Controller Area Network (CAN): Response Time Analysis with Offsets

Patrick Meumeu Yomsi, Dominique Bertrand,
Nicolas Navet, Robert I. Davis

IEEE WFCS 2012

May 22, 2012

1 Outline

- Design of (automotive) CAN networks today
- Worst-case response time analysis with offsets
- Performance evaluation

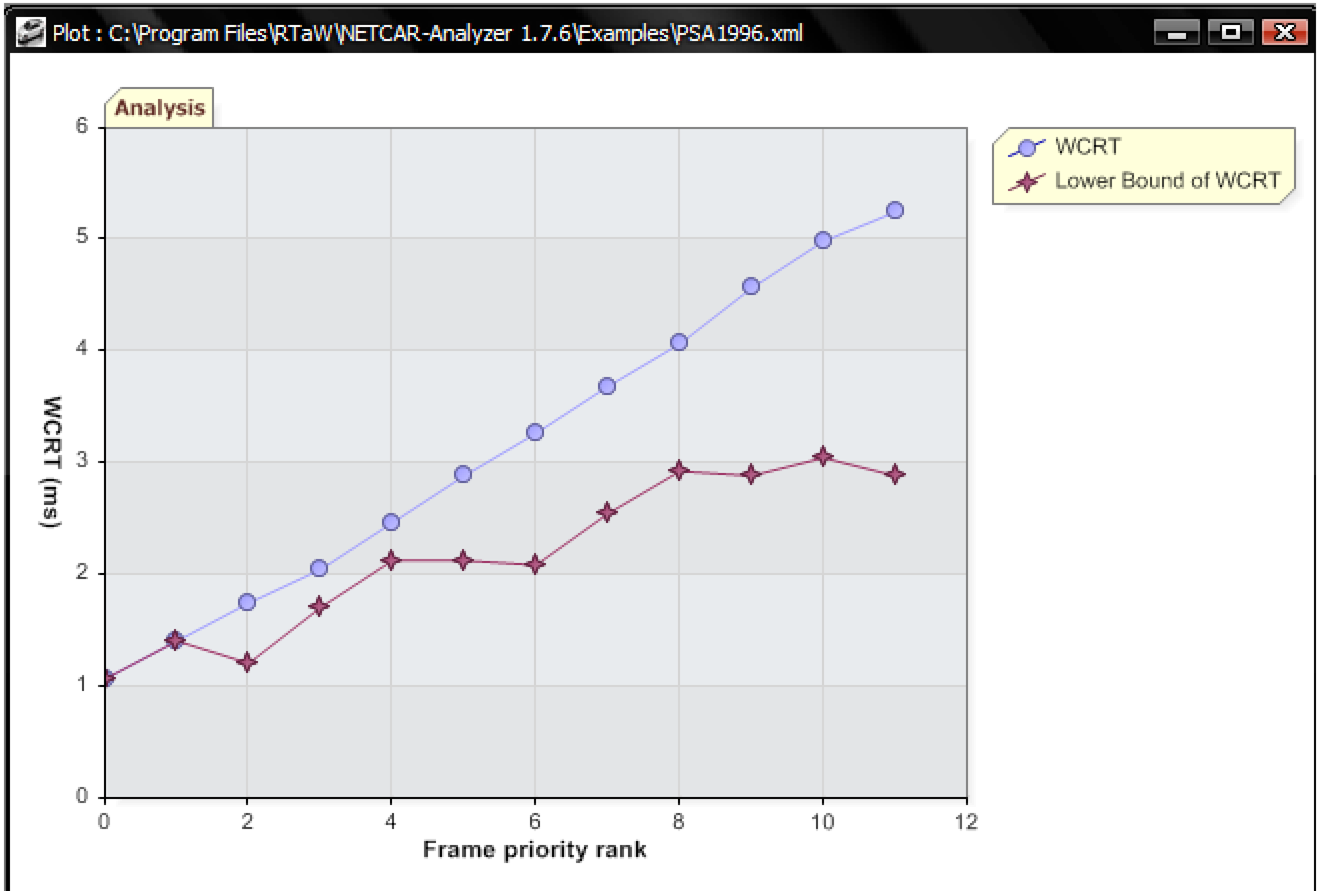
Automotive CAN: the early days (1/2)

