CAN in Automotive Applications: a Look Forward

Nicolas NAVET (INRIA/RTaW), Hervé PERRAULT (PSA)

13th International CAN Conference
Hambach Castle, March 5-6 2012.

March, 5 2012

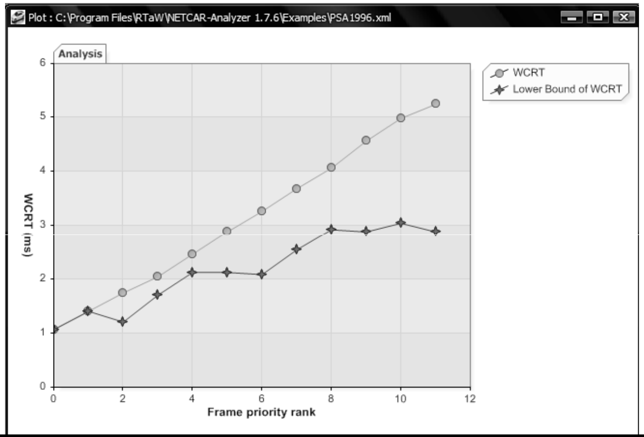
Automotive CAN: the early days (1/2)

Priority	Sender node	DLC	Period (ms)
1	Engine Controller	8	10
2	Wheel angle sensor	3	14
3	Engine Controller	3	20
4	AGB	2	15
5	ABS	5	20
6	ABS	5	40
7	ABS	4	15
8	Body gateway	5	50
9	undisclosed	4	20
10	Engine Controller	7	100
11	AGB	5	50
12	ABS	1	100

6 stations, 12 frames, 21% load

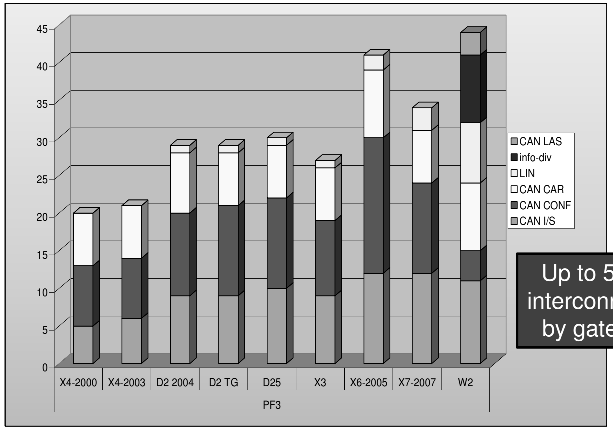
Early CAN project at PSA (1996, see [1])
250kbit/s

Automotive CAN: the early days (2/2)


Plot : C:\Program Files\RTaW\NETCAR-Analyzer 1.7.6\Examples\PSA1996.xml

Worst-case latencies (=response times) are less than 5.5 ms
NETCAR-Analyzer screenshot

Proliferation of ECUs and buses



Up to 5 CAN interconnected by gateways

# ECUs and buses in some PSA projects between 2000 and 2010 [2]

Today's set of messages

- **Size :** Up to 20 nodes and 100 frames
- **Bus-rate :** 250 or 500kbits
- **Load :** > 50%, sometimes 60% or more …
- **Max latencies :** 5ms or less
- **Gateways :** CAN/CAN or CAN/FLEXRAY induce delays and bursty traffic.
- **Aperiodic traffic** (eg, Autosar mixed transmission mode)

"easy" integration for the OEM till 35-40% - precise performance evaluation needed beyond

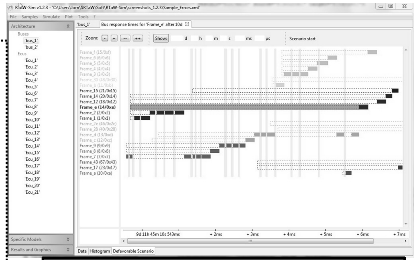
NETCARBENCH is a GPL licensed software to generate "realistic" and non confidential CAN message sets according to a set of user-defined parameters. Available at www.netcarbench.org

RTaW : help designers build truly safe and optimized systems

- **Services and Software for :** architecture design, ECU and network configuration, formal and temporal verification (simulation, analysis, trace-inspection)
- **Communications systems :** CAN, FlexRay, AFDX, industrial Ethernet, TTP, etc …
- **CAN customers:** PSA and Renault

– Most software tools are downloadable at www.realtimeatwork.com / we provide R&D, support and custom extensions

– No black box software: we publish all algorithms that are implemented (ongoing)



