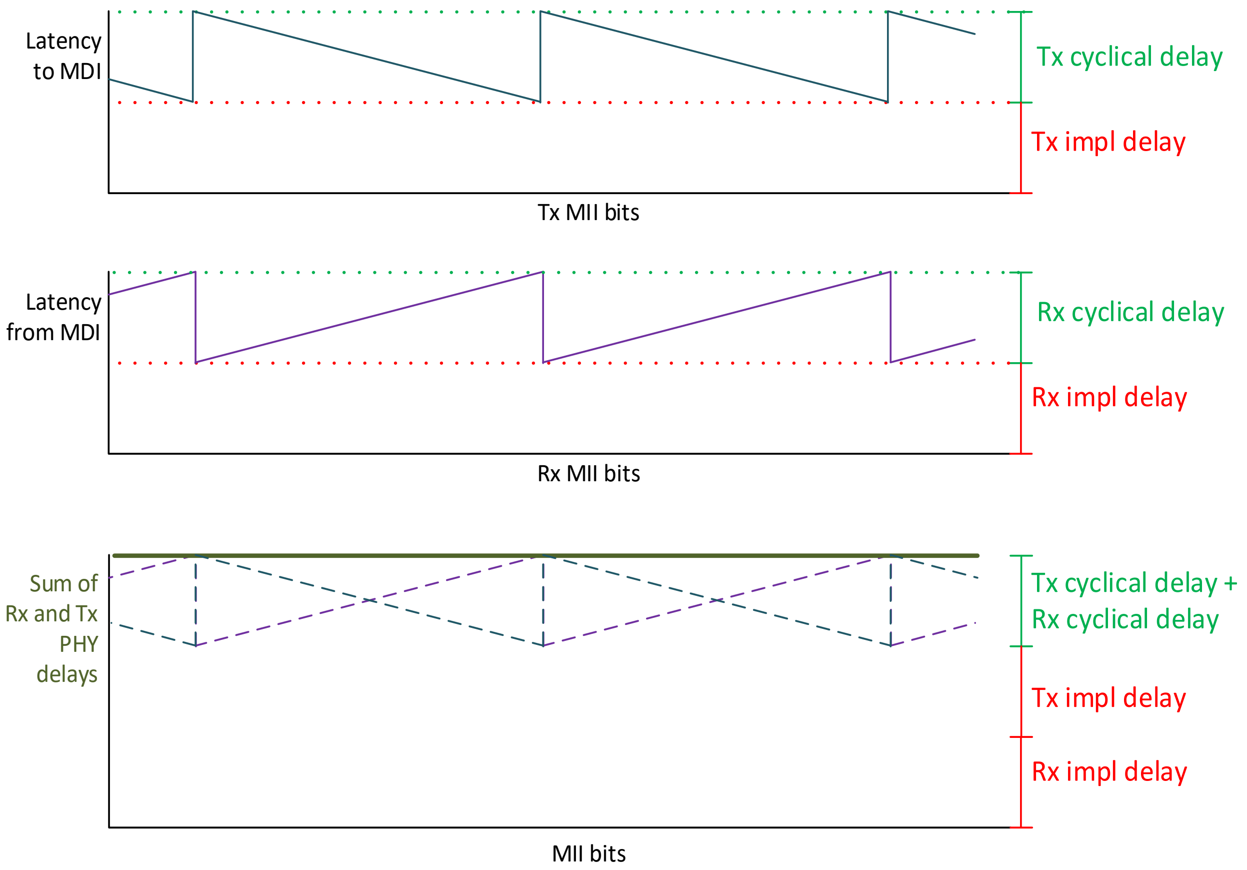
- Guaranteed by design – needs careful consideration from the start
  - FIFOs and clock-domain-crossings are tricky as they can introduce significant startup-to-startup variation.
- Measured – likely need built-in hardware to do so, as it is difficult to observe externally.

➔ This is an implementation concern, not driven by standards.

The path data delay MDIO registers consist of a pair of values (min & max) in order to express its accuracy.