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

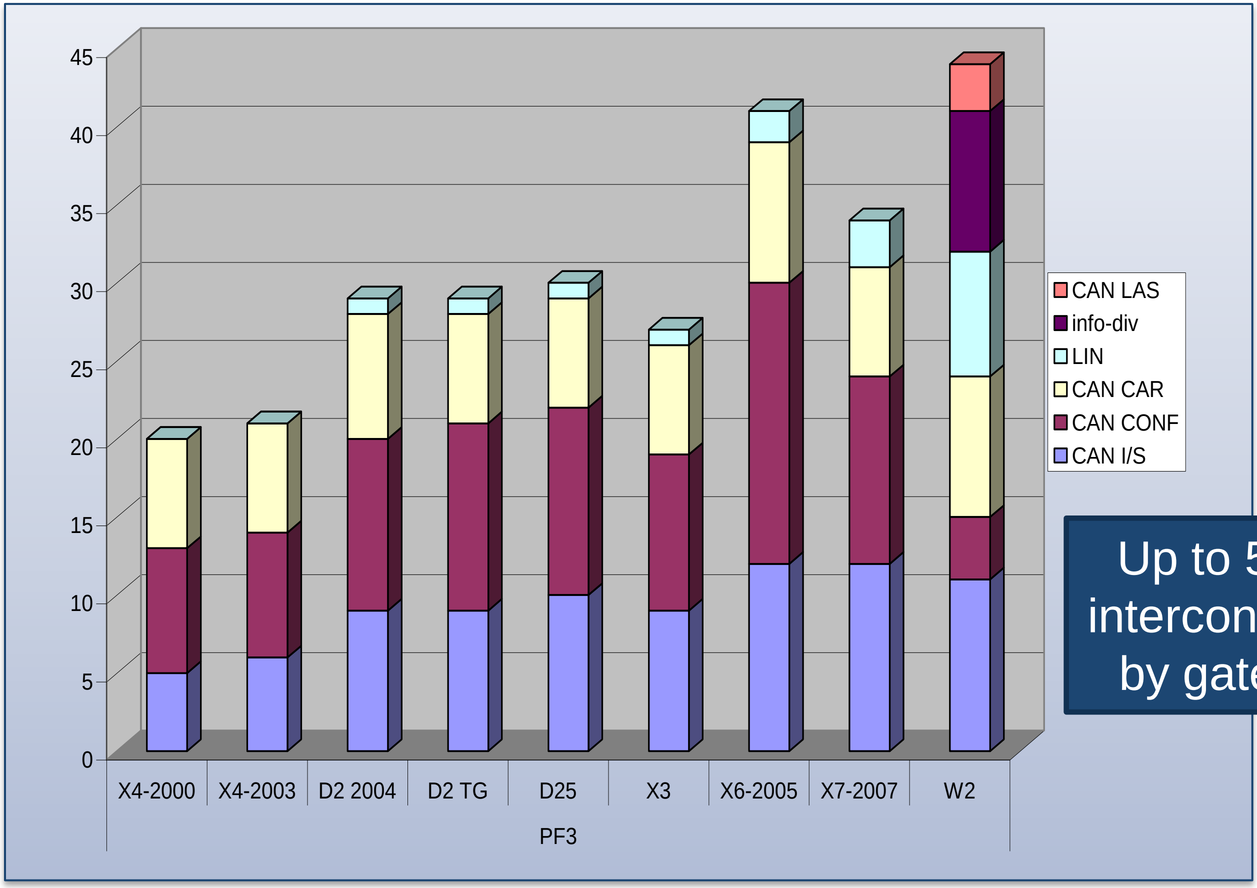
Early CAN project at PSA (1996, see [1])
250kbit/s

Automotive CAN: the early days (2/2)



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

“easy” integration for the OEM till 35-40% - precise performance evaluation needed beyond [4]

- **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.
- **Complex traffic model** : aperiodic (w/wo exclusion time), Autosar mixed transmission mode, segmented messages, download session, etc …

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

Higher load level calls for

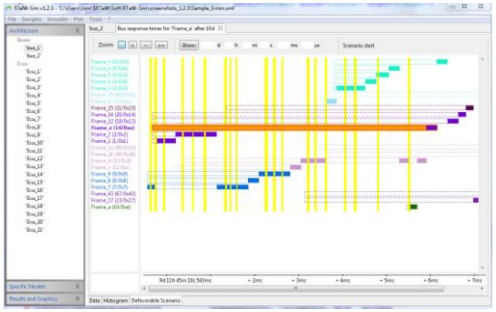
1. More constraining specifications to the suppliers / or conservative assumptions → a single node can jeopardize the whole system

2. Thorough use of Validation & Verification techniques:
   - **simulation, worst-case 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

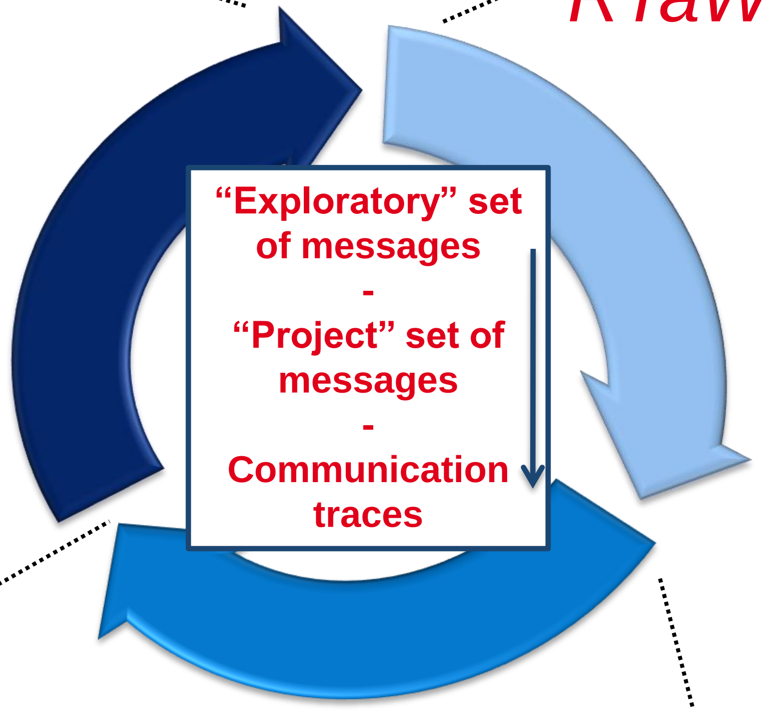
Tools & techniques complementarities the case at RTaW [4]

RTaW-Sim

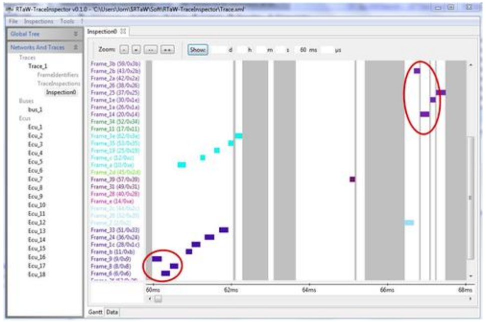


- ✓ Simulation
- ✓ fault-injection

RTaW-Sim & Netcar-Analyzer freely downloadable

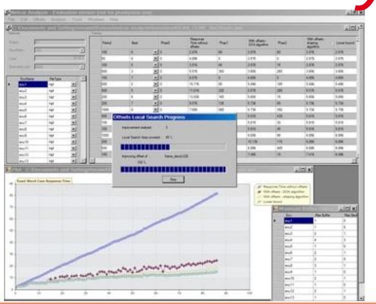


RTaW-TraceInspector



Trace analysis for :
- ✓ Error model
- ✓ Aperiodic traffic model
- ✓ Real periods, offsets, clock drifts, functioning modes, bit-stuffing, etc
- ✓ Communication stack quality
- ✓ …