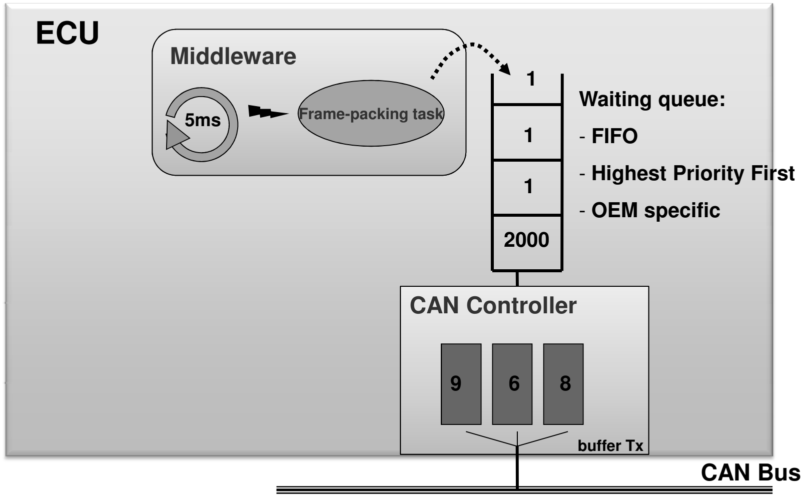
RTaW-Sim CAN simulator

2

Optimizing CAN networks

What levers do we have and what it implies ?

Automotive CAN communication stack : a simplified view



Optimizing CAN : meeting performance and robustness constraints at higher load

An industrial requirement

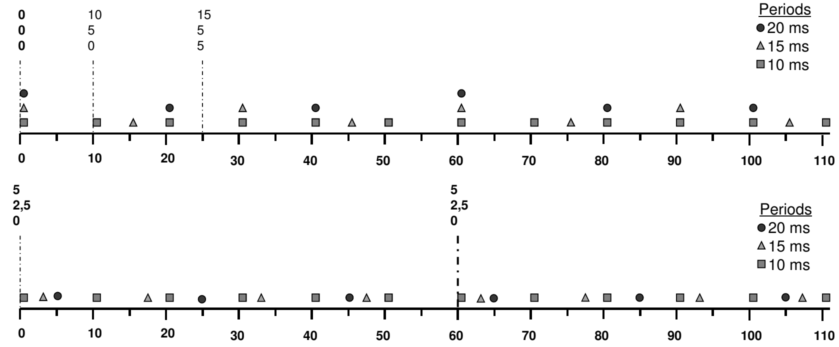
- Reduce architecture complexity, HW costs & weight, consumption and emission
- Avoid industrial risks and costs of new technologies
- Incremental design / better performances

How ?

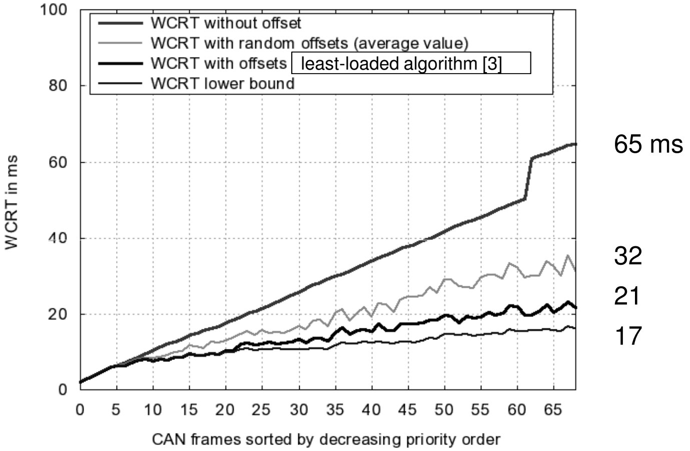
1. Keep amount of data transmitted minimum! → better identification and traceability of timing constraints
2. Synchronize producing tasks with communication tasks
3. Desynchronize frames by using offsets [3,4]
4. Assign priorities according to deadlines
5. Re-consider frame packing [12]
6. Optimize communication stacks so as to remove all “distortions” to the ideal CAN behavior

Scheduling frames with offsets ?!

