Ethernet-as-a-Service for Software Defined Vehicles:
Design objectives and orientations for an Ethernet-based network stack

Pierre LACLAU – PhD Student
with Xiaoting LI and Trista LIN
2022 IEEE Standards Association
# Agenda

## Use Cases
- E/E Evolution
- Ethernet role

## Ecosystem
- E/E Evolution
- Ethernet role

## Objectives
- Use Cases
- E/E Evolution
- Ethernet role

## Technologies
- Timing
- Routing
- Service-oriented
- Management

## Ethernet-aaS
- XaaS Inspiration
- EaaS Concept
- Map of Ethernet

## Implementation
- Proposition

## Discussion
- Future guidelines
- Open questions

---

**Ethernet-as-a-Service**

for Software Defined Vehicles
Automotive Evolution – Use Cases
A profound digital transformation

Two major digital transformations

Sustainability
Shared & Robo vehicles, Micromobility, Smart Grid, Automated Driving, Cooperative Services

User Experience
Connectivity, Smartphone on Wheels, Personalization, Subscriptions, Regular Updates


Images: Ralf Marquard from LHP Europe, Eliane Fiolet from Ubergizmo
**Architectures Evolution**

*Software is eating the car*

**Domain-oriented**

*2000-2020*

- GW - Gateway
- ZCU - Zonal Control Unit
- HPC - High Performance Computing
- CCU - Connectivity Control Unit

**Zonal-oriented**

*2020 and beyond*

- GW - Gateway
- ZCU - Zonal Control Unit
- HPC - High Performance Computing

**.opengraph:**

- Not scalable
- Interface hell
- Too fast?

**Service-oriented**

- Standard-based
- Modular
- Interchangeable
- Dynamic Lifecycle

↔ **Service-oriented**: The right compromise for automotive

---

* IEEE SA ─ Ethernet & IP @ Automotive Technology Day

*2022/11/09*
**Ethernet Requirements**
Dynamic future use cases

**Cooperative Services** (Awareness, Parking, Intersections...)

**User Applications** (Subscriptions, Infotainment, Games, Tourism, Maps...)

**Automotive Services** (ADAS L1-L3, Individual RT Insurance, Voice assistant...)

**Platform Services** (OTA Update Manager, Control Services...)

**Vehicle Features** (Regulations, Logging, Region...)

**Vehicle Platform** (E/E control, OEM Theme...)

**Hardware** (E/E Phy Topology, Motor...)

---

**Network Layers**

- **Cooperative Services**
- **User Applications**
- **Automotive Services**
- **Platform Services**
- **Vehicle Features**
- **Vehicle Platform**
- **Hardware**

---

**Questions**

- **Question:** What could a complete Ethernet software stack look like?

---

**Frequencies**

1. Static
2. OTA
3. Startup
4. Runtime

**Time**

- Configuration $C^1_{t,1}$
- Configuration $C^2_{t,1}$
- Configuration $C^3_{t,1}$
- Configuration $C^4_{t,1}$
- **Static**
- Configuration $C^1_{t,2}$
- Configuration $C^2_{t,2}$
- Configuration $C^3_{t,2}$
- Configuration $C^4_{t,2}$
- **Startup**
- Configuration $C^1_{t,3}$
- Configuration $C^2_{t,3}$
- Configuration $C^3_{t,3}$
- Configuration $C^4_{t,3}$
- **OTA**
- Configuration $C^1_{t,4}$
- Configuration $C^2_{t,4}$
- Configuration $C^3_{t,4}$
- Configuration $C^4_{t,4}$
- **Runtime**

---

**Neural for dynamic reconfigurations**

**Stack of configurations** for a semi-dynamic and safe network.

---

**Tech Stack**

- Continuous User Experience
- Real-time, Latency, Safety, Security, Energy, Costs
- Interfaces
- Runtime Requests
- Physical Topology
- Cloud-based pre-computing
- In-vehicle Configuration

---

**IEEE SA**

- **Ethernet & IP** @ Automotive Technology Day

---

**Stellantis**
**Chapter 2: State of the Art**

**Next up:** Overview of the current technologies (existing or under development, research, standards)
**Ethernet Timing**

One cable, Mixed-criticality QoS Transport

**Mixed-QoS Networking**

- **Hard Real Time**
  - Time Sensitive Networking
  - **1. Synchronization**
    - 802.1AS Time Synchronization
  - **2. Latency**
    - 802.1Qav Credit Based Shaping
    - 802.1Qbv Time Aware Shaper
    - 802.1Qcr Asynchronous Traffic Shaping
    - 802.1Qbr/bu Frame preemption
  - **3. Reliability & Safety**
    - 802.1Qci Per-Stream Filtering, Policing
    - 802.1CB Frame replication, elimination
  - **4. Management & API**
    - YANG Configuration via NETCONF
    - Dedicated APIs for Qbv, Qav, ...

