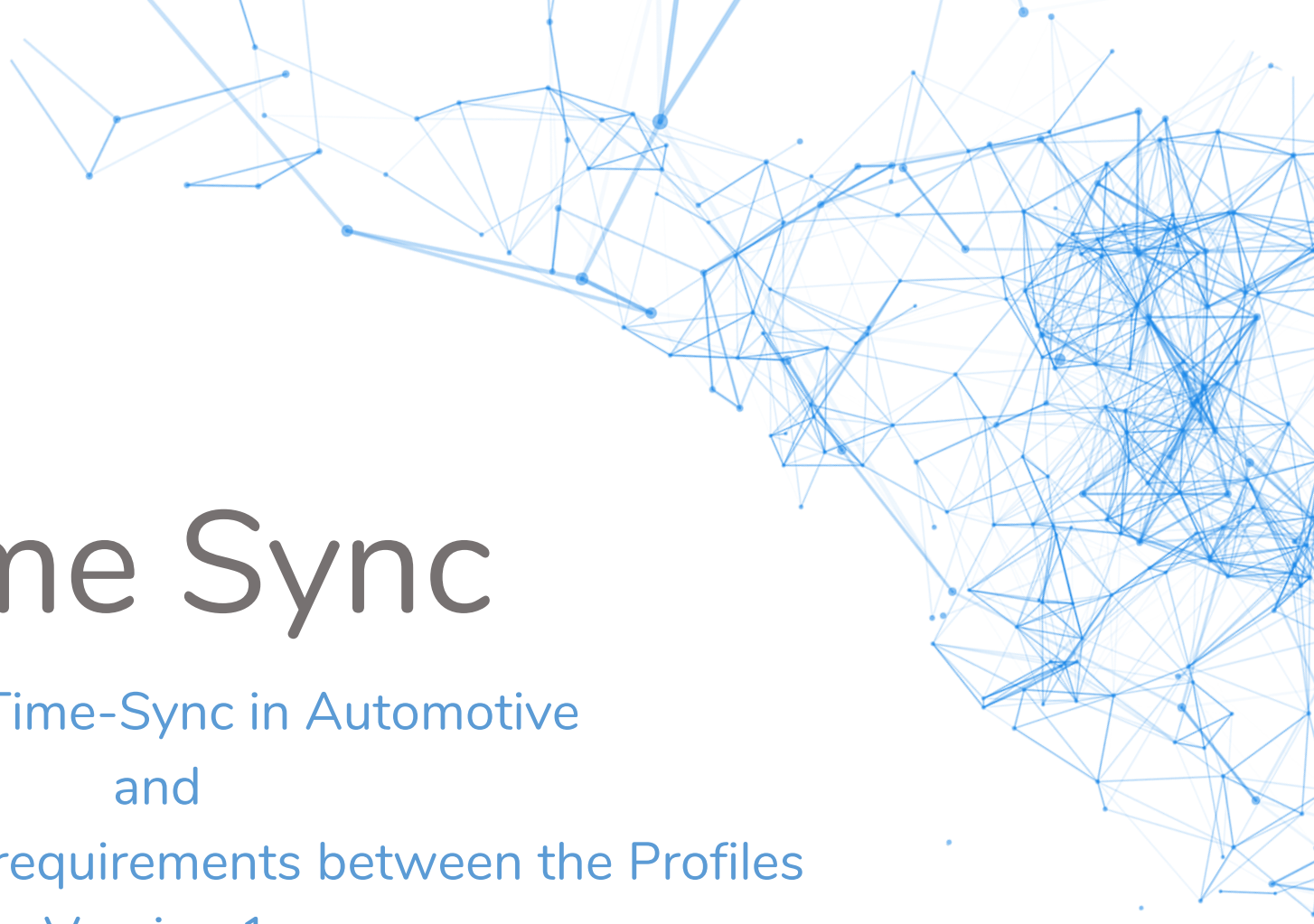
[May 9-13, 2022]

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 1

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 the resolution

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 avoided
- 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 lime 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

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

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


THANK YOU
ETHERNOVIA
VIRTUALIZING VEHICLE COMMUNICATION