Principle: desynchronize transmissions to avoid load peaks

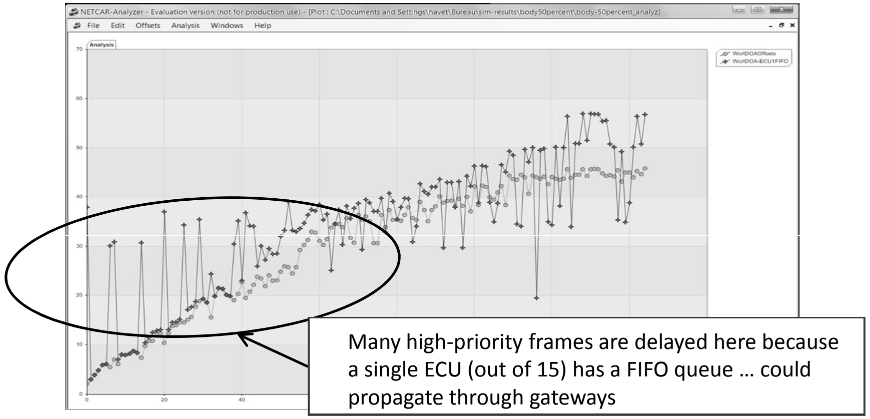


Offsets algorithm applied on a typical body network



Worst-case latencies on a 125 kbit/s body network

Let’s assume frame waiting queue is FIFO on ECU1, the OEM does not know it or software cannot handle it …

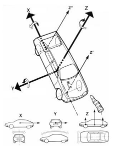

NETCAR-Analyzer - Evaluation version (not for production use) - [Plot : C:\Documents and Settings\havet\Bureau\sim-results\body50percent\body-50percent_analyz]
File
Edit
Offsets
Analysis
Windows
Help
Analysis
70
60
50
40
30
20
10
0
0
20
40
WcrtDOAOffsets
WcrtDOA-ECU1FIFO

Many high-priority frames are delayed here because a single ECU (out of 15) has a FIFO queue … could propagate through gateways

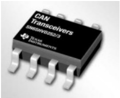
Up to recently [5,6], no response analysis on CAN was published …

Our work : bridging the gap between (analytic) models and reality

Higher load → less margin
→ **more accurate models**

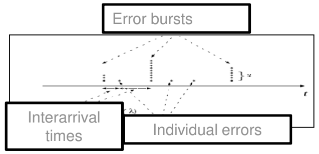


1 Hardware models

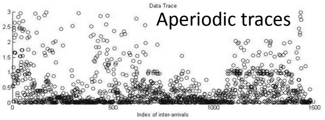


2 Software models (producer, sender, receiver, device drivers, etc)

3 Error models (reboot,errors)



4 Traffic models incl. aperiodic



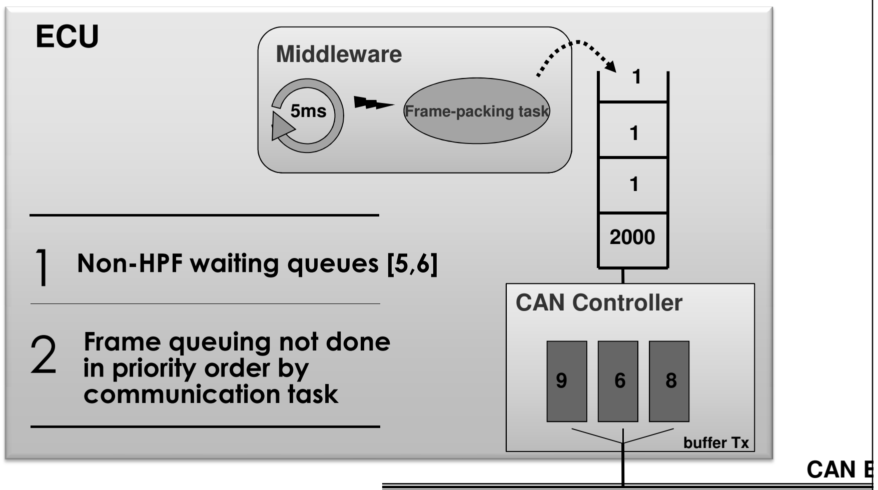
Aperiodic traces

3

Departure from the ideal CAN behavior

Some reasons

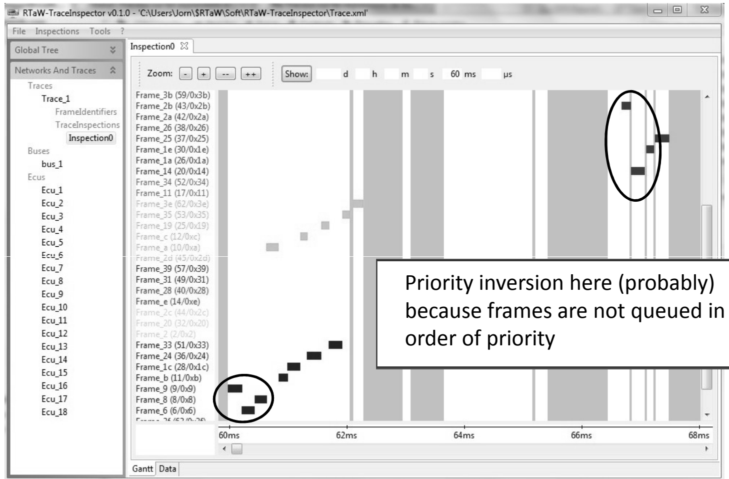
Departure from ideal CAN (1/2)



