RESPONSE TIME ANALYSIS FOR FIXED PRIORITY SERVERS

ARNE HAMANN, DAKSHINA DASARI, JORGE MARTINEZ, DIRK ZIEGENBEIN

ROBERT BOSCH

BOSCH

Response Time Analysis for Fixed Priority Servers

Outline

- Motivation
  - Why are the “industry guys” interested now in this “old” server based scheduling technology?
- Review of the State-of-the-Art
  - Everything has already been solved > 10 years ago! … really?
- Proposed Response Time Analysis for Fixed Priority Servers
- Experimental Results
- Conclusion

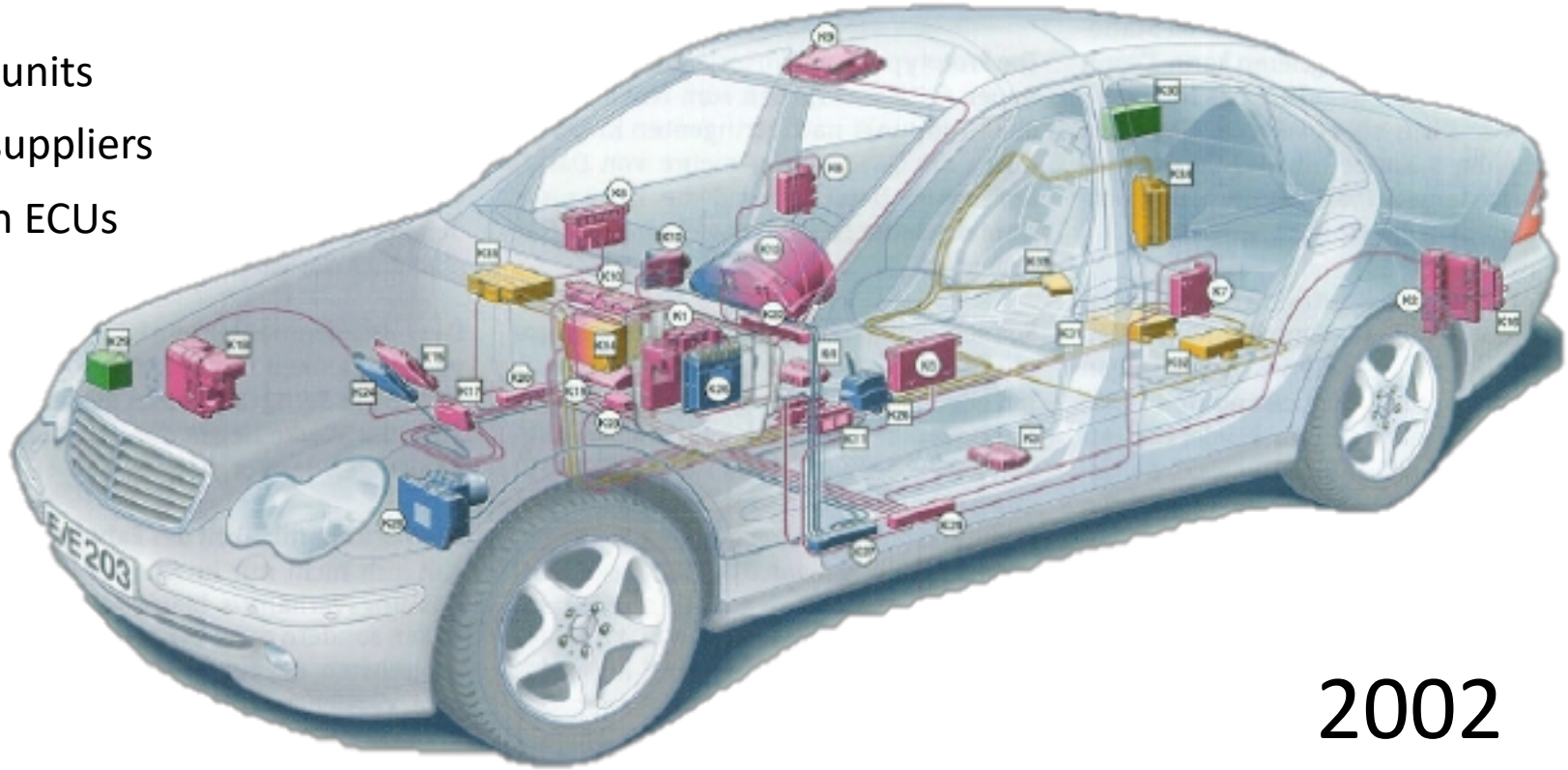
Response Time Analysis for Fixed Priority Servers
Outline

- **Motivation**
  - **Why are the “industry guys” interested now in this “old” server based scheduling technology?**
- Review of the State-of-the-Art
  - Everything has already been solved > 10 years ago! … really?
- Proposed Response Time Analysis for Fixed Priority Servers
- Experimental Results
- Conclusion

Response Time Analysis for Fixed Priority Servers

Automotive Systems – How it always was …

- Up to 100 electronic control units
- ECUs delivered by different suppliers
- Only limited “SW sharing” on ECUs
- Mainly integration on network level



2002

OEMs: “ We will reduce the number of ECUs significantly”

Response Time Analysis for Fixed Priority Servers

Conway’s Law

Any organization that designs a system

will produce a design whose structure

is a copy of the organization's communication structure.