- **Soft Real Time**
  - Strict priorities, higher bandwidth
  - **5. Strict priority**
    - Statistical analysis Like in CAN buses
    - Traditional priority-based shaping
  - **6. Rate Limiting**
    - 802.1Qav Credit Based Shaping

- **Best Effort**
  - Stream reservation, higher bandwidth
  - **7. Reservation**
    - Bandwidth reservation at design stage
    - Higher bandwidth despite costs
    - Scheduling best-effort aware

> **Question:** How can we make modules interact?
Ethernet Timing
Multiple possible technology combinations

High diversity of available technologies
Choice of solutions
Adaptability
Interoperability
Flexibility

Mixed-QoS Networking

Vendor-specific APIs

Infrastrucure Management

API by Vendor #1
API by Vendor #2
API by Vendor #3
API by Vendor #4

Control
IP
MAC
PHY

Network Routing Stack

Solution 1
- Low-bandwidth
- TSN Time Shaping
- Reconfigurable

Solution 2
- High-bandwidth
- Class-based credit
- Reconfigurable

Solution 3
- High-bandwidth
- Bandwidth overhead
- Priority only

Network Agent

802.1AS
Qbv
Qav
100BASE-T1S
1000BASE-T1S
Multi Gigabyte

Strict Priority

Packet Density

Latencies

Hard RT
Soft RT
Best Effort Traffic

15% Jitter
100% Jitter

2022/11/09
Stellantis — IEEE SA — Ethernet & IP @ Automotive Technology Day
Ethernet Timing
Mixed-criticality QoS requirements

1. Ongoing Interface Projects
   Great start!

2. What about other modules?
   How to build a coherent global configuration?

3. Keyword: Interoperability
   Necessary for collaboration on the next SDV platforms

---

**Time-Sensitive Networking (TSN) Profiles** (Selection and Use of TSN tools)

<table>
<thead>
<tr>
<th>Audio Video Bridging</th>
<th>Front haul</th>
<th>Industrial Automation</th>
<th>Automotive In-Vehicle</th>
<th>Service Provider</th>
<th>Aerospace Onboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>[802.1BA/Revision]</td>
<td>[802.1CM/dec]</td>
<td>[IEC/IEEE 60802]</td>
<td>[P802.1DG]</td>
<td>[P802.1DF]</td>
<td>[IEEE P802.1DP / SAE AS5675]</td>
</tr>
</tbody>
</table>

**Time synchronization:**
- Timing and Synchronization [802.1AS-2020]
  - a profile of IEEE 1588
- Hot Standby [P802.1Asdm]
- YANG [P802.1Asdn]
- Profile Terminology [P802.1Asdr]

**TSN Components**
(Tools of the TSN toolset)

- **Synchronization**
- **Reliability**
- **Latency**
- **Resource Management**

**Zero congestion loss = Bounded latency**

**High availability / Ultra reliability:**
- Frame Replication and Elimination [802.1CB]
- Path Control and Reservation [802.1Qca]
- Per-Stream Filtering and Policing [802.1Qci]
- Reliability for Time Sync [802.1AS-2020]

**Dedicated resources & API:**
- Stream Reservation Protocol [802.1Qat]
- Link-local Registration Protocol [802.1CS]
- TSN Configuration [802.1Qcc]
- Foundational Bridge YANG [802.1Qcp]
- YANG for CFM [802.1Qcx]
- YANG for LLDP [P802.1A8cu]
- YANG for 802.1Qbv/Qbu/Qci [P802.1Qcw]
- YANG & MIB for FRER [P802.1Cbcv]
- Extended Stream Identification [P802.1C8db]
- Resource Allocation Protocol [P802.1Qdd]
- TSN Configuration Enhancements [P802.1Qd]
- LLDPv2 for Multiframe Data Units [P802.1AB8h]
- Multicast and Local Address Assignment [P802.1CQ]

**Note:** A ‘P’ in front of ‘802.1’ indicates an ongoing project.

More on TSN standards and ongoing projects at: https://www.ieee802.org/3/tsn

---

Ongoing Interface Projects
Great start!

What about other modules?
How to build a coherent global configuration?

Keyword: Interoperability
Necessary for collaboration on the next SDV platforms

---

2022/11/09
Stellantis — IEEE SA — Ethernet & IP @ Automotive Technology Day
**Ethernet Routing**
Dynamically reconfigurable networking

**Traditional Networking**
*Local Switch Management*

1 Switch, 1 interface
Local management
Custom reconfiguration solutions

Works, but hard to reconfigure

**Software Defined Networking**
*Centralized Orchestration*

Common interfaces
Centralized monitoring
Global & Dynamic (re)configuration

(Still under research)

**Cold Topics**
- VLANs & VxLANs
- Cybersecurity
- Tunneling
- Cloud Bridge
- CAN/LIN Gateway
- External Gateway
- Firewall
- Energy management

**Problem:** TSN must be configured based on external parameters...

**Question:** How to create coherent configurations through the stack?
**Ethernet Control Plane**

**Dynamic Control Services**

1. **Network Scheduling**
   - Global Configurator
   - Time sync Manager
   - Firewall Scheduler
   - Offline Scheduler
   - SDN Scheduler
   - TSN Scheduler

