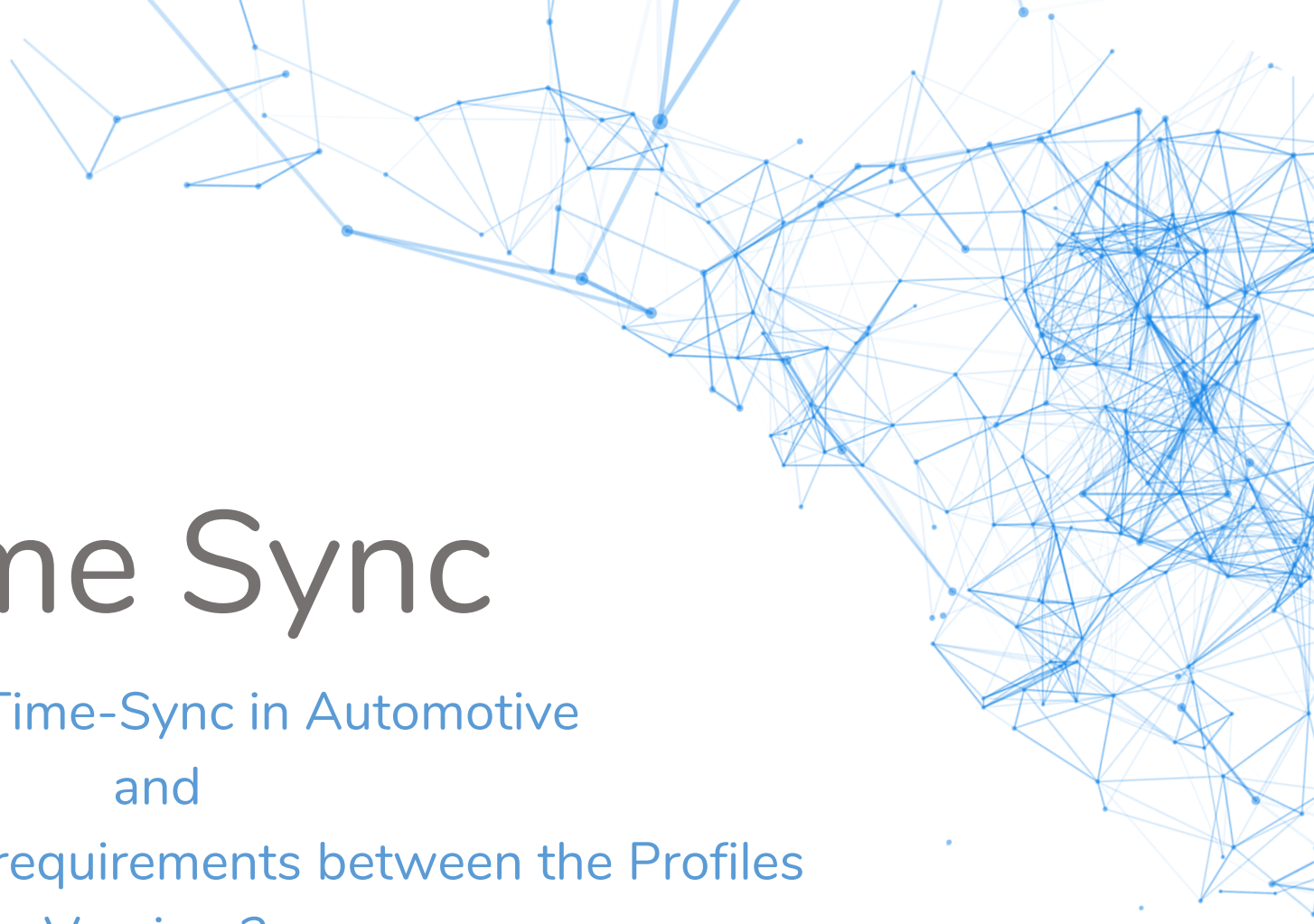

[May 9-13, 2022]
ETHERNOVIA
IEEE 802.1 Interim Meeting
IEEE Contribution



Time Sync

History of Time-Sync in Automotive
and
Aligning Time-Sync requirements between the Profiles
Version 2

History of Time-Sync in Automotive

From CAN and Autosar limitations and From Relays to Bridges

Delay Measurement Counter Service

- Data when passing from one application to another experiences latency
  - Software stacks in Talker and Listener
  - Serialisation and Arbitration
  - Relays
  - Gateways
  - …
- Initial goal was to give a “counter” which would be able to approximate time intervals between 1ms and 1s (in TAI terms)
- between all communication busses (CAN, FlexRay, Ethernet)!