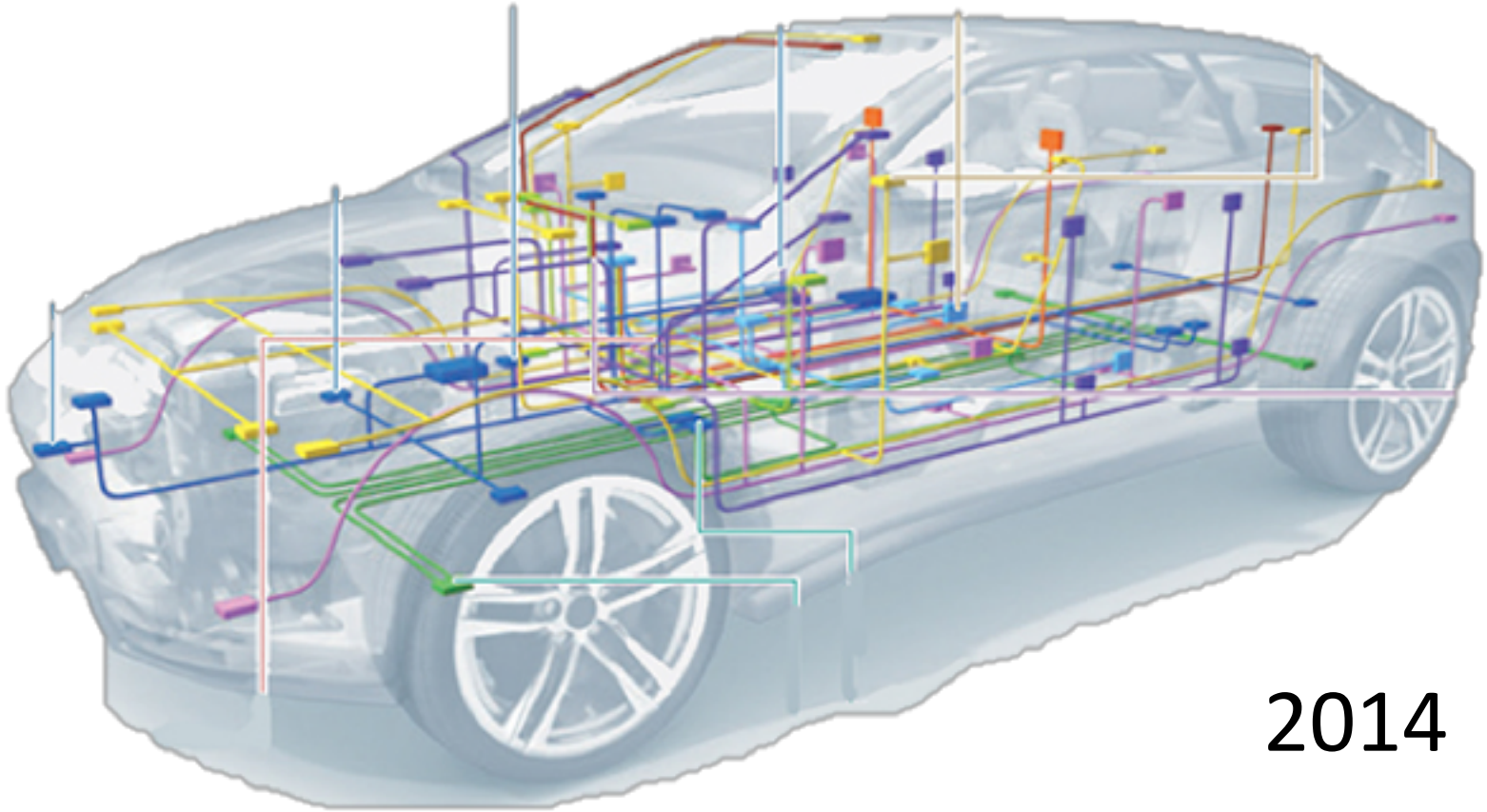
Melvin Conway, 1968

Response Time Analysis for Fixed Priority Servers

Automotive Systems – …until recently has been…

- Conway's Law at work

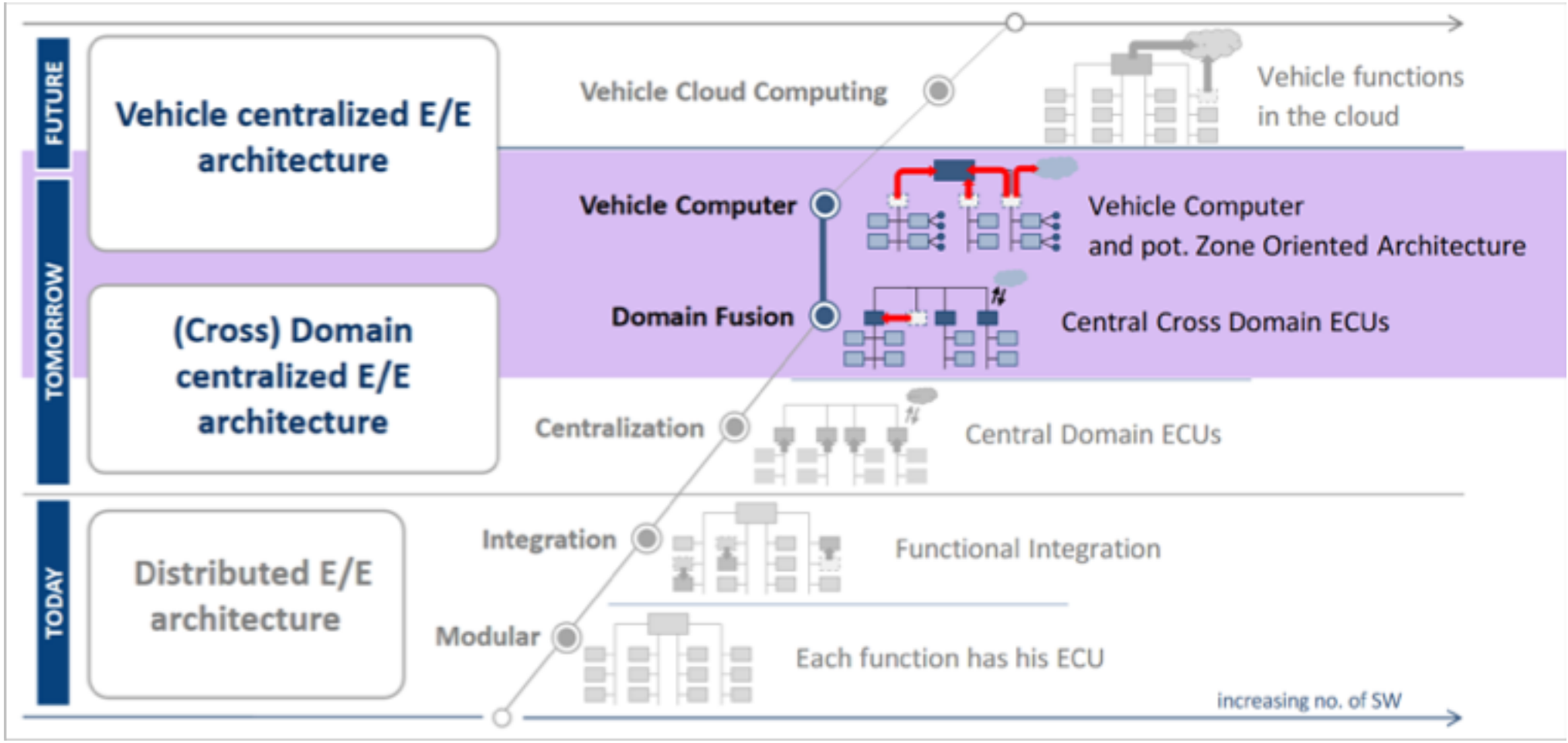
Typical Org-Chart



2014

Response Time Analysis for Fixed Priority Servers

Automotive Systems – …but tomorrow not be anymore



Response Time Analysis for Fixed Priority Servers

The Move to Centralized E/E Architectures – Why Now?

- Conway's law overcome by market entry of new OEMs
  - starting on a clean sheet without legacy (organization)