Apart from setting a total latency ceiling for each PHY layer (for PAUSE), 802.3 does not concern itself with the implementation delay.

Mirrored Cyclical Delays through the PHY

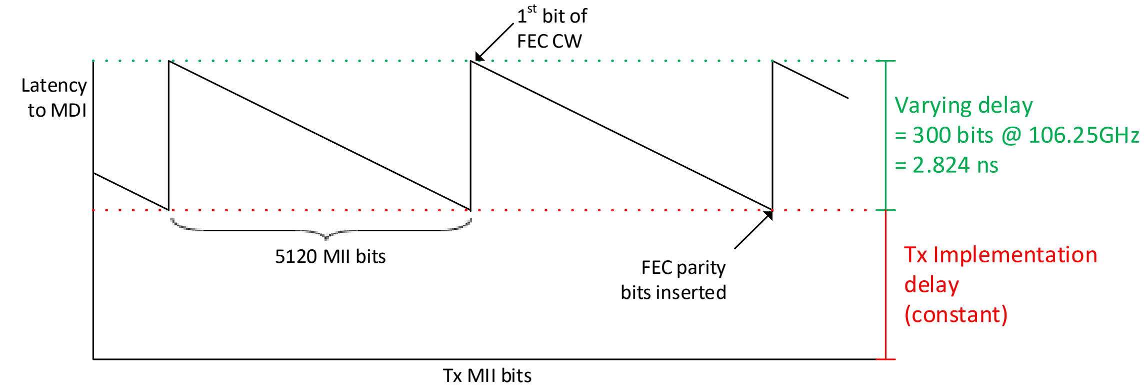


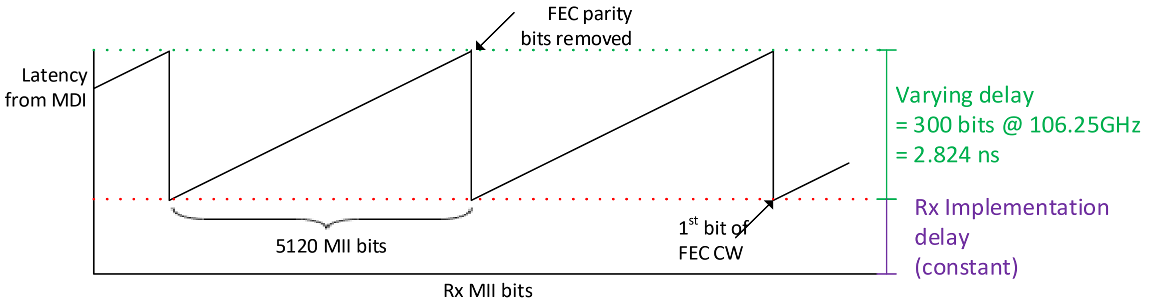
The variable delay through the TxPHY always mirrors the variable delay through the RxPHY.

- Tx PHY functions are always undone on the Rx side, so the same MII stream is recovered at the far-end.
  - Tx variable delay + Rx variable delay is always **constant**
- (Ignore the effects of AM and idle insertion/deletion for this presentation, they are dealt with through other means)

- How to allocate the constant sum of variable delays between the Tx and Rx path data delay values?
- If the two ends do not follow the same convention, then it results and too much or too little delay overall, causing greater cTE.
- MUST have a clear convention to allocate the “shared” latency to the path data delay values.

Practical Example : RS(544,514) for a 100GBASE-R PHY





Parity bits are inserted every 5120 MII bits

- At line rate, the 300 parity bits are 2.824 ns
- Can be seen as a discontinuity in the MII-MDI latency

Tx :

- MII-MDI latency pattern is a saw-tooth
- The beginning of the FEC CW has the largest MII-to-MDI latency

Rx :

- MDI-MII latency pattern is a reverse saw-tooth
- The beginning of the FEC CW has the smallest MII-to-MDI latency

Clause 90.7.1 provides explicit instructions as to how to calculate the path data delay:

- “it is recommended that the transmit and receive path data delays be reported as if the DDMP is at the start of the FEC codeword and/or at the start of the PCS lane distribution sequence”

Annex 90A.7 generalizes the rule to *any* PHY function that has mirrored Tx/Rx cyclical delays:

- “recommended to … allocate the maximum value of the intrinsic delay to the transmit PHY and the minimum value of the intrinsic delay to the receive PHY.”

