

# Several Details Related to Resolution of Comments 69, 70, and 88 on 802.1ASdm/D1.0

**Geoffrey M. Garner**  
**Analog Devices (Consultant)**

**[gmgarner@alum.mit.edu](mailto:gmgarner@alum.mit.edu)**

*IEEE 802.1 TSN TG*  
2023.05.15

# Introduction - 1

---

- ❑ The resolution of comments 69, 70, and 88 on 802.1ASdm/D1.0 will allow the GptpCapableTransmit and GptpCapableReceive state machines to be enabled and disabled (i.e., turned on and off)
- ❑ The resolution of these comments is:
  - ACCEPT IN PRINCIPLE. Group 69, 70, 88. Add a managed object that can turn the gPtpCapableTransmit and gPtpCapableReceive state machines (together) on and off. The default is that they are turned on. The configured values shall be persistent (over power cycles). Add a NOTE that indicates the implications of turning the state machines off, especially in a network that is not completely engineered. Make appropriate changes, consistent with the above, to the conformance and PICS and all other locations needed. "Turned off" means that the state machines are disabled (i.e., the operation is as though the state machines were not present); this means that the pdelay exchange can occur immediately, i.e., it does not need to wait until the gPTP instances have exchanged the gPtpCapable TLVs; include this description in the NOTE.

# Introduction - 2

---

- However, several additional details must be decided on
  - Should the enable/disable be per PTP port or per PTP Instance?
  - How should the global variable neighborGptpCapable (see 10.2.5.16 of 802.1AS-2020) be set?

# Enable/Disable of State Machines - 1

---

- ❑ The GptpCapableTransmit and GptpCapableReceive state machines are per port
- ❑ This is because it must be determined for each PTP Port whether the PTP Port at the other end of the attached link is capable of invoking the 802.1AS-2020 protocol
- ❑ It has already been decided that the GptpCapableTransmit and GptpCapableReceive state machines will be enabled or disabled together (i.e., either both are enabled or both are disabled)
- ❑ However, the enable/disable could be done separately on each PTP Port
  - E.g., they could be enabled on some PTP Ports and disabled on other PTP ports
  - In this case, a separate, per port, managed object is needed (it would likely be placed in the portDS)

# Enable/Disable of State Machines - 2

---

- Alternatively, the enable/disable could be done for the entire PTP Instance, i.e., the state machines are enabled on every PTP port or disabled on every PTP Port
  - In this case, a single managed object would be needed; it would likely be placed in the defaultDS
- In both cases, the managed object would be configurable, i.e., RW, and the default setting would be ENABLE

# Global Variable neighborGptpCapable - 1

---

- ❑ The per PTP Port global variable neighborGptpCapable (see 10.2.5.16 of 802.1AS-2020) indicates whether or not the neighbor PTP Port is capable of invoking gPTP:
  - 10.2.5.16 neighborGptpCapable: A Boolean, set by the GptpCapableReceive state machine (see 10.4.2), that indicates that the neighbor of this PTP Port (i.e., the PTP Port at the other end of the link attached to this PTP Port) is capable of invoking gPTP.
- ❑ neighborGptpCapable is used in 11.2.2, 12.4, and 13.4 in the setting of the global variable (and managed object) asCapable (which also is per PTP Port)
- ❑ At present, neighborGptpCapable is set by the GptpCapableReceive state machine
- ❑ If the GptpCapableTransmit and GptpCapableReceive state machines are disabled, neighborGptpCapable must be set via some other means

# Global Variable neighborGptpCapable - 2

---

- ❑ One possibility for setting neighborGptpCapable is to simply set it to TRUE if the GptpCapableTransmit and GptpCapableReceive state machines are disabled
- ❑ In this case, it would be the responsibility of whoever configures the network to ensure that all PTP Ports for which the GptpCapableTransmit and GptpCapableReceive state machines are disabled are, in fact, capable of invoking gPTP
  - If the enable/disable is per PTP Instance, this means that all PTP Ports and PTP Instances must be capable of invoking gPTP if the state machines are disabled

# Global Variable neighborGptpCapable - 3

---

- ❑ A second possibility for setting neighborGptpCapable is to allow it to be configured via a managed object
- ❑ This would be somewhat more complex than the first possibility
  - Would the configuration be per PTP Port or per PTP Instance (likely this would follow the decision on whether enable/disable would be per PTP Port or per PTP Instance)?
  - Would configuration of neighborGptpCapable be limited to the case where the state machines are disabled, or would it be allowed in all cases (likely the former; it seems that allowing it in the latter case could lead to unintended behavior)?



# Summary

---

## ❑ Decisions are needed on two items

- Should the enable/disable be per PTP port or per PTP Instance?
- How should the global variable `neighborGptpCapable` (see 10.2.5.16 of 802.1AS-2020) be set?

## ❑ For the second item, should `neighborGptpCapable` be set to TRUE automatically if the state machines are disabled, or should it be configurable

- If configurable, should configuration be per PTP Port or per PTP Instance?
- If configurable, should configuration be limited to when the state machines are disabled?

---

Thank you