- New functionality requires new powerful HW platforms
  - for the first time there exists a "vehicle computer" to absorb functionality
- Most cost-effective way to…
  - Realize fail-operational behavior (as required by e.g. automated driving)
  - Implement cloud connectivity
  - Provide spare resources for upgrades

Response Time Analysis for Fixed Priority Servers

Challenges – New and Old

- Energy & cost efficiency
- Predictability
- **Efficient isolation**
- **Composability**

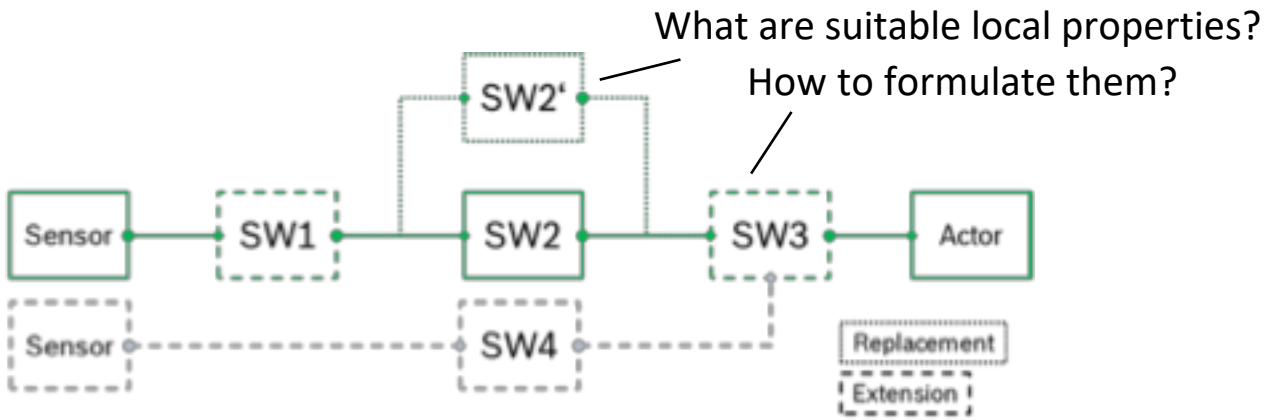
} Shift of integration from network to ECU level