- Tx path data delay = tx implementation delay + 2.824ns
- Rx path data delay = rx implementation delay + 0ns

Apply the path data delay principles to other PHY functions : Clause 177 Convolutional Interleaver

The convolutional interleaver (CI) consists of 3 delay-lines : Zero-delay, 4CW delay, and 8CW delay.
The delay variation through the CI is thus 8 RS-FEC CWs

- For a 200GBASE-R, 8CWs is 204.8ns

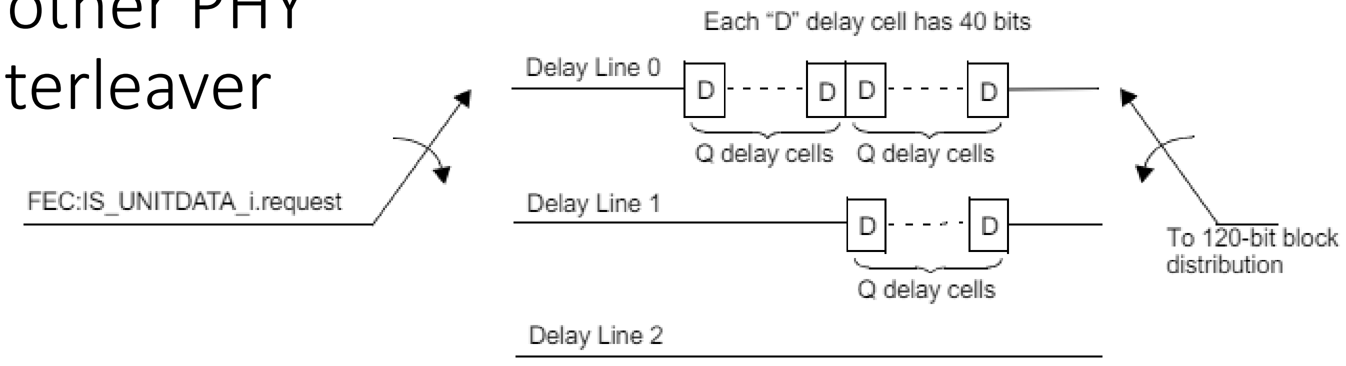
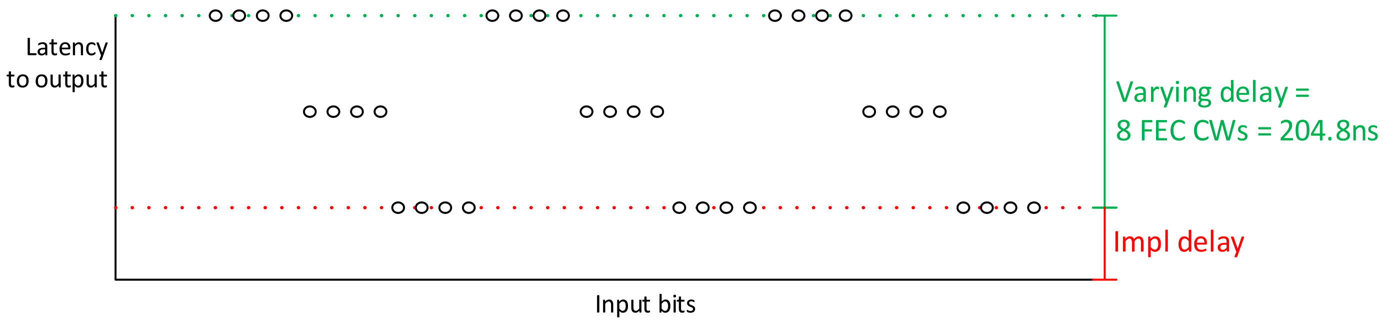

