Pushing the limits of CAN - Scheduling frames with offsets provides a major performance boost

Nicolas NAVET

INRIA / RealTime-at-Work

http://www.loria.fr/~nnavet

http://www.realtimeatwork.com

Nicolas.Navet@realtimeatwork.com

ERTS – 30/01/2008

Joint work with Mathieu GRENIER and Lionel HAVET

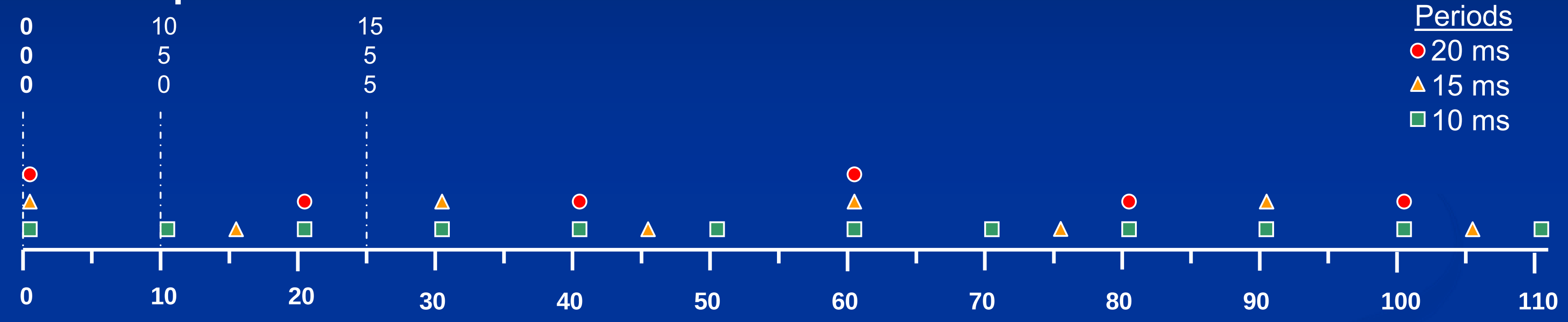
In-vehicle networking : will CAN be able to keep up the pace?

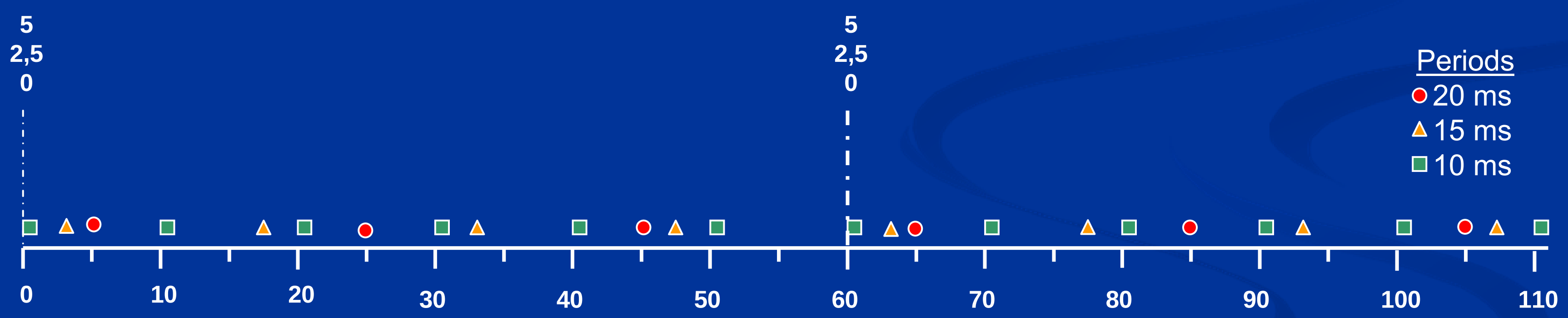
- Typically max. bus load is set to 35%
- Not enough wrt to short/medium term bandwidth needs …
  - Solution 1: multiple CAN networks … but gateways induce heavy overhead
  - Solution 2: switch to FlexRay … expensive for bandwidth alone
  - Solution 3: optimize the scheduling of CAN frame .. Offsets provide a solution to make CAN predictable at higher network load (≥60%)

RTaW RealTime-at-Work INRIA

Scheduling frames with offsets ?!