Initial DMCS Challenges

- CAN (classic, 500kbit/s, 8Byte payload)
  - No hardware time-stamping
  - Autosar CP “TX-confirmation” (buffer release) was used as time-stamp event
  - No time-sync software stack available in Autosar CP
  - 8Byte payload can not carry the 10Byte PTP time information field
  - Accuracy (TAI) vs. Resolution (number of bits transmitted)
- Ethernet (100Mbit/s, automotive 2-wire)
  - No time-sync software stack available in Autosar CP (neither bridge nor end-station)
  - KISS! (keep it simple stupid) - limit interrupt load!
  - No good reason to be (much) better than CAN
  - No SW supporting time-sync in Bridges - Relays only (cost, legacy, suppliers, …)
  - Data Loggers interfere with Time-Sync
- Allow for (limited) “jumps” within multiple 100’s µsecs (no rate-correction!)

How to evolve to AVB?

- From the start is was clear, that long term “real PTP” might be needed to support e.g. audio applications
  - make sure both system could potentially be operated on the same link (resulted in VLAN & addressing differences)
- AVB required closer to TAI times (one might hear differences in syntonisation)
- Audio applications need ~1µs accuracy (not feasible with CAN)
- Still only Relays available (no software to support 2-step Bridges)
  - “Franken Two-Step” concept with one-step capable hardware
  - SW stacks had to add data from Sync and FollowUp
  - pDelay not feasible, do rate-correction from Sync

Start-Up Time and Load is key

- Any smoothing creates a delay during start-up
- “Jumps” must always be limited to within the resolution to avoid error entries (DTC)
- Time “running backwards” (due to backward “jumps”) must be limited for other application
- Following the central time source (GM) quickly is more important than “smoothness”
- Highest message/application load during start-up, whatever can be supported during start-up can “easily” run in steady state (removal of signalling messages)

Time in the context of Security

- Can pDelay be used to detect “Person in the Middle” attacks (IEEE Std 802.1X, EAPOL)? Requires independent operation to prevent information leakage, but high accuracy and authentication (need to trust the other clock)
  - Repair and assembly line restrictions apply
- Can “Time” be used as a (predictable) Nonce? Even more sensitive to backwards jumps!
  - NVM write cycle restrictions apply
- TAI is not always available (underground parking) - can certificates be validated after long parking periods?
  - Power consumption while parked prevents good time keeping
- GPS signal can easily be spoofed
- NTP may not be accurate to run other (no security) application off
  - Traffic light phase measurement

Safety in the context of Time-Sync

- “Time” is a physical concept, the actual passage of time can not be protected using checksums!
- PTP is a one way (top-down) protocol, there is no feedback to the outside if the receiver has properly processed the received information
- Comparing time between different nodes requires physical reference to actual “events”
- Observing physical events “through” a Bridge is almost impossible
- Adopt testing procedures to be run during operations
  - Safe Car Time - limit the number of components with safety load
  - Reverse-Sync - test very link by itself

Redundancy Discussion

What might be missing in 802.1ASdm for the automotive use-case?

- The secondary GM can currently not verify the GM's time (is what it is syncing to and forwarding to the other domain really what the GM is sending?)
- A node receiving both gPTP domains can currently not compare them
- Due to the CMLDS there is a single point of failure for all domains in every device
- Would redundancy in the Bridge reduce changes to a node (two time domains on the last link protect from few failure cases)?

Aligning Time-Sync requirements between the Profiles

Different Priorities in Profiles

- Accuracy compared with TAI (synchronization, syntonization)
- Stability vs. TAI or follow central source
- Availability of TAI references
- Start-Up time (from “cold and dark”) with and without connectivity
- Recovery from different fault conditions
- Safety and Certification
- Accessibility of components and Links
- System Test vs. Component Test
- Security of Time
- Using Time for Security
- Redundancy and Hold-Over scenarios
- Plug and Play (Plug and Produce)

Reducing Base to bare Minimums?