Each "D" delay cell has 40 bits
Delay Line 0
D
D
D
D
Q delay cells
Q delay cells
FEC:IS_UNITDATA_i.request
Delay Line 1
D
D
Q delay cells
Delay Line 2
To 120-bit block distribution

Figure 177–3—Convolutional interleaver

The path data delay for the CI function can thus be calculated using the directive in Annex 90A.7:

- Tx path data delay : allocate the maximum variable delay to the transmitter - use a bit that takes the 8CW branch through the CI
- Rx path data delay : allocate the minimum variable delay to the receiver – use a bit that takes the zero-delay branch through the convolutional de-interleaver.
  - Any bit that goes through the max-delay branch on Tx will necessarily go through the zero-delay branch on Rx.
- This corresponds to the conclusions in : https://www.ieee802.org/3/dj/public/24_05/he_3dj_01a_2405.pdf

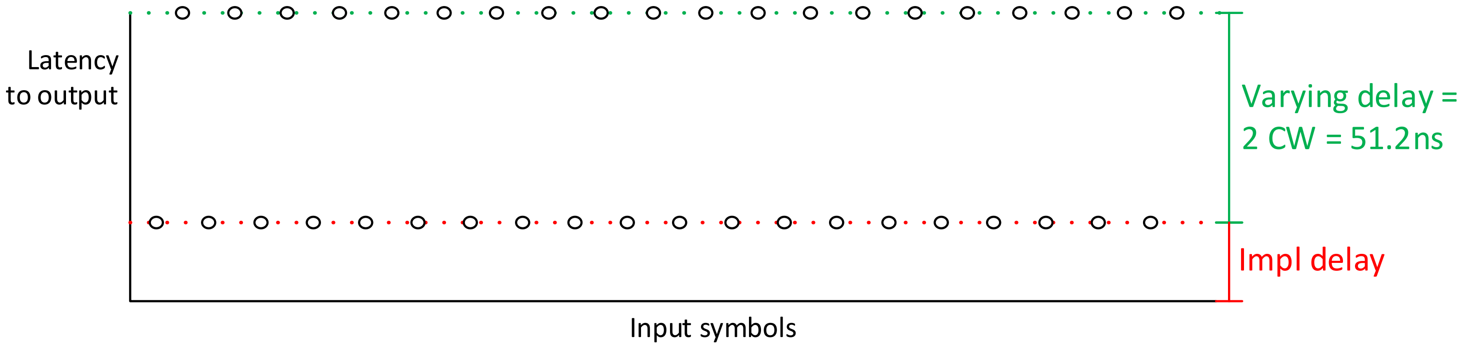


Apply the path data delay principles to other PHY functions : Clause 176 8:1 and 1:8 SM-PMA

In the 8:1 SM-PMA:

