Comparing 802.1AS structure options for P802.1ASds integration

and looking at 802.1AS structure for options overall

Alon Regev
Keysight Technologies

2025-09-16

Introduction

- This presentation captures one person’s observations about 802.1AS regarding the organization of options in 802.1AS
- For 802.1ASds, my reading of the group consensus is that we should not change the structure now, but may consider doing so in a separate project
- In this presentation I explore different options and attempt to compare proposed solutions as well to propose alternatives
- I am looking at multiple optional features in 802.1AS so that a chosen solution is universal rather than just addressing half-duplex Ethernet

802.1AS structure observations

- 802.1AS divided into two layers: media independent and media dependent
  - At the time 802.1AS-2011 was released, the MD layer was mostly the only "option" with layering mostly clean
    - 802.1AS-2011 had CSN defined in Annex E, mostly referring to portions of clauses 10 and 11
  - 802.1AS-2020 added several features that affected structure
    - Example: multiple domain support and CMLDS was added, changed some variables to be per-domain and others to be across domains; but not completely propagated to clauses 12, 13, and 16
  - 802.1ASdm added several features
    - Example: Drift Tracking was specified to only work with Clause 11 MD, but the feature (potentially in the future) may be needed for other MD types as we may have the same issue with temperature-driven frequency drift occurring on other MD types.
      - Drift Tracking involved changes to clauses 9 and 10 to support it (in addition to clause 11), but it was decided to only support on Clause 11
- We are moving towards having more options (i.e. external port configuration, optional announce messages, peer delay measurement, etc.)
- Over time, the majority of "optional" features were added to clause 11 and often not to other MD clauses, and often in a way that also affects common clauses

Different reuse requirements for different Implementations

- Different implementations cover different option spaces
- Example: Many implementations are derived from linuxptp
  - “draft” linuxptp variants exist supporting WiFi, driftTracking_TLV, HDE, etc., but not all in the same repo / branch
  - Assuming that all changes were accepted, the linuxptp codebase would cover many of the 802.1AS options
- Example: Aerospace implementations
  - Needs to be qualified for safety & airworthiness
  - Implementations tend to minimize complexity / conditionals to simplify proving correctness under all conditions
  - Typically, only implements the minimum subset of gPTP and options
    - This is likely a subset of clause 11 and a subset of clause 10
- From a code perspective, many implementations do not implement just “all of clause X”, but rather pick and choose from multiple subclauses

HDE: Current 802.1ASds

- 802.1ASds adds a new clause 19 which mostly points to clause 11
- Also changes clause 11 rather that duplicating state machines / variables with small changes
- While a couple of state machine have changes to support HDE, the biggest changes are
  - Functions that behave differently in the two modes
  - Many variables are per-domain in HDE and shared across domains in FDE
- The changes to functions and variables make this hard to resolve in a “clean” way
- CSN also has its own clause with pointers to clause 11, but does not modify clause 11 (rather it describes changes to behavior of clause 11 in clause 16) and does not completely address the multi-domain issues or the fact that referenced variables from clause 10 are not shared across domains

The role of CMLDS / multi-domain support

- I believe that the choices made during the integration of multi-domain support and CMLDS into 802.1AS-2020 cause a lot of the issues today
  - Multi-domain support assumes that some variables (those associated with link delay measurement or compensation) are only on domain 0 and are effectively shared with other domains
    - This is true for full-duplex peer-to-peer Ethernet, but not necessarily for other networks
  - CMLDS behavior is defined in BOTH clauses 10 & 11 even though it is effectively only applicable to full-duplex peer-to-peer Ethernet
  - A good example is table 10-1 which defines which clause 10 variables are shared across within a Link Ports for CMLDS (i.e. shared across domains sharing the same physical port)
- This could have been done in a cleaner way
  - One example: In clause 10, leave all variables as per PTP Instance or per PTP Port per PTP Instance and in clause 11 have the CMLDS and transport-specific delay peer-delay state machines or functions set the values across all PTP instances, leaving clause 10 as generic to all MDs.

HDE: options (1/2)

- Two options have already been presented
  - Merging clause 19 into clause 11: https://www.ieee802.org/1/files/public/docs2025/ds-specht-cl11-cl19-merge-proposal-0725-v01.pdf
    - This avoids having a clause that is mostly references to another clause, but adds to the amount of optional features in clause 11
  - Remove clause 11 changes and make all changes in clause 19: https://www.ieee802.org/1/files/private/asds-drafts/d1/ds-rodrigues-cl19-0825-v00.pdf
    - This avoids making changes to clause 11, but makes clause 19 harder to read as it effectively references clause 11 but makes exceptions to what is referenced

HDE: options (2/2)

- Other options
  - Completely replicate the relevant parts of clause 11 to clause 19 so that clause 19 has to make no references to clause 11
    - Makes clause 19 easier to comprehend, but makes maintenance much harder as changes have to be duplicated
  - Replicate the IEEE Std 1588 approach ("General optional features" in clause 16 of IEEE Std 1588)
    - This allows having many features, but they are not always fully structured and many are of the "do what clause X does except for this change" form, making it a bit obtuse to understand and sometimes having contradictions between different options
  - Restructure 802.1AS such that "features" are separated from the MD layer and different MD layers can use different "features", some of which are required and some are optional
    - This should be done in a structured way, defining the interfaces between the features and clients of the features.
    - Avoid use of mostly global variables in the features and instead pass the data to/from the feature through the interface, allowing the client to determine how the data flow to/from the feature is structured
    - Index variables by PTP Instance / PTP Port / Link Port
      - i.e. asCapable[this PTP Port] to refer to a specific instance
      - i.e., meanLinkDelay[all PTP Ports on Link Port X] to refer to all the PTP ports sharing the same physical port and CMLDS Instance and Link Port
    - May achieve a more structured / flexible approach, but is a bigger lift. May introduce errors. Effect on maintenance not clear.
  - Other approaches?

Next Steps

- I have started applying the "restructure" approach to clause 11 to provide a preview of how it would look
  - This work is unfortunately not ready to share at this interim (due to scope and time limitations)
    - I plan to have this ready before the November Plenary
- If the group decides to pursue any of these approaches, likely a new PAR (new project) or PAR modification to another project (i.e. AS-2020-Rev) would be needed