NETCAR-Analyzer



- ✓ Worst-case analysis
- ✓ Offset optimization

Different sets of messages along the development process : our view

"Exploratory" sets of message

- ✓ Virtual sets of messages generated from real sets of messages
- ✓ Architecture design
- ✓ Technological choices
- ✓ "Coarse grained" verification
- ✓ Incremental design possibility
- ✓GPL tool Netcarbench

"Project" sets of messages

- ✓ Configuration: offsets, ID, etc,
- ✓ "Fine grained" verification
- ✓ Evolutions: adding frames, ECUs

Communication traces

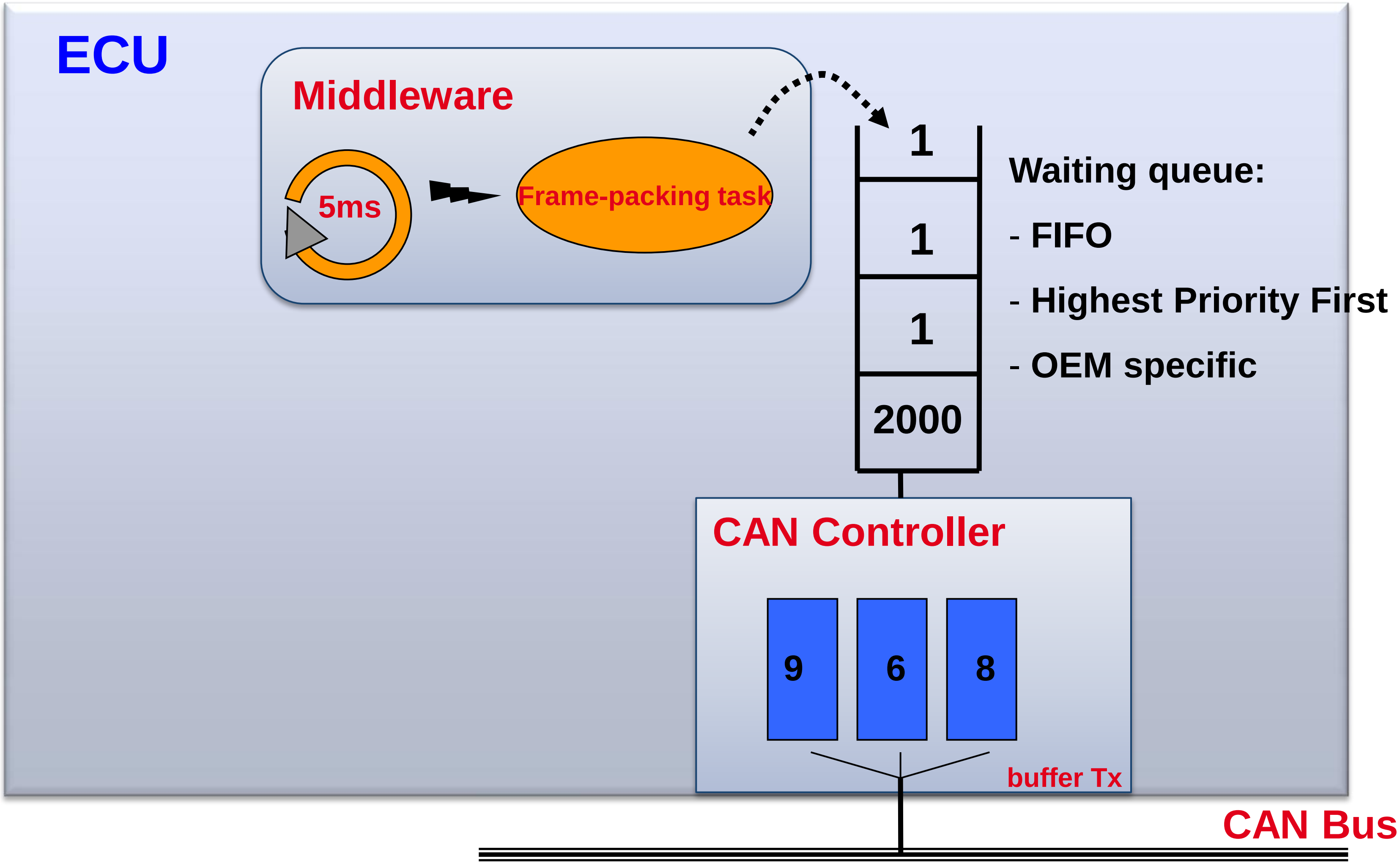
- ✓ Verifying specification respect
- ✓ Impact of non-conformance

2

Optimizing CAN networks

What levers do we have ?

