Necessity of a Latency Objective

Dayin XU
Rockwell Automation

Technical Facts

- We want **SINGLE** 100BASE-T1L PHY with low latency mode (e.g., FEC disabled) and long reach mode (e.g., FEC enabled)
- We **DON'T** want two 100BASE-T1L PHYs: one for long reach and one for low latency.

- In the presentation of 100BASE-T1L Reach and Connectors, we understand that
  - Most likely FEC is required for the 100BASE-T1L PHY to achieve the link segment objective of "supporting up to 5 inline connectors for up to at least 500m reach" for process automation use cases
  - 100BASE-T1L PHY FEC would possibly add at least 3-5us latency

- Disabling the FEC would allow 100BASE-T1L PHY operating in a low latency mode with minimal complexity to the PHY

- Providing the capability to enable/disable the FEC function to achieve low latency operation has been proven in available PHYs (e.g., 1000BASE-T1 PHY)
  - It should not be a challenge for the 100BASE-T1L PHY

Current Situation

- We have discussed the latency objective for several times
- In Nov. plenary meeting
  - Concern on the 1.5usec number was raised
  - Not all participants heard the presentation Latency Objective for > 10Mb/s SPE supporting the latency objective

- Consensus needs to be built on whether to have this objective or not

POTENTIAL ADDITIONAL FEATURE OBJECTIVES

Consensus:

To work:

- Latency:
  - Support a low latency mode of operation with ≤ 1.5usec latency for constrained link segment specifications (e.g., insertion loss or noise)

Why is the magic number of 1.5us

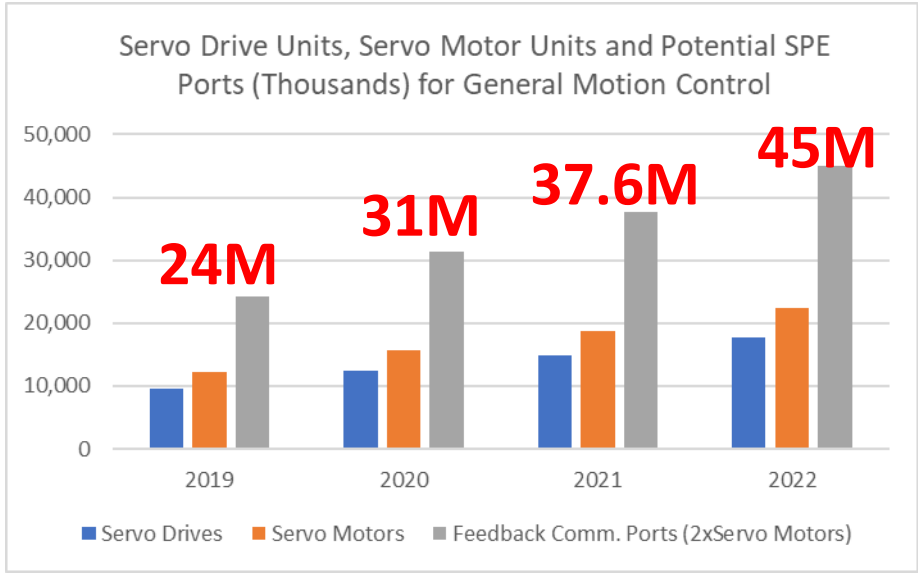
- Production 100BASE-T1 PHY achieves 900ns latency
  - It is feasible
- Servo Motor Control requires <1.64us
  - It meets requirement
- 1.5us is the trade-off between what is the feasible and what is required

Refer to Latency Objective for > 10Mb/s SPE for more details

Why we need the latency objective: Silicon Vendor Perspective

- W/o a latency objective, 100BASE-T1L PHY chip may not support a FEC disabling option in an **INTEROPERABLE** way
- As a result, this 100BASE-T1L PHY chip is useless for low latency applications (e.g., servo motor control, …)
- This sounds a very bad outcome if it does happen since it ignores the big market potentials
- **A latency objective is needed for a potential higher volume adoption of 100BASE-T1L PHY chips!**

- **31M** feedback comm. ports in 2020, 44% of them (**13.6M**) require low latency PHYs (<=1.5us)
- 20% annual growth