- System safety
- Migration of legacy code

Response Time Analysis for Fixed Priority Servers

Composability is Key to Master Complexity

- "SW Sharing"
  - SW from different suppliers is integrated onto the same platform
  - Need for **efficient** temporal isolation
  - Composability for the integration needed
- Complexity due to upgrades
  - Adding or exchanging also safety-critical software components during product life-time
- Complexity due to variants
  - Build 1000s of variant products from one product line



Response Time Analysis for Fixed Priority Servers

Efficient Isolation

TDMA

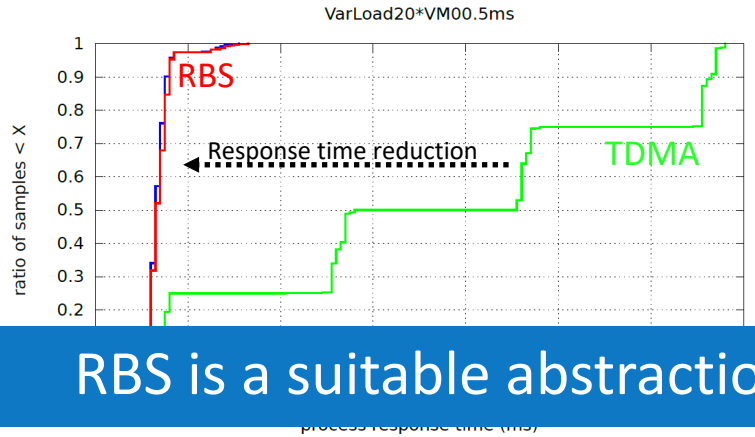
Periodic task A:
wasted CPU time due to mismatch
slot sizes ←→ execution demand

Sporadic task B:
wasted CPU time due to wasted slots
that are reserved to ensure responsiveness

RBS

Periodic task A:
only needed CPU time is used,
remaining budget is released back

Sporadic task B:
no wasted CPU time, responsiveness
is assured through guaranteed budget



- TDMA is standard scheduling paradigm for isolation…
  …but quite inefficient.
- Reservation-based scheduling (RBS) as new scheduling paradigm for integration platforms
  - Budget-based reservations instead of fixed time slices (TDMA)
  - Efficient temporal isolation compared to TDMA due to work-conservation and capacity sharing
- Simulations of concrete vehicle computer project show shorter response times & more efficient system utilization