Automotive CAN communication stack : a simplified view



Optimizing CAN : meeting performance and robustness constraints at higher load

An industrial requirement

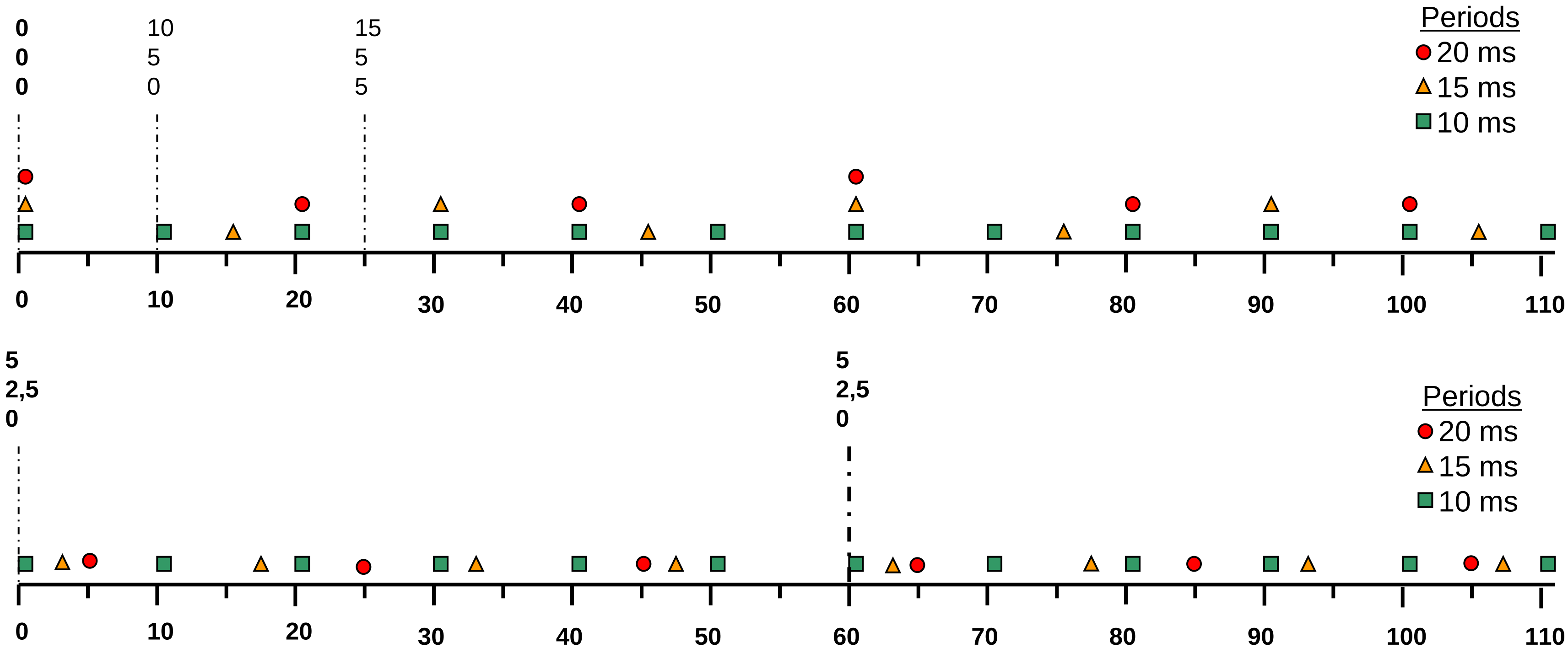
- Reduce architecture complexity, HW costs & weight, consumption and emission
- Avoid industrial risks and costs of new technologies
- Incremental design / better performances

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 "departure" from the ideal CAN behavior**

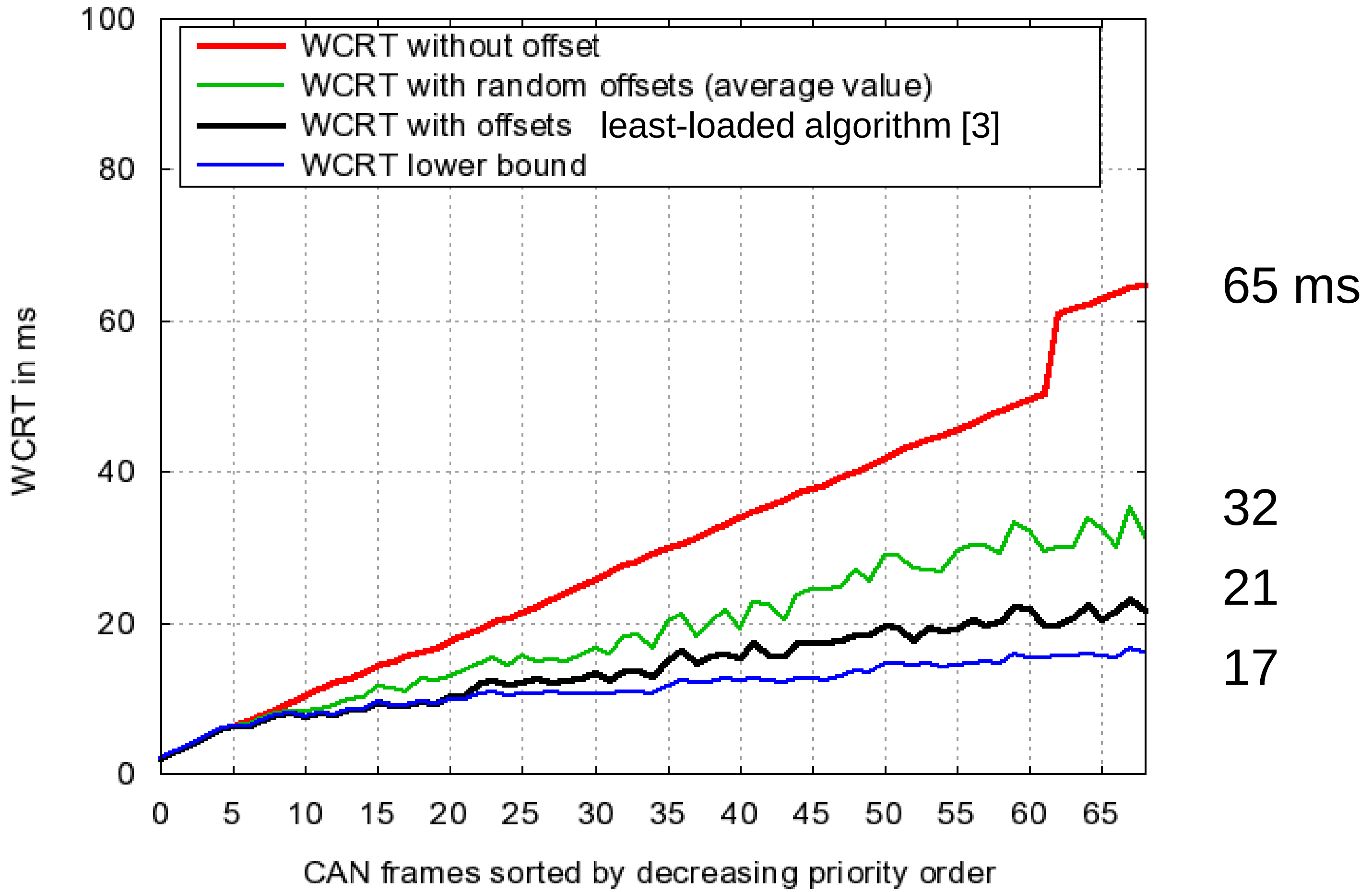
Scheduling frames with offsets ?!