1 **Non-HPF waiting queues [5,6]**

2 **Frame queuing not done in priority order by communication task**

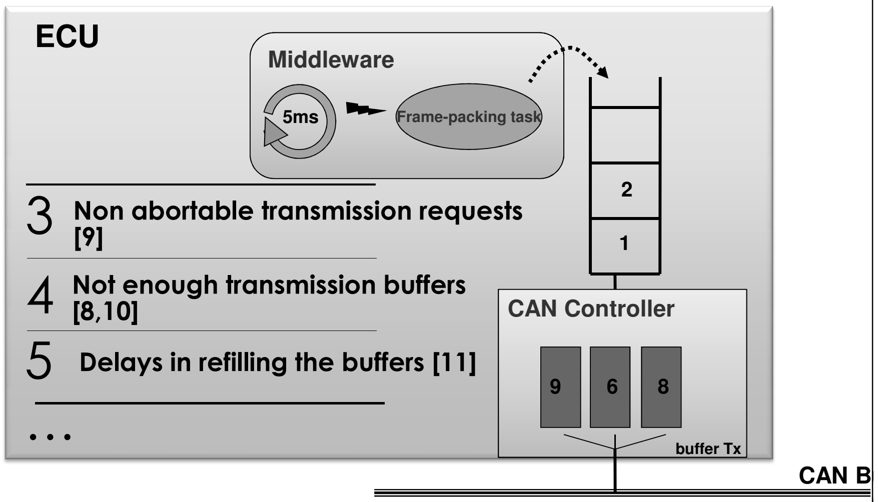
Analyzing communication traces : priority inversion



Priority inversion here (probably) because frames are not queued in the order of priority

RTaW-TraceInspector : check comm. stack implementation, periods, offsets, aperiodic traffic, clock drifts, etc ..

Departure from ideal CAN (2/2)



3 Non abortable transmission requests [9]

4 Not enough transmission buffers [8,10]

5 Delays in refilling the buffers [11]

…

Higher load level calls for

1. More constraining specifications / or conservative assumptions → a single node can jeopardize the system

2. Thorough use of Validation & Verification techniques:
   - simulation, analysis and trace inspection
   - none of them alone is sufficient !

Know-how, embedded software, verification techniques, and tool support have progressed to a point where **highly loaded CAN networks - yet safe** are possible

References

References

[1] N. Navet, Y-Q. Song, F. Simonot, “Worst-Case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network)“, Journal of Systems Architecture, Elsevier Science, vol. 46, n°7, 2000. Available at www.realtimeatwork.com

[2] N. Navet, B. Delord (PSA), M. Baumeister (Freescale), “Virtualization in Automotive Embedded Systems : an Outlook”, talk at RTS Embedded Systems 2010, Paris, France, March, 2010. Available at www.realtimeatwork.com

[3] M. Grenier, L. Havet, N. Navet, “Pushing the limits of CAN – Scheduling frames with offsets provides a major performance boost“, Proc. of the 4th European Congress Embedded Real Time Software (ERTS 2008), Toulouse, France, January 29 – February 1, 2008. Available at www.realtimeatwork.com

[4] P. Meumeu-Yomsi, D. Bertrand, N. Navet, R. Davis, "Controller Area Network (CAN): Response Time Analysis with Offsets", to appear in Proc. of the 9th IEEE International Workshop on Factory Communication System (WFCS 2012), May 21-24, 2012, Lemgo/Detmold, Germany.

[5] R.I. Davis, S. Kollmann, V. Pollex, F. Slomka, "Controller Area Network (CAN) Schedulability Analysis with FIFO queues”. In proceedings 23rd Euromicro Conference on Real-Time Systems (ECRTS), pages 45-56, July 2011.

[6] R. Davis, N. Navet, "Controller Area Network (CAN) Schedulability Analysis for Messages with Arbitrary Deadlines in FIFO and Work-Conserving Queues", to appear in Proc. of the 9th IEEE International Workshop on Factory Communication System (WFCS 2012), May 21-24, 2012, Lemgo/Detmold, Germany.