- Odd PCS lanes are delayed by
  - 1 10-bit symbol
  - 2 RS-FEC Codewords

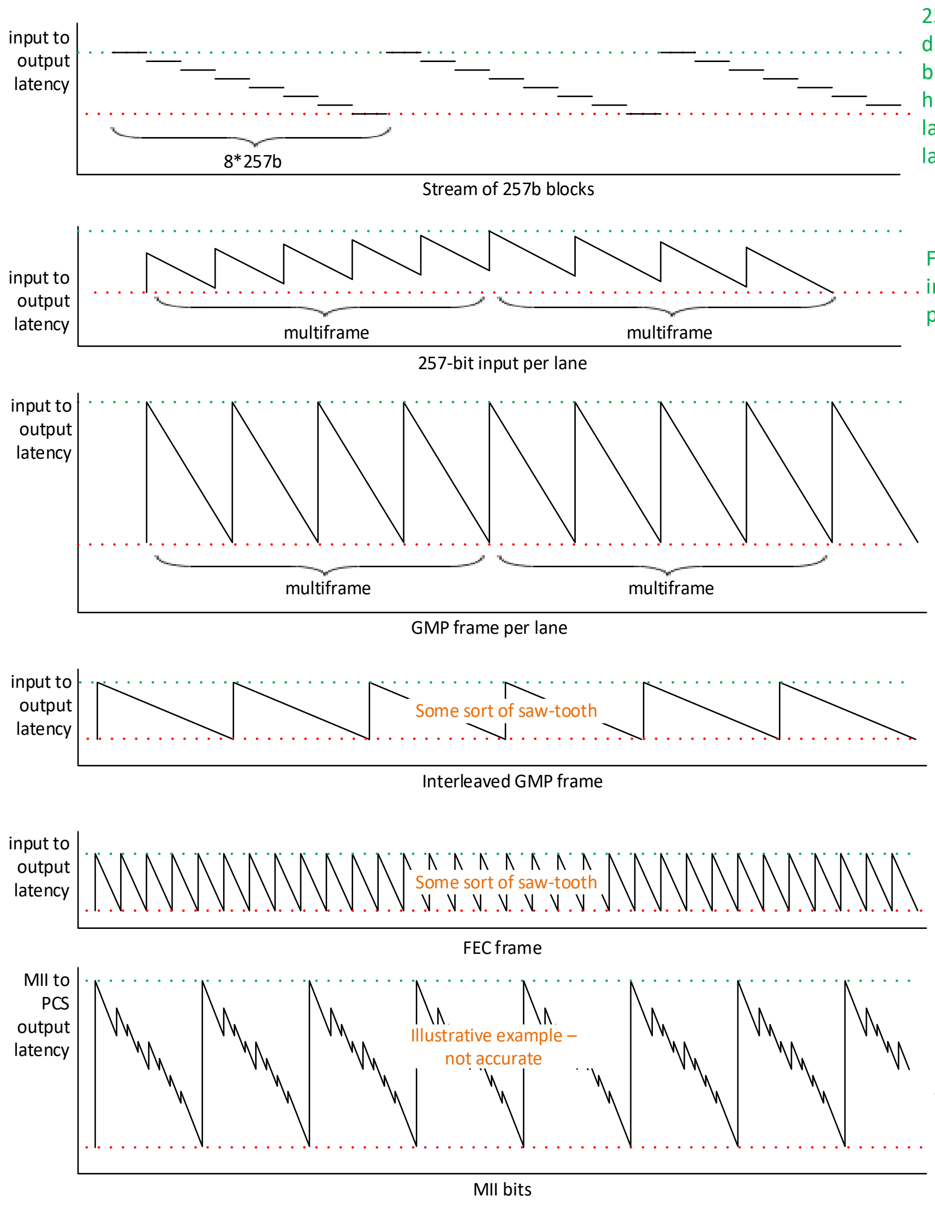
The delay of the symbols are thus scattered between two values



- In the 1:8 SM-PMA, the pattern is reversed, with the even lanes delayed by 2 CW.

- Overall :
  - Allocate all of the variable delay (51.2ns) to the transmitter’s (8:1 SM-PMA) path data delay
    - Corresponds to a symbol from an odd PCS lane
  - Allocate none of the variable delay to the receiver’s (1:8 SM-PMA) path data delay
    - Corresponds to a symbol from an odd PCS lane
  - Simple enough using the established rules for allocating the path data delay!

How to calculate the path data delay through the 800GBASE-ER1/ER1-20 PCS?



257b block distribution: block to the first lane has 7*257b more latency than to the last lane

Fixed stuff: 257bits inserted 3, 4 or 5 times per multiframe

GMP AM/pad/OH/pad: 1285 bits inserted for every 656635 (payload + FS bits)

CRC32 and 64-bit pad

FEC parity bits

Overall: Sum of all function delays.

- Can the general principle of "max latency for tx, min for rx" still work for a floating, asynchronous Ethernet payload within a GMP frame?
  - Not exactly cyclical in the same way as variable latency due to FEC parity bits, for instance.
  - But the principles (max on tx, min on rx) still apply.