Principle: desynchronize transmissions to avoid load peaks



Offsets algorithms: DOA [14], least-loaded [3], SOA [RTaW], local optimization (GA), etc..

Offsets algorithm applied on a typical body network [3]



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

Analyses for safety critical systems : simple, peer-reviewed and documented

- ✓ Flawed analyses are dangerous in safety critical systems but (fine-grained) analyses are complex and error prone. Remember "CAN analysis refuted, revisited, etc" [6] ?!
- ✓ Implemented analysis have to make simplifications esp. in a heterogeneous systems (and tools do not document that well)
- ✓ Solutions ?
  - peer-review of the WCRT analyses is needed
  - coarse-grained / conservative but simple as far as possible : e.g., [5,6] vs [9]
  - no black-box software: documentation of implemented analyses and underlying hypotheses
  - cross-validation between tools on benchmarks

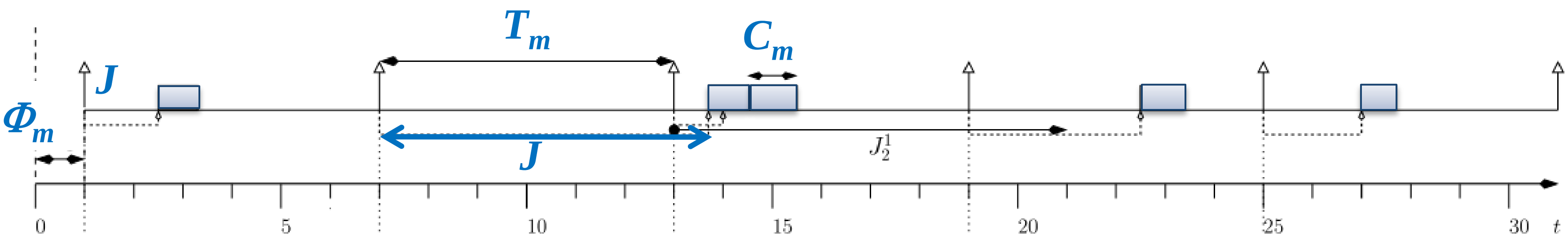
3

Response time analysis with offsets

Contribution : exact response time analysis with offsets

- ✓ Adaptation of Palencia & Harbour work to non-preemptive scheduling [15]
- ✓ Periodic and sporadic frames / with and without jitters / arbitrary jitters and deadlines
- ✓ Complexity is exponential but usable for medium-size systems with typical automotive characteristic (e.g., non arbitrary periods)
- ✓ Performance evaluation with jitters shows that offsets bring major performance boost
- ✓ Sound basis for optimization and non-ideal CAN behavior

System model



- ✓ Stations are not synchronized (no global clock)
- ✓ Ideal CAN nodes
- ✓ Each CAN message has a:
  - Unique sending node
  - Unique priority m
  - Maximum transmission time C_m
  - Minimum inter-arrival time or period T_m
  - An offset Φ_m : first activation time after origin (multiple of the communication task period if any)
  - Arbitrary deadline D_m
  - Arbitrary max. queuing jitter J_m

WCRT analysis with offsets: principles

1 On each station, model the outgoing traffic as the minimum number of transactions

2 Identify the smallest set of scenarios that must contain to the worst-case response time for a specific CAN ID

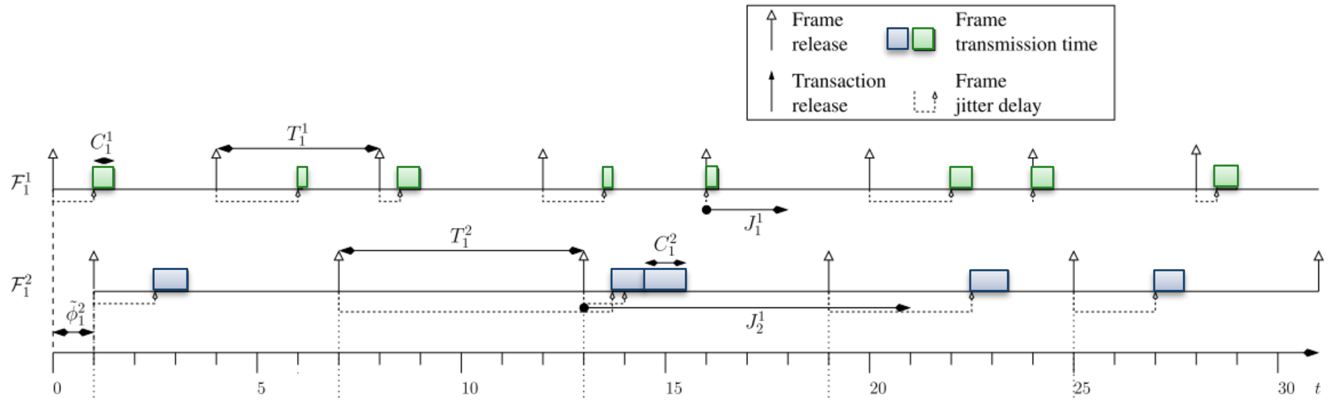
3 Compute the max. response time for the current ID on each identified scenario