RBS is a suitable abstraction for composability/efficiency in time domain

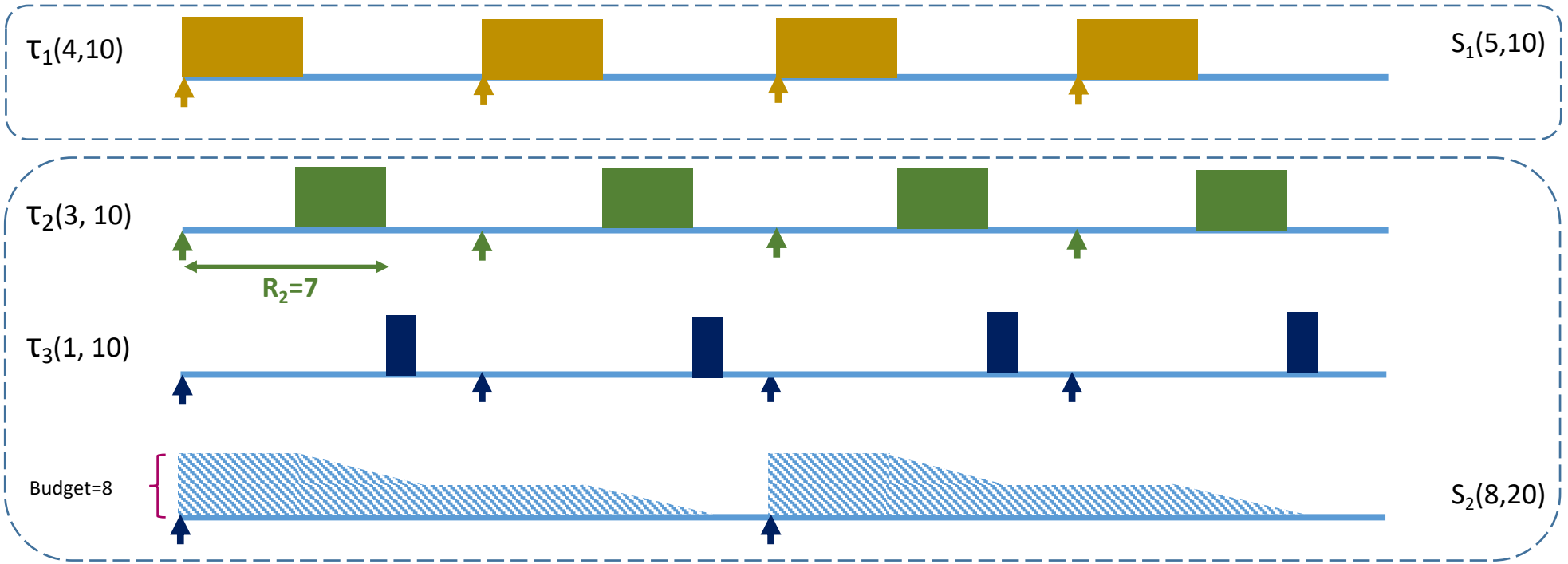
BOSCH

Response Time Analysis for Fixed Priority Servers
Outline

- Motivation
  - Why are the “industry guys” interested now in this “old” server based scheduling technology?
- **Review of the State-of-the-Art**
  - **Everything has already been solved > 10 years ago!** **… really?**
- Proposed Response Time Analysis for Fixed Priority Servers
- Experimental Results
- Conclusion

Response Time Analysis for Fixed Priority Servers

Example to Demonstrate Pessimism in SoA



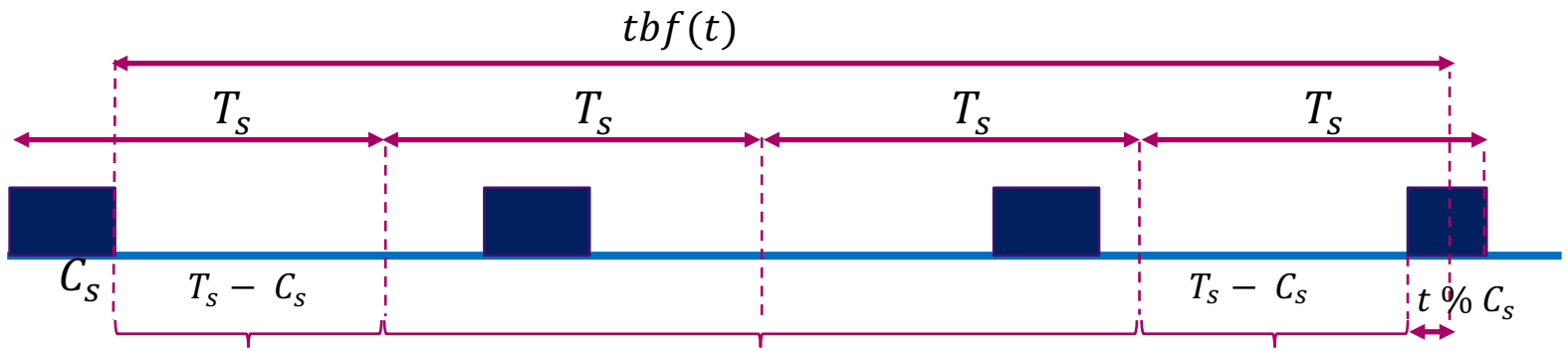
- τ_2 is completed in the first replenishment interval of S_1 …
- … and is only delayed by 1 execution of τ_1
- System-wide behavior repeats after hyper period of 20

Response Time Analysis for Fixed Priority Servers

SoA: Service Time Bound based Approaches

- tbf(t): The maximum time for the server to provide “t” units of service

$$r_i^k = tbf(I_i^k) \iff I_i^k = C_i + \sum_{j \in hp(i)} \left\lceil \frac{r_i^{k-1}}{T_j} \right\rceil * C_j$$



$S_2(8,20)$

1. systematic internal budget depletion

2. get C_s time units every T_s

3. maximum external interference

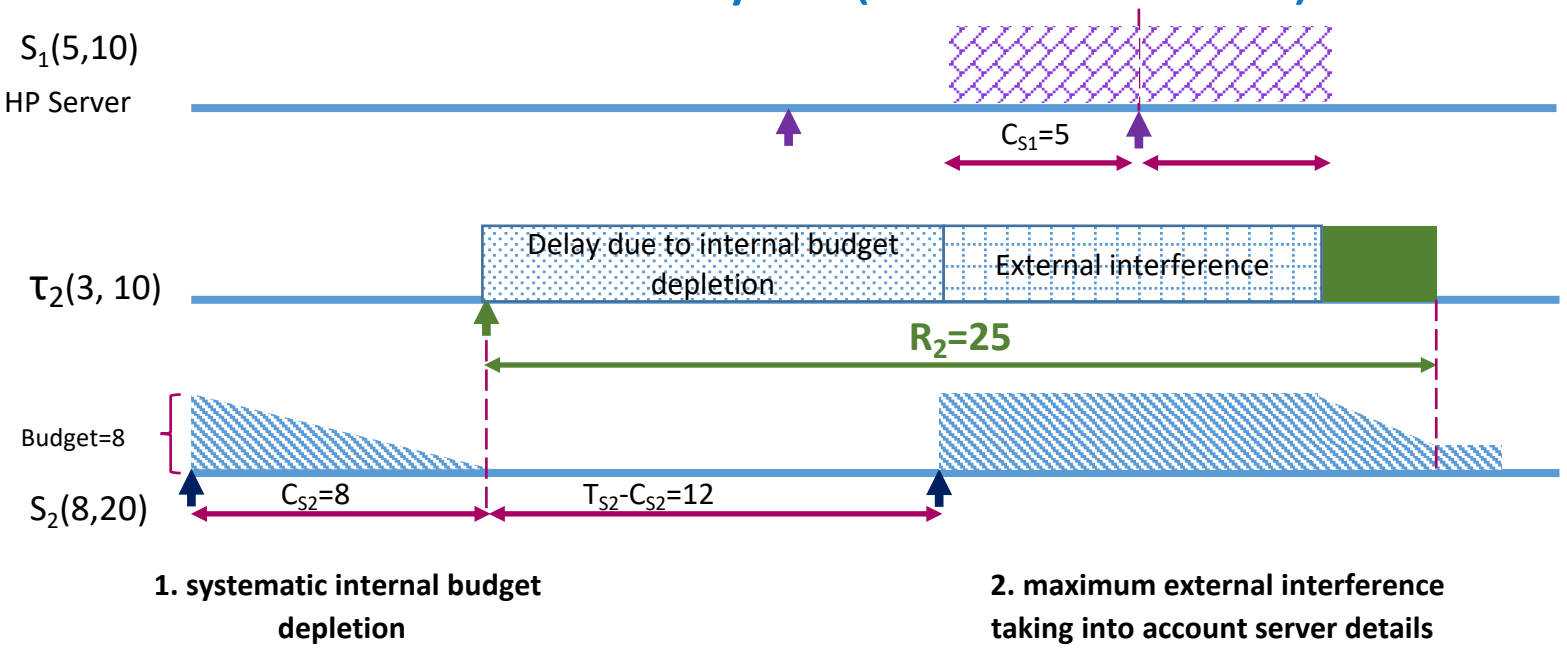
- Approach is agnostic to other servers/workloads in the system
- In order to provide 3 time units the tbf function computes 27 time units (12 + 12 + 3)
- Worst-case response time of $\tau_2 = 27$ (>> 7)

