QoS-Predictable SOA on TSN: Insights from a Case-Study

Josetxo VILLANUEVA, Groupe Renault
Jörn MIGGE, RealTime-at-Work (RTaW)
Nicolas NAVET, University of Luxembourg / Cognifyer.ai

Automotive Ethernet Congress
February 09-11, 2021 | Munich, Germany
Outline & Objectives

✓ Takeaways learned in the design and implementation of the Renault FACE Service Oriented Architecture over Ethernet TSN backbone
✓ Concrete illustration of the use of services for two QoS-demanding use-cases: Light Service Architecture (actuator) & Smart Sensor Fusion Use-Case
✓ The challenges in configuring Ethernet TSN for services & possible solutions
✓ Experiments: optimizing TSN configuration and the difference it makes in timing & memory
1. Designing next-generation service-oriented E/E architectures
SOA & Central Computing from OEM perspective

✓ SOA & Central Computing benefits
  – Decoupling of HW & SW
    • Service & clients can be instantiated everywhere
  – Re-use & modularity of Services (Building Blocks)
  – Ease software-based innovation
  – Personalization, new business models, by software updates

✓ Change of communication paradigms
Multi-platform Middleware (Eg. SOME/IP), flexibility & automation of network configuration, guarantee of network QoS
A hierarchy of services & applications

Services are executed:
✓ In dedicated ECUs, ex: cam. & radars
✓ In zone controllers for basic services, ex: sensor data processing
✓ In High Performance Computing ECUs for composed services, ex: ADAS
✓ In the cloud, ex: infotainment
✓ In the infrastructure, ex: speed limit

Services can be grouped to define re-usable Software Components or complete Virtual ECUs
SOA Use-Case #1: Lighting Services

How to configure Service Communication when thousands of flows generated by hundreds of Services are competing for network resources?
SOA Use-Case #2: Smart Sensor Fusion Use-Case

✓ Smart Sensors: Translate analogue data into Service communication (Eg. CAM, Radar, ...)

✓ Solution shaping with CBS (in HW or SW), pre-shaping (w/o) sub-bursts

✓ Which protocol to use, to ease spacement?
  – Upon choice, none either or both request & response can be segmented, and thus
  – Not all services, REQ & RESP of the same service may require the same QoS mechanisms
2. FACE E/E architecture: Topology, Protocol Stack, Services Characteristics and their QoS Requirements
Ethernet Simulation Model of FACE E/E architecture

1 Central Computer (“Physical Computing Unit”) → PCU hosts Composed Services & Applications

5 Zone Controllers (“Physical Interface Units”) → PIUs host Basic Services

17 ECUs on Ethernet including 3 front/rear cameras, 2 radars, 3 displays, off-board module + dedicated ECUs e.g. for I/Os and chassis on 5 split CANs behind PIUs

100Mbit/s links but two 1Gbit/s links
Protocol Stack with a Focus on Segmenting & Shaping

- **TCP**: reliable & segmented transmissions but not real-time!
- **SOME/IP TP**: timing predictable & segmented transmissions but no shaping capability
- Segmenting server’s messages into several *Events* may be a work-around for not using SOME/IP TP.
  - Defining Event period is not enough. Need to space *Events* transmission.
- **CBS**: limited memory in egress ports, additional shaping in SW in sender’s comm. stack may be needed, but no standard solution.
Non-functional Requirements on Services

1. **Timing** (→ service colocation constraint)

2. **Memory** (SW, HW), **computing power**, **funct. modes** (power)

3. **Security & Safety** (e.g., redundancy on ≠ cores, core ASIL levels)

- Predictable real-time behavior required to suppress dedicated chassis or ADAS ECUs
- **Timing depends on the execution platform**: Classic platform on dedicated core more predictable than Adaptive platform (HPC does not mean real-time)
- **Service allocation is key for optimized resource usage / extensibility** → design-space exploration coupled with timing-accurate simulation can help optimize the allocation
- **TSN QoS mechanisms such as shaping have an impact on memory usage in HW & SW**
Configuration should ensure that all streams, not only services, meet their timing constraints.

How to set priorities and TSN QoS mechanisms?

### Request-Response communication

The server executes the method corresponding to the request.

- **Client**
  - `request`
  - `deadline`

- **Server**
  - `method execution (RPC)`
  - `response`
  - `$\delta > $ exclusion time`

In addition, the server can send:

- (periodic) **Notification Events** to the subscribed clients.