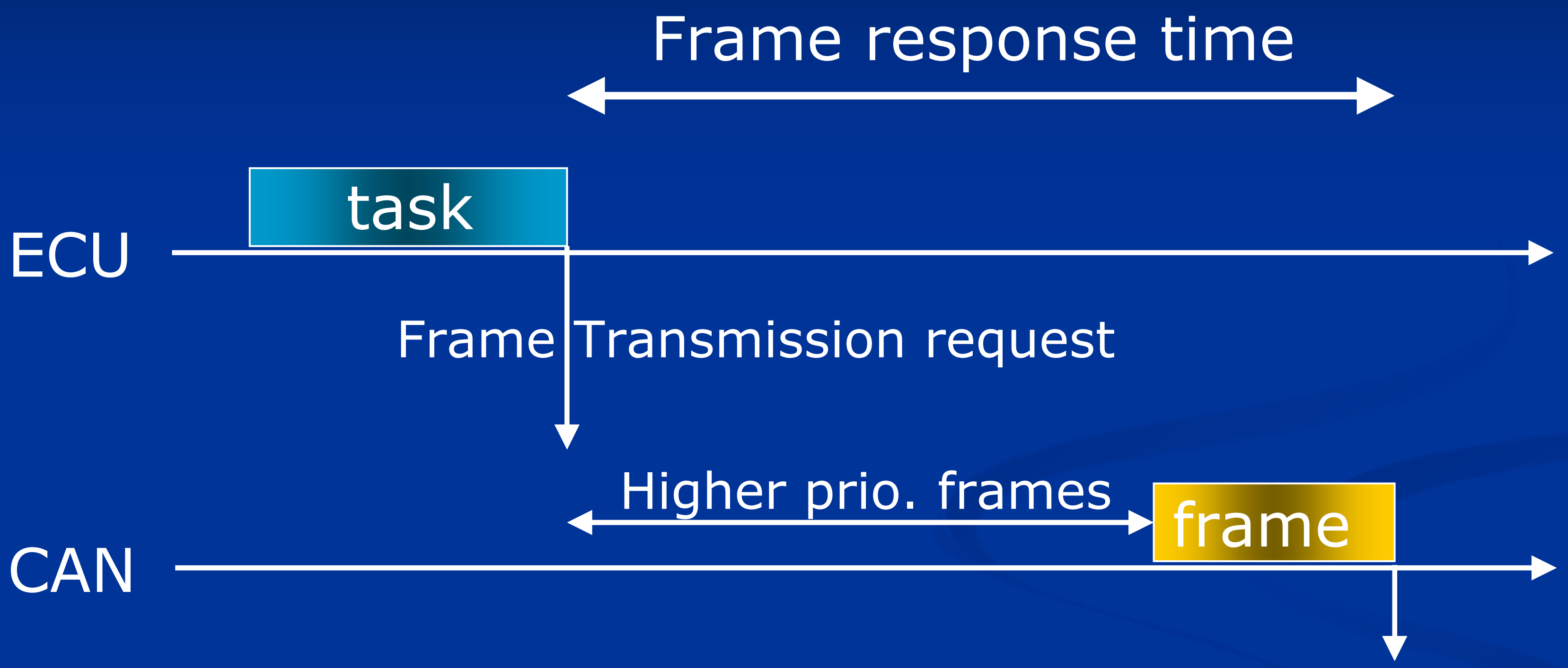
Principle: desynchronize transmissions to avoid load peaks





Algorithms to decide offsets are based on arithmetical properties of the periods and size of the frame

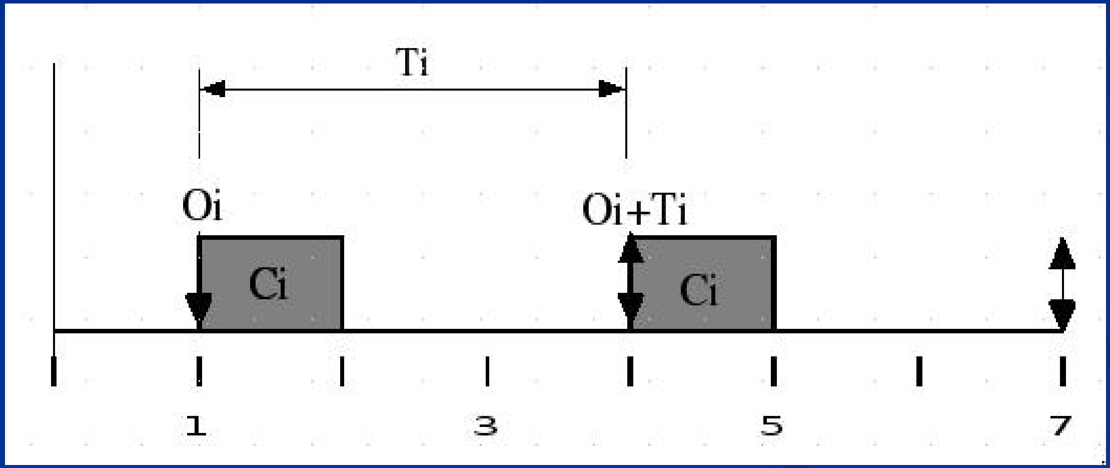
System model (1/2)