- Many options in the Base Standard
- Few options in the Profiles (could still be sections of 802.1AS document!)
- Base Standard:
  - Define Message format (already in IEEE 1588?)
  - Define Reference-Planes for Time-Stamping
  - Define default TLVs (extensible)
  - Modular independent function blocks (Sync/FollowUp, one-way pDelay, ..)
  - NEW: Define PHY-MAC info transfer (tiggered but not fully covered by IEEE802.3cx) - e.g. RX-timestamp TLV
  - less “automation”
- Profiles:
  - Details on Redundancy
  - Details on Accuracy and Traceability (TAI)
  - per-Port dependent operation
  - enable “automation”, but not mandatory

BMCA: Implement vs. “use”

IEEE Std 802.1AS-2020

f) Support the following best master clock algorithm (BMCA) requirements:

1) Implement the BMCA (10.3.2, 10.3.3, 10.3.4, 10.3.5, 10.3.6, 10.3.8, and 10.3.10).
2) For domain 0, implement specifications for externalPortConfigurationEnabled value of FALSE (10.3.1).
3) Implement the PortAnnounceReceive state machine (10.3.11).
4) Implement the PortAnnounceInformation state machine (10.3.12).
5) Implement the PortStateSelection state machine (10.3.13).
6) Have the BMCA as the default mode of operation, with externalPortConfiguration FALSE, on domain 0.
7) Implement at least one of the possibilities for externalPortConfigurationEnabled (i.e., FALSE, meaning the BMCA is used, and TRUE, meaning external port configuration is used) on domains other than domain 0.

Announce (10.3.12)

IEEE Std 802.1AS-2020

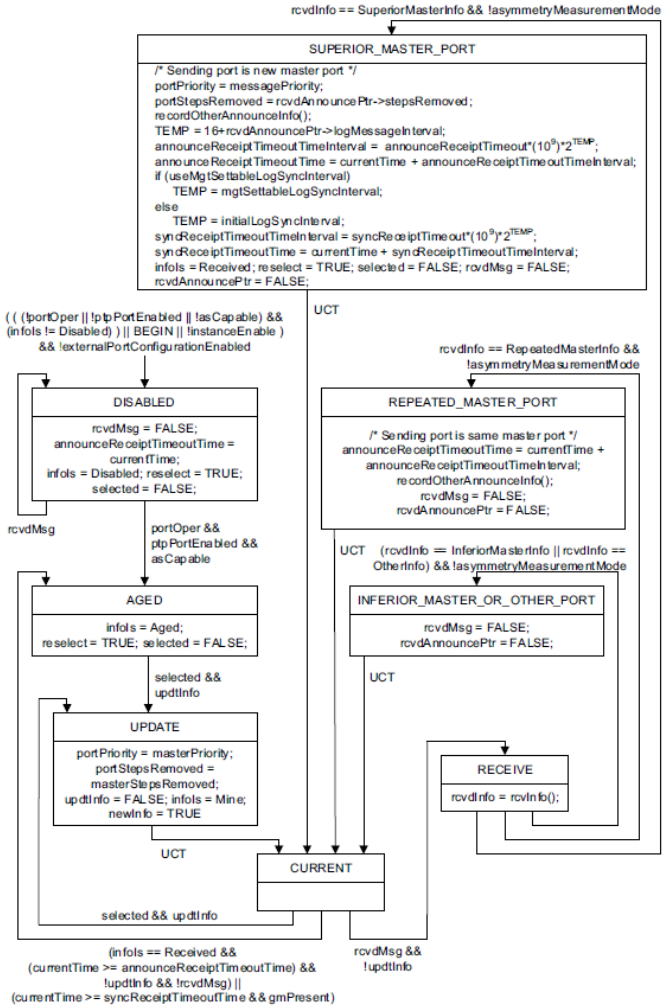


Figure 10-14—PortAnnounceInformation state machine

f) Support the following best master clock algorithm (BMCA) requirements:

1) Implement the BMCA (10.3.2, 10.3.3, 10.3.4, 10.3.5, 10.3.6, 10.3.8, and 10.3.10).
2) For domain 0, implement specifications for externalPortConfigurationEnabled value of FALSE (10.3.1).
3) Implement the PortAnnounceReceive state machine (10.3.11).
4) Implement the PortAnnounceInformation state machine (10.3.12).
5) Implement the PortStateSelection state machine (10.3.13).
6) Have the BMCA as the default mode of operation, with externalPortConfiguration FALSE, on domain 0.
7) Implement at least one of the possibilities for externalPortConfigurationEnabled (i.e., FALSE, meaning the BMCA is used, and TRUE, meaning external port configuration is used) on domains other than domain 0.

j) Support the performance requirements in B.1 and B.2.4.