- **Configuration challenges with services:**
  - ∃ deadlines on req.-resp. transactions not only individual transmissions.
  - Some messages, typ. resp., can require segmenting & shaping.
  - ≠ timing constraints for each req.-resp. transact. and Events of the same service.
  - Thousands of streams! Which calls for an automated process based on models.
Characteristics of the Services

✓ Subscribers are SW components not ECUs
✓ “period” for calls to methods means their *exclusion time*

**Smart Sensor Basic Services**
ex: object & infrastructure detection

✓ >60 – Typ. period: 20-100ms
✓ Segmented messages (mostly)
✓ Hard deadlines

**Basic Services**
ex: environment sensing

✓ >40 – Typ. period: 10-100ms
✓ Non-segmented messages (mostly)
✓ Hard deadlines

**Composed Services**
ex: fusion & resources arbitration

✓ >60 – Typ. period: 20-100ms
✓ Typ. 5x larger than basic services
✓ Hard deadlines

**Cloud services**
ex: heating remote control

✓ 10 < - Typ. period: 5s
✓ Segmented messages
✓ Soft deadlines

A single service can generate a lot of traffic! E.g. 100 unicast streams and 5 multicast streams for a composed service offering 10 methods and 5 events to 10 clients
Non-Service Related Traffic

**CAN (FD) Snapshots**
Re-forwarded CAN (FD) frames
- $\approx 30$ – period: 10 or 20ms
- Non-segmented messages
- Hard deadlines: 5ms

**TFTP + RTP Video Streams**
ex: infotainment
- $<5$ – Period: from 33ms (30FPS) to 1s
- Segmented messages
- Mixed deadline and throughput constraints

**TCP & HTTP streams**
ex: off-board comm., DoIP
- $\approx 10$ – sporadic: typ. 1s
- Segmented messages
- Throughput constraints
3. Optimized TSN configuration for services to maximise network capacity and reduce memory consumption
% of overloaded networks as a function of the number of services: KPI of Evolutivity

Overloaded network = the load of one link or more is higher than 100% → no TSN policy can meet the timing constraints

- > 10% of the networks are overloaded above 75 services (3920 flows)
- Suggests that whatever the TSN policy, this architecture is suited for at most 60-80 services
- Bottleneck link: CGW to PCU2. Switching it to 1Gbit/s would increase the network capacity

©2021 - Renault- RTaW - UL - Cognifyer
Manual Traffic Prioritization

- TSN mechanism: priority, no shaping
- Traffic classification based on deadlines with manual tuning:
  - Urgent Services: deadlines < 10ms

<table>
<thead>
<tr>
<th>Class</th>
<th>Priority</th>
<th>Express</th>
<th>Scheduling Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAN SNAPSHOTS</td>
<td>7</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>SERVICES (urgent)</td>
<td>6</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>SERVICES</td>
<td>5</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>ADAS-SERVICES</td>
<td>4</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>3</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>TFTP+TCP</td>
<td>2</td>
<td>false</td>
<td>FIFO</td>
</tr>
</tbody>
</table>

- Schedulability limit is 27 services
  - 1469 streams for a max link load of 26.6%

- Shaping not helpful: ADAS-Services are at low priority level & shaping non-segm. packets of limited use
- Preemption not helpful as deadline misses do not occur at top priority
Traffic prioritization with Concise Priorities algorithm

- TSN mechanism: priority, no shaping
- Automated traffic classification
  - Streams of different types will be mixed at all priority levels

<table>
<thead>
<tr>
<th>Class</th>
<th>Priority</th>
<th>Express</th>
<th>Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 0</td>
<td>7</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 1</td>
<td>6</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 2</td>
<td>5</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 3</td>
<td>4</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 4</td>
<td>3</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 5</td>
<td>2</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>Class 6</td>
<td>1</td>
<td>false</td>
<td>FIFO</td>
</tr>
<tr>
<td>TCP-TCP</td>
<td>9</td>
<td>false</td>
<td>FIFO</td>
</tr>
</tbody>
</table>

- All 8 priority levels are used
- Deadline misses are top priority
- Preemption not helpful as blocking from low prio. packets limited in delays
- Shaping only marginally effective for memory as there is little slack time

Streams of the “critical” class (prio. 6)

Schedulability limit is 72 services

\[ \Rightarrow \text{3785 streams for a max link load of 53.8\%} \]
4. Algorithm-based Configuration

Reducing Memory Usage with Shaping

Per-egress ports max. memory usage:
with CBS (red) and without (black)

System with 20 services
(1152 flows, manual traffic classification)

50KBytes per port

- CBS CMI: 1333us, comparable gains with CMI 250us or SW shaping on senders
- Per device memory usage reduction with CBS – max: 97%, average: 12.3%
- No gain in switch ports where video & radar streams are merged
- Memory in egress ports can be exceeded even with moderate load!

[✓] In practice, ∃ constraints not only on latencies but memory & CPU usage → ≠ TSN sched. sol. lead ≠ tradeoffs
[✓] Shaping improves delays for lower priority packets but not always memory usage

©2021 - Renault- RTaW - UL - Cognifyer
Conclusion and a look forward
Takeaways

Finding the right granularity for services definition is fundamental — “big” services, generating each up to 1Mbps of data, have been used in this work but is it always the right choice?

Correct service execution relies on the QoS provided by Ethernet TSN, which requires proper protocols selection and configuration.

Challenges in ensuring network QoS:
- Tight collaboration between OEM and Tier1/2 needed during integration phase due to limited maturity in SOA/Central Architecture (COTS, tools, ...)
- Shaping in Autosar comm. stacks?
- Non-deterministic execution platforms place additional constraints on network (e.g., retransmissions)
- System complexity & subtle interactions between QoS mechanisms calls for model-based configuration and verification for optimized resource usage

©2021 - Renault- RTaW - UL - Cognifyer
Thank you for your attention!