- The MII-to-PCS-output latency plot may look like a “jagged saw-tooth” with different-sized steps. A series of superimposed saw-tooths with different heights & periods:
  - 257bit block distribution to 8 lanes
  - GMP frame overhead
  - Fixed stuff
    - position varies from one multiframe to another
    - Latency through delta-sigma
  - Trib frame Interleave
  - CRC32
  - Extra 64-bit pad to create the FEC frame
  - FEC parity bits
    - 17 parity bits added to every 111 FEC frame bits.
    - The FEC frame aligns with the PCS frame every 29 PCS frames

- More investigation required
  - What is the highest height of the jagged sawtooth?

- Implementation delays are tricky through the ER1 as well!

Possible Additions to the 802.3dj : Reference TimeSync Clauses

PHY Clauses should refer to Clause 90 and Annex 90A as optional.

- Add Clause 90 (and Annex 90A?) to the existing PHY layer tables.
- Seems very reasonable.
- Would be a maintenance headache if adding to older PHY clauses is desired for consistency.

Rate Introductory Clauses can refer to Clause 90

- Use the form of the function/sublayer references (see AN example from Cl 174 : Introduction to 1.6Tb/s networks)
- Seems reasonable
- Would be a maintenance headache if adding to older rate introductory clauses is desired for consistency.

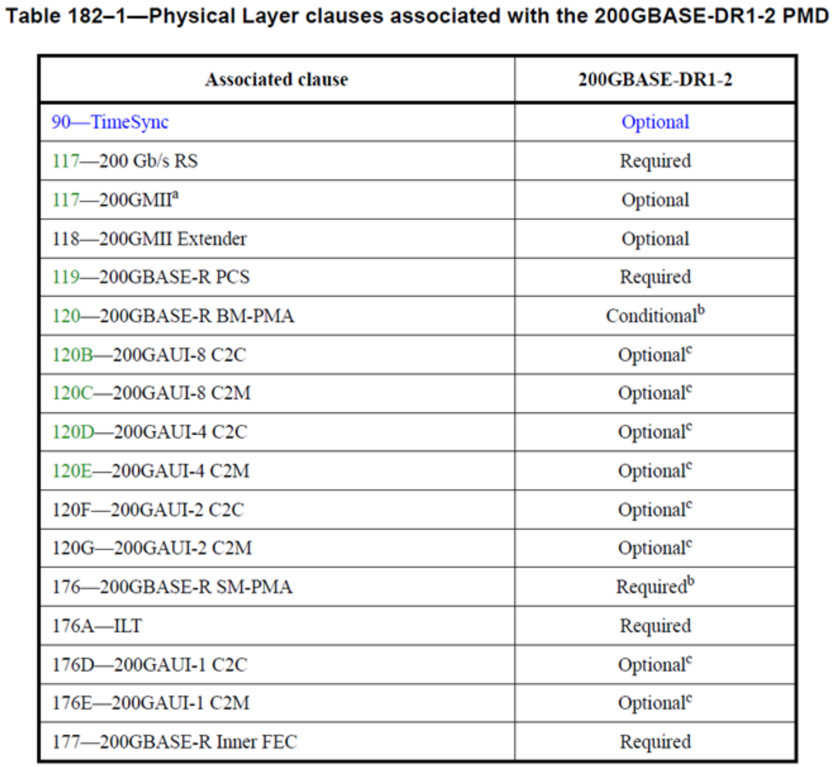
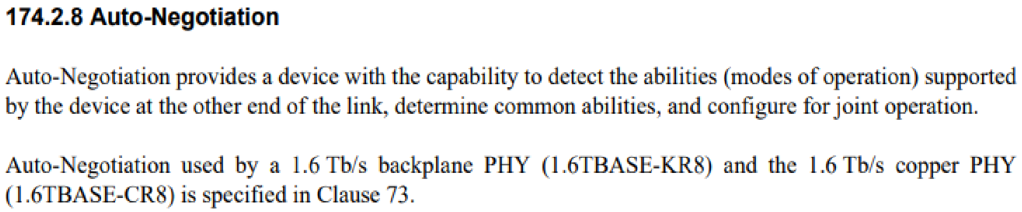