Insik Shin and Insup Lee. 2008. Compositional Real-time Scheduling Framework with Periodic Model

Response Time Analysis for Fixed Priority Servers

SoA: Critical-instant Based Analysis (Davis & Burns)

R. I. Davis and A. Burns. 2005. Hierarchical Fixed Priority Pre-Emptive Scheduling



- Approach is agnostic to other workloads in the system but considers server parameters and type
- Worst-case response time of Task τ_2 = 25 (>> 7)

SoA prohibitively pessimistic for application in automotive systems

Response Time Analysis for Fixed Priority Servers
SoA: “Optimistic” in Case of Deferrable Servers 1/3

- Why is the SoA “optimistic?”
- The SoA assumes that each server is capable of providing **C_s time units every T_s time units**
- In the presence of Deferrable Servers this **“service contract” is not trivial to check/ensure**
  - Complex situations where **multiple double hits** of different deferrable servers coincide are possible **depending on the replenishment periods and task arrivals**
  - Sometimes the **“service contract”** might be violated yielding **“optimistic” results**
  - Insidious, since other sources of pessimism (e.g. initial T_s-C_s delay) might compensate for this optimism
- To be fair …
  - System configurations where this happens do not conform to the assumptions of the SoA analysis
  - However, this far from trivial to check, and thus **the SoA analysis is not applicable** for systems containing Deferrable Servers

Response Time Analysis for Fixed Priority Servers

SoA: “Optimistic” in Case of Deferrable Servers 2/3

- Deferrable Server $S_1(C_s = 1.5, T_s = 5)$
  - Task $\tau_1(T_1 = 11, C_1 = 3)$
- Deferrable Server $S_2(C_s = 1, T_s = 3)$
  - Task $\tau_2(T_2 = 200, C_1 = 50)$

- Server reservation 63,33%
- Task utilization ~ 52,3%

- Applying the SoA analysis from Davis & Burns yields a worst-case response time of 153 for τ_2
- Optimistic!! The real worst-case response time of τ_2 is equal to 154

R. I. Davis and A. Burns. 2005. Hierarchical Fixed Priority Pre-Emptive Scheduling

Response Time Analysis for Fixed Priority Servers

SoA: “Optimistic” in Case of Deferrable Servers 3/3

Deferrable Servers cannot be treated as black-boxes in analysis

Response Time Analysis for Fixed Priority Servers

Outline

- Motivation
  - Why are the “industry guys” interested now in this “old” server based scheduling technology?
  - Review of the State-of-the-Art
  - Everything has already been solved > 10 years ago! … really?
- **Proposed Response Time Analysis for Fixed Priority Servers**
- Experimental Results
- Conclusion

Response Time Analysis for Fixed Priority Servers
Proposed Analysis

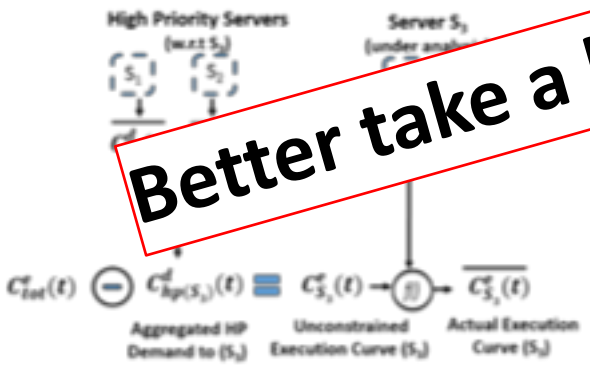
- Based on **Service/Demand Curve** abstractions
  - Not in delta-time but in time domain
- Capable of considering **actual interference of other servers and workloads**
  - Mixed Server Polling Periodic and Deferrable Server
  - Periodic task with offsets, arbitrary deadlines, backlogged executions
- Scope of the analysis motivated by **integration projects**
  - Several legacy systems that are OSEK based need to be integrated
  - In future: extension to more irregular activation patterns for upcoming integration scenarios involving heterogeneous applications from different domains
- We assume **partitioned scheduling**
  - Each reservation can serve multiple task but each task is served by exactly one reservation only
  - Most realistic setting for introducing the technology in industry