Step 1: from frames to transactions

A transaction captures all the periodic traffic sent by a node

$T_1 = 4, J_1 = 2, \Phi_1 = 0$

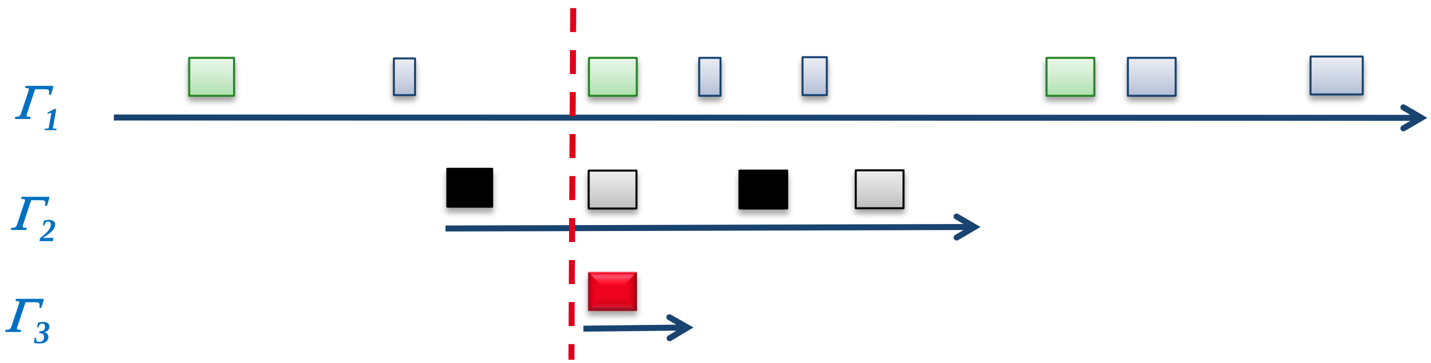
$T_2 = 6, J_2 = 7, \Phi_1 = 1$



All periodic frames a of node forms a single transaction, each sporadic frame needs its own transaction

Step 2: set of scenarios to examine

What is the worst scenario for of priority *p* ?



Theorem (adapted from [15]): the worst-case scenario for the frame of priority *p* belongs to the set of scenarios in which one frame with a priority higher than or equal to *p* in each transaction is released simultaneously after having experienced its maximum jitter

Simple optimization : reconstruct smaller transactions that only include frames with priority higher than or equal than *p*

Step 3: response time in a specific scenario – source of interferences

1 Lower priority frame that has started transmission

2 Instances of the same priority

3 Higher priority frames : due to jitters, they might be several instances released simultaneously at the start of the busy period tc

$$K_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\text{max number of instances that can accumulate at } tc} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor + 1}_{\text{max number of instances arriving after } tc}$$