Table 182–1—Physical Layer clauses associated with the 200GBASE-DR1-2 PMD

Associated clause	200GBASE-DR1-2
90—TimeSync	Optional
117—200 Gb/s RS	Required
117—200GMII[a]	Optional
118—200GMII Extender	Optional
119—200GBASE-R PCS	Required
120—200GBASE-R BM-PMA	Conditional[b]
120B—200GAUI-8 C2C	Optional[c]
120C—200GAUI-8 C2M	Optional[c]
120D—200GAUI-4 C2C	Optional[c]
120E—200GAUI-4 C2M	Optional[c]
120F—200GAUI-2 C2C	Optional[c]
120G—200GAUI-2 C2M	Optional[c]
176—200GBASE-R SM-PMA	Required[b]
176A—ILT	Required
176D—200GAUI-1 C2C	Optional[c]
176E—200GAUI-1 C2M	Optional[c]
177—200GBASE-R Inner FEC	Required



174.2.8 Auto-Negotiation

Auto-Negotiation provides a device with the capability to detect the abilities (modes of operation) supported by the device at the other end of the link, determine common abilities, and configure for joint operation.

Auto-Negotiation used by a 1.6 Tb/s backplane PHY (1.6TBASE-KR8) and the 1.6 Tb/s copper PHY (1.6TBASE-CR8) is specified in Clause 73.

Possible Additions to the 802.3dj : Instructions for path data delay

Are explicit instructions for calculating the path data delay required for the new PHY functions?

- As the presentation shows, the existing directives provide the method to calculate the path data delay for any PHY function.
- However, there is no concise rule for many new PHY layers like there is for the RS-FEC.
  - Looks strange that some PHY functions have instructions on how to calculate the path data delay while others do not.
  - Reasonable to expect readers to figure it out with the path data delay rules in Clause 90 and Annex 90A?

If similar a similar rule to that of the RS-FEC can be established, then it would be beneficial:

- "Through the <PHY layer> it is recommended that the transmit and receive path data delays be reported as if the DDMP is at <specific bit>"

Where would such recommendations be placed?

- In Clause 90?
  - Means that every new project needs to amend Clause 90, or
  - Clause 90 amended in maintenance after every project.
- In the specific PHY Clauses?
  - Might be cleaner, but not consistent with what exists today for FEC and lane distribution.
  - Forces each new PHY project to consider Time Synchronization. Good precedent to set.

Conclusion

- New PHY/Rate clauses should reference Clause 90 (and possibly Annex 90A).
  - Whether to update older PHY/Rate Clauses in the same way is a maintenance matter.
- Tx/Rx path data delay reporting:
  - Presentation has shown that all the tools exist to calculate and report the path data delay values in the MDIO registers
    - Implementors may not be immediately aware of how to use the existing rules, however.
  - Explicit instructions would be helpful for non-trivial PHY functions
    - Where to place those instructions is a matter of debate:
      - In Clause 90, like the instruction for FEC codewords
      - In the relevant PHY sublayer clauses themselves

Thanks!

Appendix (From 802.3cx CFI): 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: constant time error
  - dTE_L: low-passed dynamic time error
    - MTIE: Maximum Time Interval Error
    - TDEV: Time Deviation
  - TE_L: constant time error + low-passed dynamic time error
  - TE: constant time error + unfiltered dynamic time error

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

Time Error Type	Class	Requirement (ns)
$\max\lvert\text{TE}\rvert$	A	100
	B	70
	C	30
	D	for further study
$\max\lvert\text{TE}_L\rvert$	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	$m < \tau \le 1000$ (for constant temp)
	A and B	MTIE = 40	$m < \tau \le 10000$ (for variable temp)
	C	MTIE = 10	$m < \tau \le 1000$ (for constant temp)
	D	MTIE = for further study	
	A and B	TDEV = 4	$m < \tau \le 1000$ (for constant temp)
	C	TDEV = 2	
	D	TDEV = for further study	