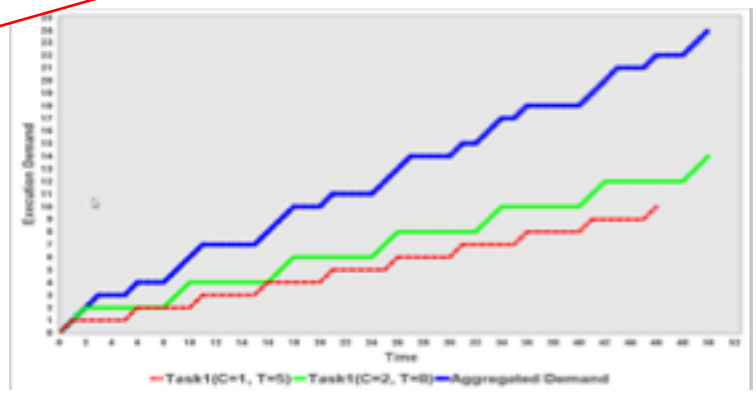
Response Time Analysis for Fixed Priority Servers

Proposed Analysis – How it works

- **Server supply** and **workloads demands** are modeled as **curves**
- **Internal server demand** is computed by aggregating the individual workload demands
  - … and constrained to the server specifics (type and parameters)
- **External interference** is computed by aggregating demand curves of higher priority servers
- **Service available to the server** is computed considering external interfer… …ernal server demand
  - This service is distributed among the constituent tasks on a priorit… …esponse times

Better take a look into the paper ;-)





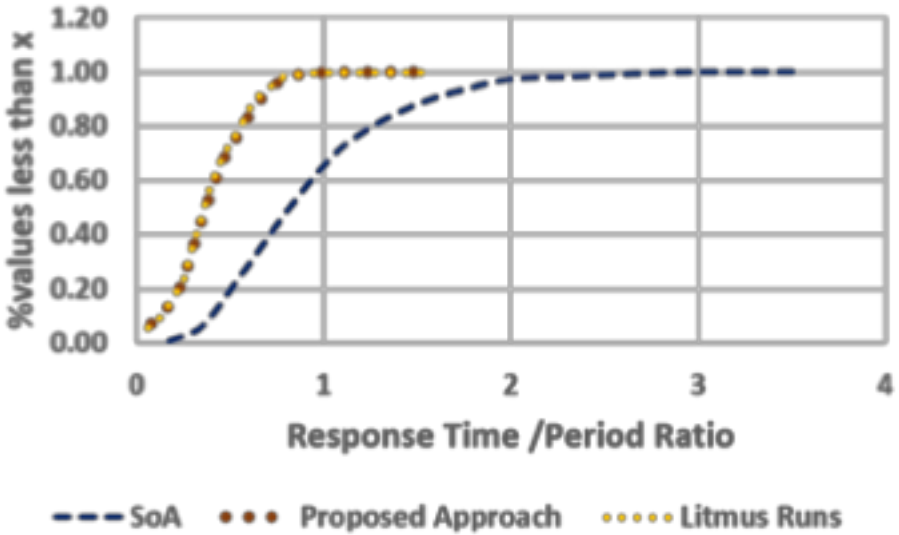
Response Time Analysis for Fixed Priority Servers

Outline

- Motivation
  - Why are the “industry guys” interested now in this “old” server based scheduling technology?
  - Review of the State-of-the-Art
  - Everything has already been solved > 10 years ago! … really?
  - Proposed Response Time Analysis for Fixed Priority Servers
- **Experimental Results**
- Conclusion

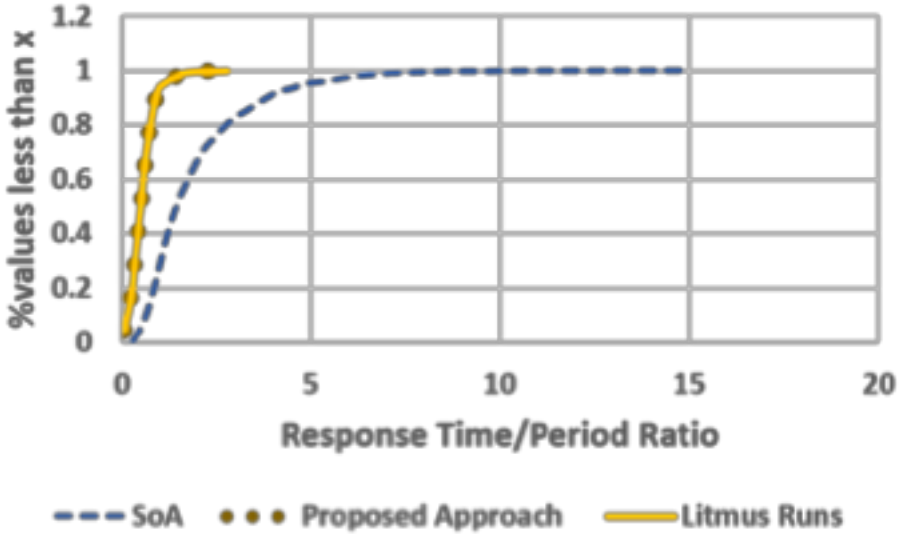
Response Time Analysis for Fixed Priority Servers
Experiments 1/2