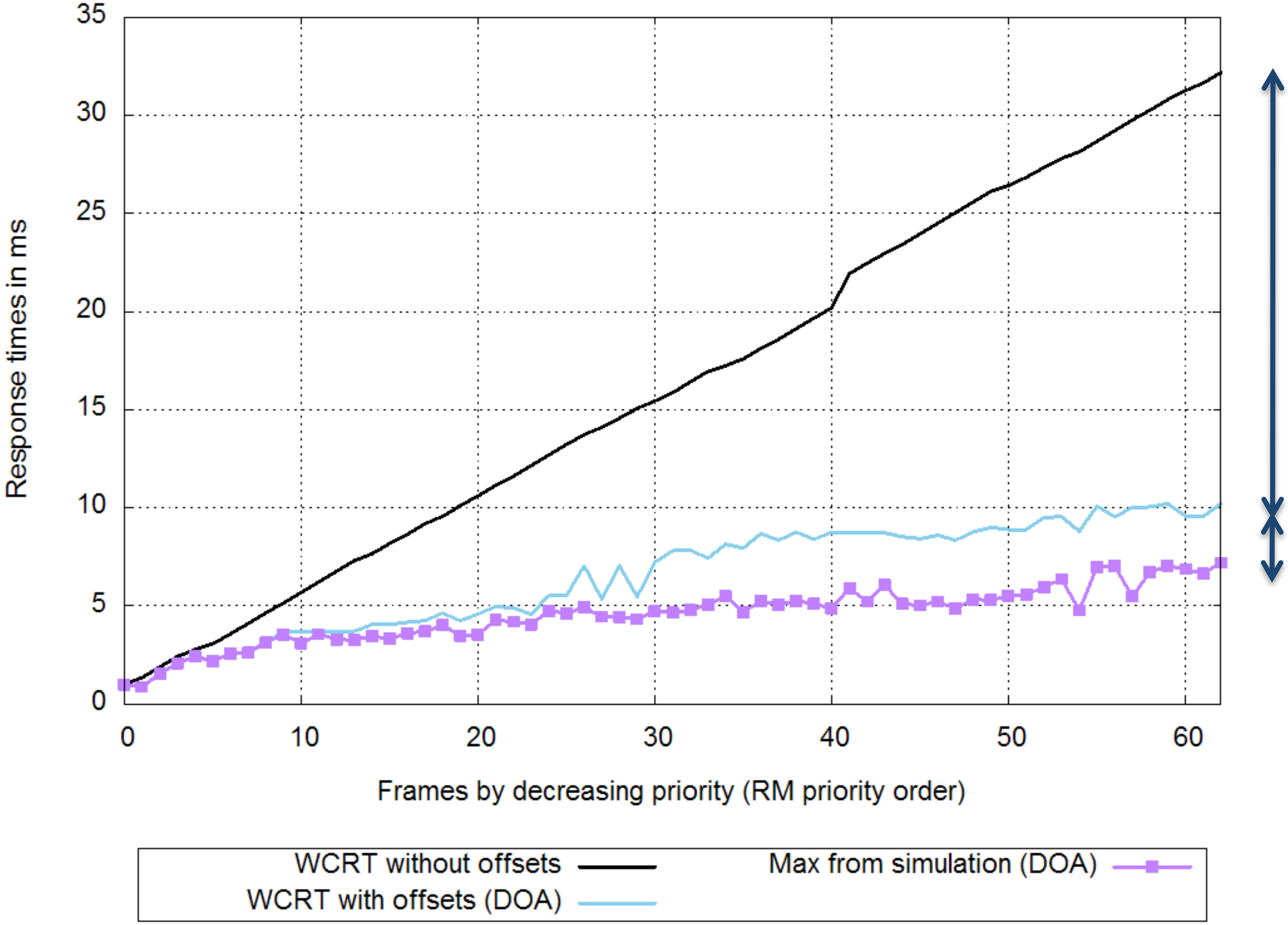
4

Experimental results

Setup : medium-size automotive body networks

- Generated using NETCARBENCH
- 8-12 ECUs - 250Kbit/s - load from 38 to 42%
- One station transmits 20% of the traffic
- Message periods : 20, 50, 100, 200 or 500 ms
- Deadlines equal to periods
- Queuing jitter: with (below period for 10% of the frames) and without
- Offset algorithm: DOA [14]
- Priorities: transmission deadline monotonic order (TDMPO) i.e. D-J order

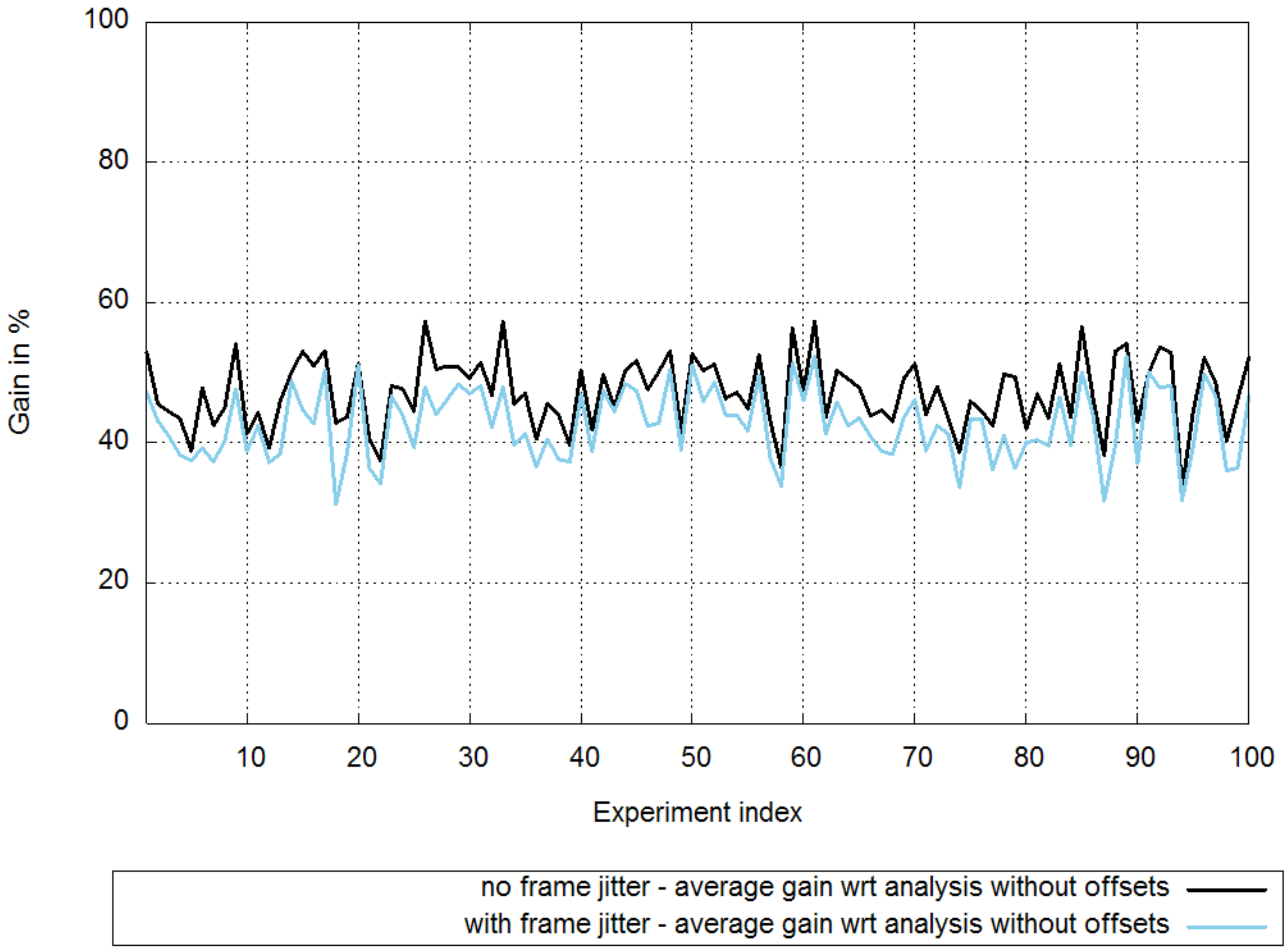
Focus on a single (typical) configuration – WCRT and max from simulation