IEEE Std 802.1AS-2020

- B.1.1 Frequency accuracy

  TAI frequency (see ISO 80000-3:2006 and Annex C) shall be within ± 100 ppm
- B.1.2 Time measurement granularity
- B.1.3 Noise generation
  - B.1.3.1 Jitter generationration
  - B.1.3.2 Wander generation
- B.2.4 Measurement of rate ratio

  measure rate ratio shall not exceed ±0.1 ppm

AS capable

IEEE Std 802.1AS-2020

12.4 Determination of asCapable

The per-PTP Port, per-domain instance of the global variable asCapable (see 10.2.5.1) is set to TRUE if the following conditions hold (see 12.5.1 and 12.5.2):

a) The value of tmFtmSupport is not zero.
b) neighborGptpCapable is TRUE.
c) At least one of the following conditions hold:
   1) Bit 0 of tmFtmSupport is TRUE.
   2) Bit 1 of tmFtmSupport is TRUE and, if the PTP Port is a master port, it can support (i.e., grant) the parameters requested by the slave with either FTMs per burst equal to 3 or FTMs per burst equal to 2.
   3) Bit 1 of tmFtmSupport is TRUE and, if the PTP Port is a slave port, the master port at the other end of the link can support (i.e., grant) the parameters requested by the slave with either FTMs per burst equal to 3 or FTMs per burst equal to 2.

If the value of domainNumber is zero (which is required for support of the 2011 edition of this standard) and bit 0 of tmFtmSupport is TRUE, asCapable can be set to TRUE. In all other instances, asCapable shall be set to FALSE.