- 500 periodic task sets per experiment (UUnifast)
- Proposed analysis vs. SoA vs. Litmus runs
- Results visualized with CDFs displaying normalized response times wrt. to activation period



5 tasks mapped to a single DS

Utilization = 60%

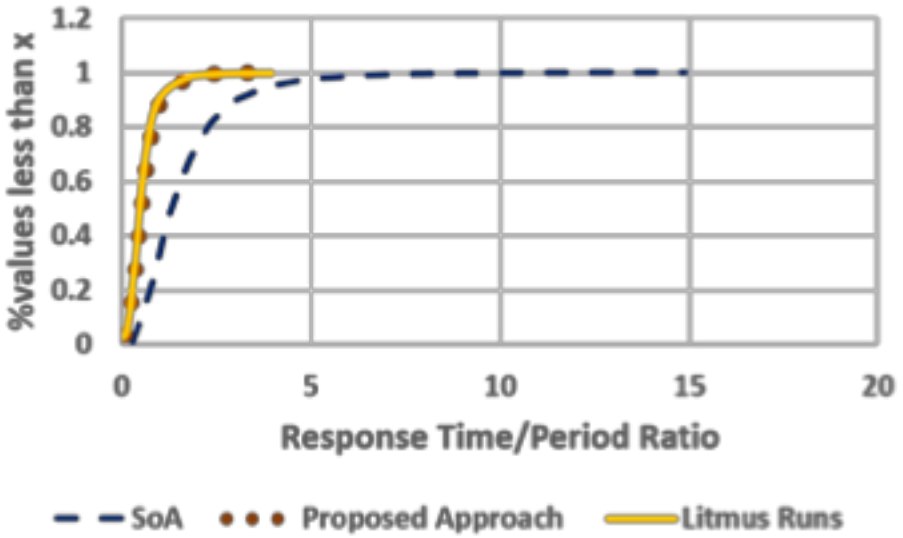


7 tasks mapped to two DS

Utilization = 70%

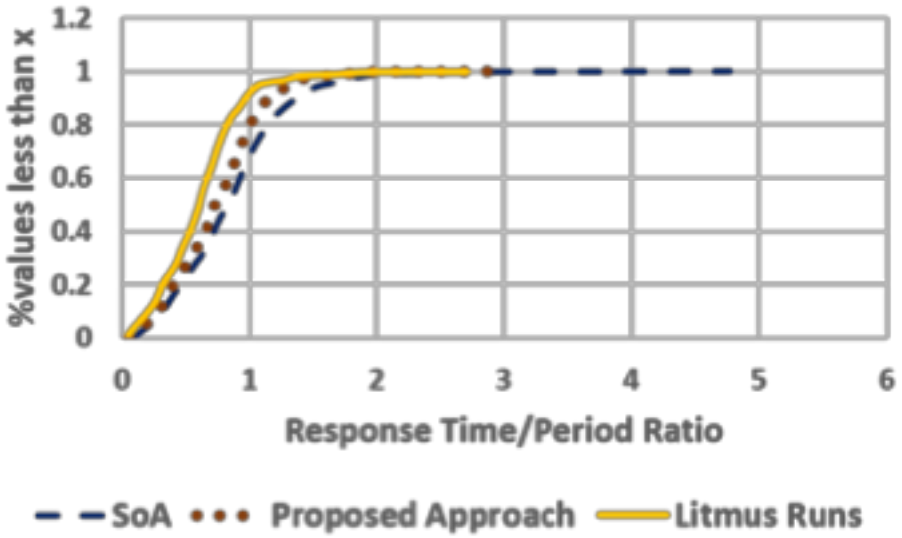
Response Time Analysis for Fixed Priority Servers

Experiments 2/2



10 tasks randomly mapped to three DS / PS

Utilization = 70%



7 tasks mapped to three PS

Utilization = 60%

Response Time Analysis for Fixed Priority Servers
Conclusion

- Reservation-based scheduling of high interest for centralized E/E architectures in automotive systems
- SoA in scheduling analysis for reservation-based scheduling too conservative for intended scope of usage
- “Black box” abstraction for Deferrable Servers not reasonable
- Proposed analysis significantly improves precision and extends supported application model

THANK YOU

QUESTIONS ???

BOSCH

ECRTS

31th Euromicro Conference on Real-Time Systems

9-12 July 2019 | Stuttgart, Germany

BOSCH