Gain with offsets: 68% for the lowest priority

Difference WCRT vs max on random simulations: avg: 25% max: 45%

Results over 100 random configurations – average WCRT gain over all frames

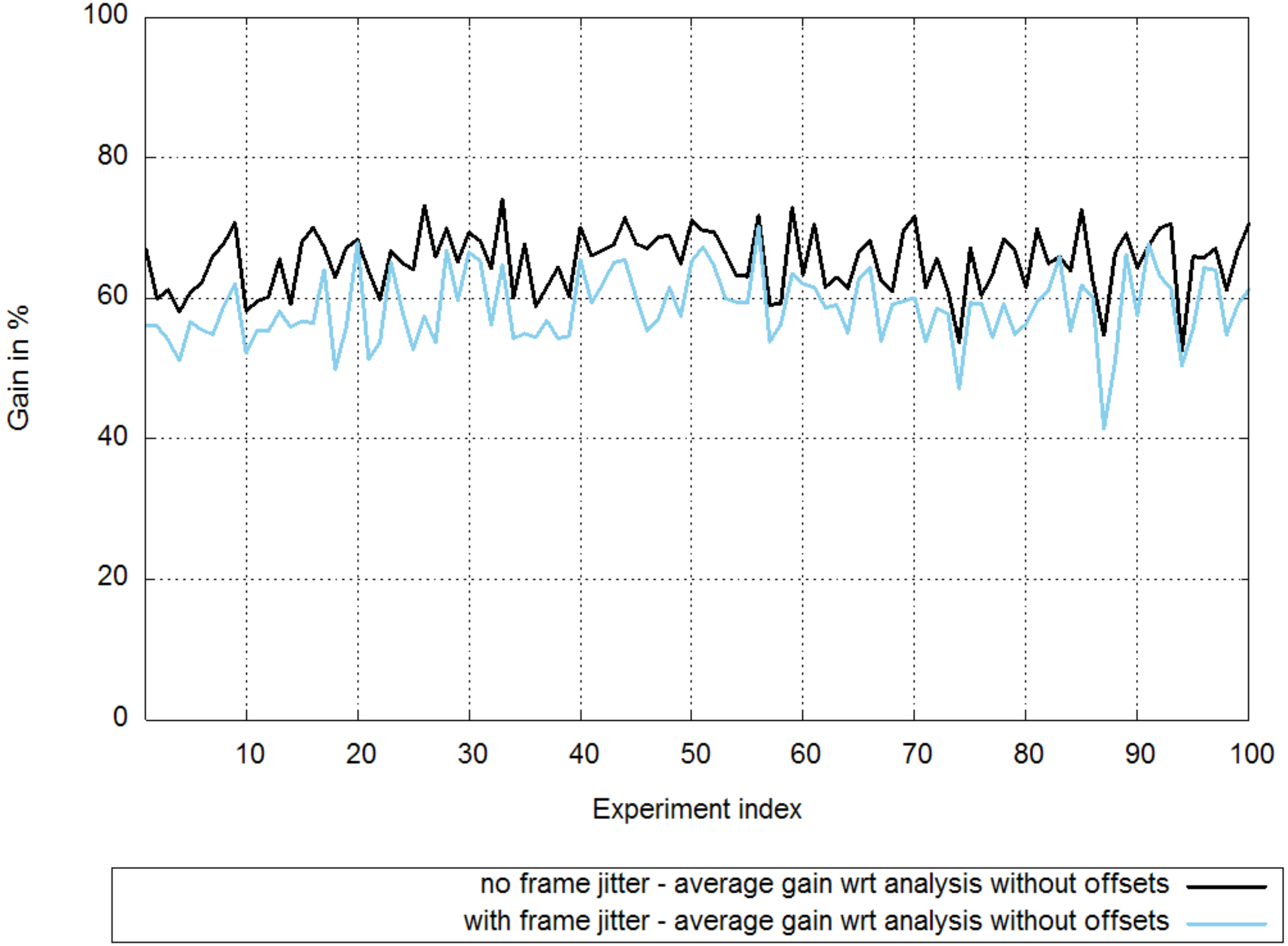


Gain with offsets

without jitter
Avg: 47%
Max: 57%

with jitter (10%)
Avg: 42%
Max: 52%

Results over 100 random configurations – average WCRT gain over the 20% lowest priority frames



Gain with offsets

without jitter
Avg: 65%
Max: 74%

with jitter (10%)
Avg: 59%
Max: 70%

Future work

- [ongoing] Optimization so as to make exact analysis usable on arbitrary large CAN networks
- [ongoing] Extension to heterogeneous networks with non-ideal CAN behavior
- Extension to segmented messages

References

References (1/2)

[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] N. Navet, H. Perrault (PSA), “CAN in Automotive Applications: a Look Forward“, 13th International CAN Conference, Hambach, Germany, March 5-6, 2012. Available at www.realtimeatwork.com

[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", 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

References (2/2)

[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] 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.

[14] M. Grenier, J. Goossens, N. Navet, "Near-Optimal Fixed Priority Preemptive Scheduling of Offset Free Systems", Proc. of the 14th International Conference on Network and Systems (RTNS'2006), Poitiers, France, May 30-31, 2006.

[15] J. Palencia and M. G. Harbour, “Schedulability analysis for tasks with static and dynamic offsets”, in Proc. the 19th IEEE Real-Time Systems Symposium, RTSS’1998, dec 1998, pp. 26–37, Madrid, Spain.