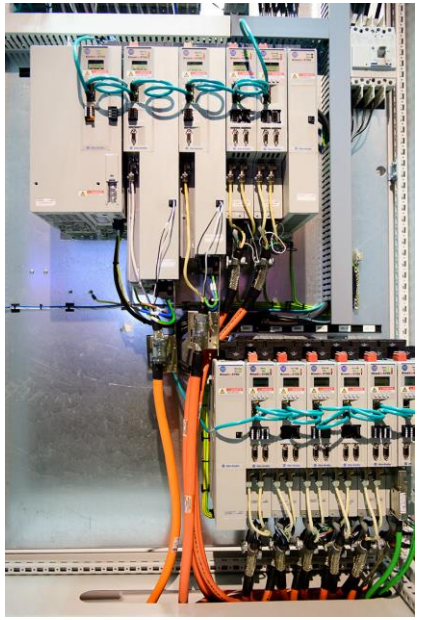


Data source: Omdia (former IHS), July 2021

Why we need the latency objective: Automation Vendor Perspective

- Automation vendors want to adopt 100BASE-T1L into products for low latency applications (e.g., servo motor control, linear switched topology, …)
- The latency requirement is technically achievable for their application scenarios, but unfortunately 100BASE-T1L PHY does not have it since the low latency operation model is not an objective
- As a result, automation vendors lose opportunities to upgrade the products with 100BASE-T1L technologies
- This sounds a very bad outcome too if it does happen

- **A latency objective is needed for a potential technology upgrade for low latency automation use cases (e.g., servo motor control, linear switched topology)!**



Conclusion

- We should have the following latency objective (preferred)
  - Support a low latency mode of operation with ≤ 1.5usec latency for constrained link segment specifications (e.g., insertion loss or noise)

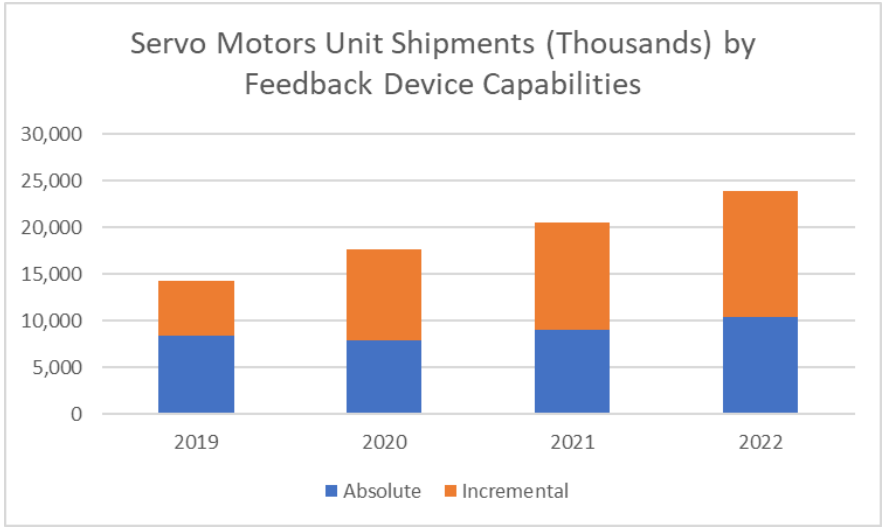
- If we can not build the consensus on the 1.5us number, we at least should have the latency object as
  - Support a low latency mode of operation for constrained link segment specifications (e.g., insertion loss or noise)
- This will assure that the low latency operation model will be supported in the 100BASE-T1L PHY, the real latency number depends on vendors' implementations.

Thank You

Servo Motors Unit Shipments by Feedback Device Capabilities

Absolute encoder always require low latency, it is about 44% of total

	2019	2020	2021	2022
Absolute encoder (thousands)	**8,444 (58.9%)**	**7869 (44%)**	**9002 (44%)**	**10459 (44%)**
Incremental encoder (thousands)	5880 (41.1%)	9859 (56%)	11585 (56%)	13404 (56%)



Data source: Omdia (former IHS), July 2021

(note here is all servo motors including General Motion Control servo motors)