[7] R. Davis, A. Burn, R. Bril, and J. Lukkien, “Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised”, Real-Time Systems, vol. 35, pp. 239–272, 2007.

[8] M. D. Natale, “Evaluating message transmission times in Controller Area Networks without buffer preemption”, in 8th Brazilian Workshop on Real-Time Systems, 2006.

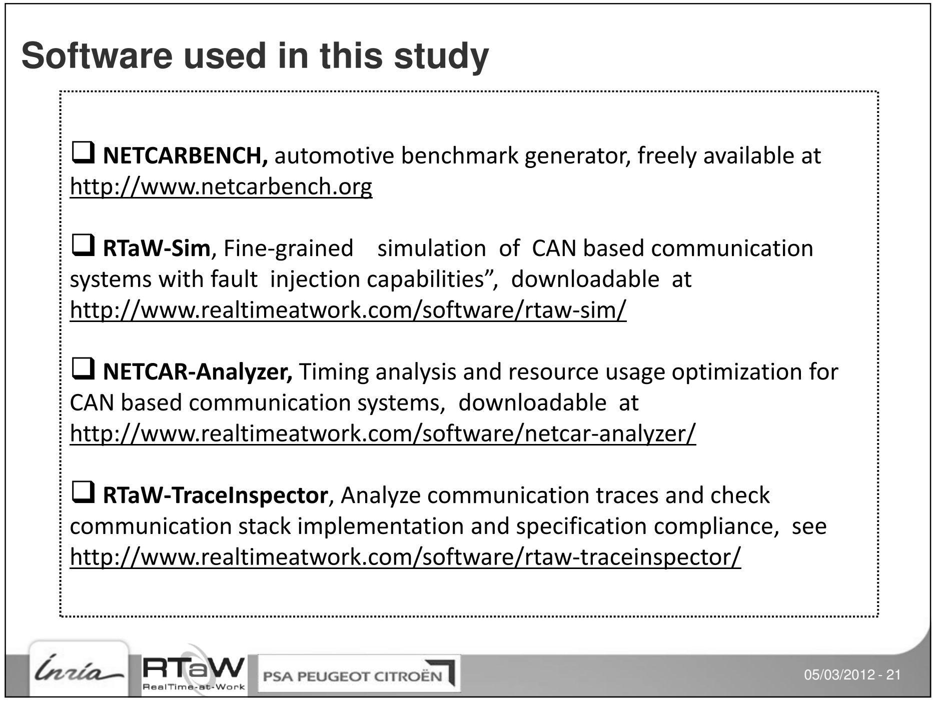
[9] D. Khan, R. Davis, N. Navet, “Schedulability analysis of CAN with non-abortable transmission requests“, 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France, September 2011. Available at www.realtimeatwork.com

[10] U. Keskin, R. Bril, and J. Lukkien, “Evaluating message transmission times in Controller Area Network (CAN) without buffer preemption revisited”, to appear in Proc. of the 9th IEEE International Workshop on Factory Communication System (WFCS 2012), May 21-24, 2012, Lemgo/Detmold, Germany.

[11] D. Khan, R. Bril, N. Navet, “Integrating Hardware Limitations in CAN Schedulability Analysis”, WiP at the 8th IEEE International Workshop on Factory Communication Systems (WFCS 2010), Nancy, France, May 2010. Available at www.realtimeatwork.com

[12] R. Saket, N. Navet, “Frame Packing Algorithms for Automotive Applications“, Journal of Embedded Computing, vol. 2, n° 1, pp93-102, 2006. Available at www.realtimeatwork.com

[13] N. Navet, H. Perrault, “CAN in Automotive Applications: a look forward”, 13th International CAN Conference, Hambach Castle, March 5-6, 2012. Available at www.realtimeatwork.com



Software used in this study

❑ **NETCARBENCH,** automotive benchmark generator, freely available at http://www.netcarbench.org

❑ **RTaW-Sim**, Fine-grained simulation of CAN based communication systems with fault injection capabilities”, downloadable at http://www.realtimeatwork.com/software/rtaw-sim/

❑ **NETCAR-Analyzer,** Timing analysis and resource usage optimization for CAN based communication systems, downloadable at http://www.realtimeatwork.com/software/netcar-analyzer/

❑ **RTaW-TraceInspector**, Analyze communication traces and check communication stack implementation and specification compliance, see http://www.realtimeatwork.com/software/rtaw-traceinspector/