One-Way pDelay

IEEE Std 802.1AS-2020

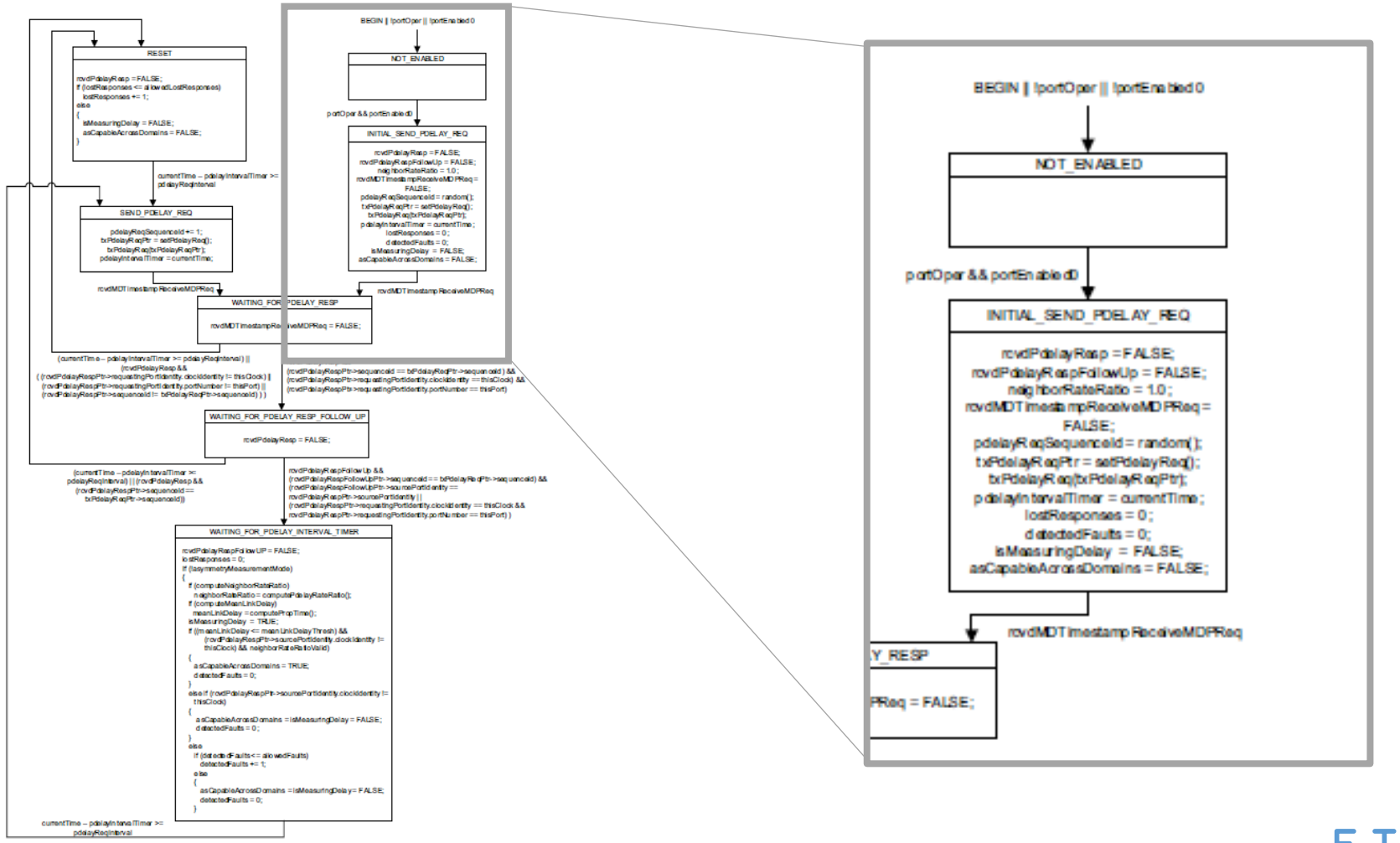


Figure 11-9—MDPdelayReq state machine

ETHERNOVIA
VIRTUALIZING VEHICLE COMMUNICATION

Timescale Domain 0

IEEE Std 802.1AS-2020

8.2 Timescale

8.2.1 Introduction

The timescale for a gPTP domain is established by the Grandmaster Clock. There are two types of timescales supported by gPTP:

— The timescale PTP: The epoch is the PTP epoch (see 8.2.2), and the timescale is continuous. The unit of measure of time is the second defined by International Atomic Time (TAI) (for the definition of TAI, see Service de la Rotation Terrestre [B28], with further amplification in IAU [B27]; and for more information on TAI, see Jekeli [B22], international system of units (SI) brochure [B14], and Petit and Luzum [B26]). The timescale of domain 0 shall be PTP. See IEEE Std 1588-2019 for more details.