2. **Network Monitoring**
   - Intrusion Detection
   - Self-healing
   - Diagnostics

3. **Gateways**
   - External Configurator
   - Hardware Discovery
   - Legacy Configurator
   - Cloud Bridge

**Interaction Map**

**Problem:** How to make all services interact together?

4. **Vehicle Management**
   - Vehicle State Manager
   - Telematics & Logging
   - Global Configuration

5. **Service Management**
   - SOTA/FOTA Update Manager
   - Lifecycle Manager
   - Service Monitor

6. **Communication Management**
   - Dynamic Flow Mapping
   - Service Discovery
   - Load balancing

7. **Platform Management**
   - Time Synchronization Manager
   - Global Diagnostics
   - Data Collector
Conclusion so far...

1. High diversity is good for choice
2. Different combinations will exist
3. 1 module, many implementations

New objective

1. **Objective:** Interchangeable solutions
2. **Method:** Interfaces + States standardization
3. **Solution:** Take inspiration from Cloud Computing
Chapter 3: Integration

Next up: Inspirations from other industries that could help us define a fully integrated Ethernet solution
Ethernet Requirements
For a continuous user experience

SDV Ecosystem
Hardware is abstracted from Services

1. Real-time traffic shaping
2. Dynamic Routing
3. Mixed QoS Transport
4. Security
5. Dynamic reconfiguration
6. Synchronization
7. Real-time
8. Signals ≠ Services
9. Security policies
10. Legacy & V2X Gateways

Architecture Design Requirements
- Time-to-market: Lightweight
- Easy to develop: Interface to ZOA
- Maintainability: Interface to SOA
- Security: Dynamic reconfigurations
- Standardized: Traceability
- Legacy compatible

Vehicle Applications
- Connected Services
- Comfort & Safety
- Energy Efficiency

Middleware (Car.OS)
- Automotive SDK
- Platform Management Services
- Automotive Core Services
- Vehicle Abstraction Layer

Embedded Software
- Hardware Environments
- Software Environments
- Network Environments

IEEE SA
- Ethernet & IP @ Automotive Technology Day

2022/06/27
Stellantis — IEEE SA — Ethernet & IP @ Automotive Technology Day
Cloud Computing Architectures
Everything-as-a-Service (XaaS)

Common features:
- ✅ Service Scheduling
- ✅ Service Discovery
- ✅ Network config.
- ✅ Self-Healing
- ✅ Redundancy

Automotive only:
-🚗 Heterogeneous environments
-🚗 Low & embedded resources
-🚗 Real-time functions
-🚗 Safety

IT only:
-❌ Storage Orchestration
-❌ Backups & Rollbacks
-❌ Batch Execution

Software Defined Vehicle
“Data Center on Wheels”

1. Flexible & Instantaneous Updates
2. Easy monitoring & diagnostics
3. Virtualized networking
4. Plug-and-play solutions
5. Dynamic service scheduling
6. Standardized Interfaces
SDVs are (kind of) like the Cloud
Similar problem, different constraints

Why not do the same?
Software Defined Vehicle Stack
Dynamic service management

Current challenges:
1. Virtual environments
2. QoS Management
3. Dynamic mechanisms

Question: How can we design and implement a self-contained Ethernet?
Cloud Orchestration Architecture
Set the desired state, watch it unfold!

Centralized State Representation
(Current & Desired)

Infrastructure Endpoint

Centralized Control Services
(Interchangeable)

Local Agent
(configures network)

Network Endpoint
(Load Balancing)

Many startups offer independent plug-and-play components

APIs are fixed!

Ashish Patel, “Kubernetes - Architecture Overview”, Medium, 2021
Implementation proposition
A high-level overview

Full Network Stack
Virtual + Ethernet

⇒ Standards will be needed across suppliers!
Discussion

Chapter 1
Requirements

Chapter 2
State of the Art

Chapter 3
Proposition
Discussion
Our take on the next steps

Summary
1. **Interchangeability** is what the industry needs
2. A **common language** can be made from standardized APIs
3. **Ethernet-as-a-Service** is a promising way to organize our standards

Propositions
1. **Discuss** how to define an application’s requirements
2. **Standardize** the common vehicle state representation first
3. **Adopt XaaS** from Cloud Computing for a loosely coupled architecture

Takeaway
SDV development will need attention on “Common Data Representation for Ethernet Requirements”
Thank you for your attention!

Pierre LACLAU, Xiaoting LI and Trista LIN
IEEE SA 2022 – Automotive Technology Day
pierre.laclau@stellantis.com

Designing a safe, real-time, secure, embedded, and cost-effective Data Center that can be used like a Smartphone (oh, and it can drive)

All images imported in this presentation are under open licenses. All rights reserved for images produced by Stellantis.

Watch the talk replay on IEEE SA’s YouTube channel