Continued Sync TX blocked - syncLocked

IEEE Std 802.1AS-2020

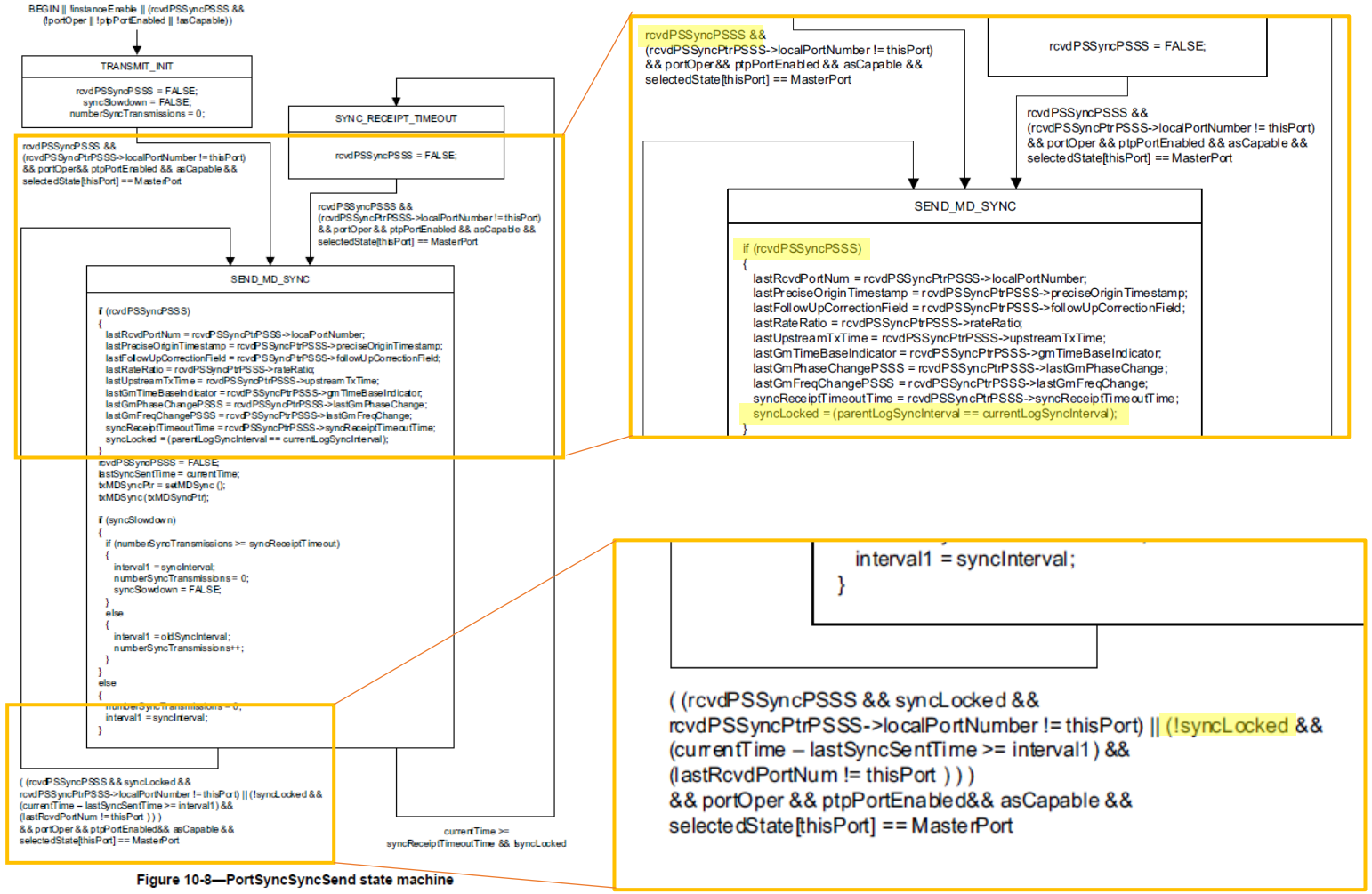


Figure 10-8—PortSyncSyncSend state machine

If all stations use the same LogSyncInterval, then syncLocked will always TRUE.

If syncLocked == TRUE, there will be no continued transmission of Sync-Messages.

ETHERNOVIA
VIRTUALIZING VEHICLE COMMUNICATION

Can a Bridge become a GM?

Thanks to Rodney Cummings for the pointer!

IEEE Std 802.1AS-2020

10.1.3 Grandmaster-capable PTP Instance

A PTP Instance may be grandmaster-capable. An implementation may provide the ability to configure a PTP Instance as grandmaster-capable via a management interface.

10.3.6.2 PTP Port state assignments when external port configuration is used

If external port configuration is used, one of the states MasterPort, SlavePort, PassivePort, or DisabledPort is assigned to each PTP Port by an external entity, as described in this subclause.

The DisabledPort state is assigned if portOper is FALSE (see 10.2.5.12), ptpPortEnabled is FALSE (see 10.2.5.13), or asCapable is FALSE (see 10.2.5.1).

The member externalPortConfigurationPortDS.desiredState (see 14.12.2) is used by an external entity to set the state of the respective PTP Port to MasterPort, SlavePort, or PassivePort. When this member is set, its value is copied to the per PTP Port local variable portStateInd (see 10.3.15.1.5). If portOper, ptpPortEnabled, and asCapable are all TRUE for this PTP Port, the PTP Port state is set equal to the value of externalPortConfigurationPortDS.desiredState by copying the value of this member to the element of the selectedState array (see 10.2.4.20) for this PTP Port.

Max Turner
Utrechtseweg 75
NL-3702AA Zeist
The Netherlands
+49 177 863 7804
max.turner@ethernovia.com


THANK YOU
ETHERNOVIA
VIRTUALIZING VEHICLE COMMUNICATION