- Performance metric: worst-case response time

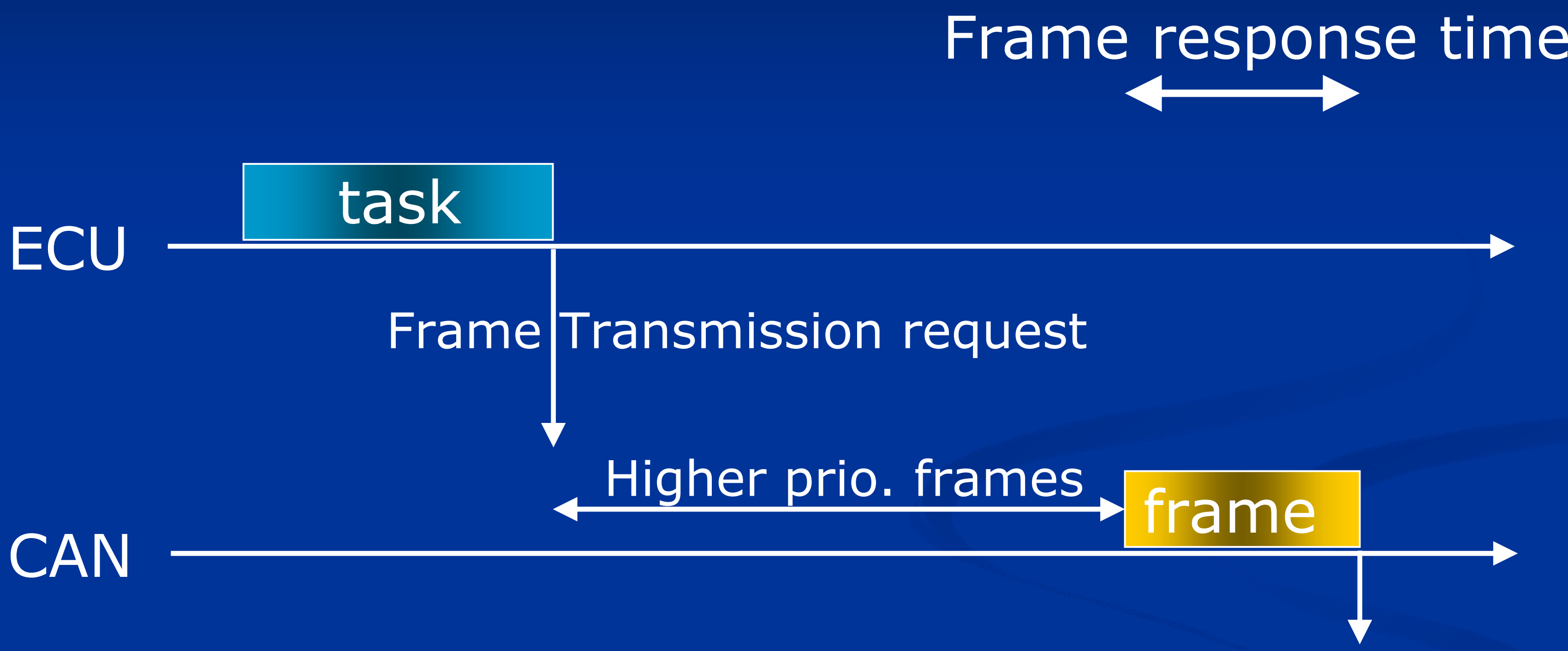
System model (2/2)

- The offset of a message stream is the time at which the transmission request of the first frame is issued



- Complexity: best choosing the offsets is exponential in the task periods → approximate solutions
- Middleware task imposes a certain granularity
- Without ECU synchronisation, offsets are local to ECUs

But task scheduling has to be adapted…



- In addition, avoiding consecutive frame constructions on an ECU allows to reduce latency

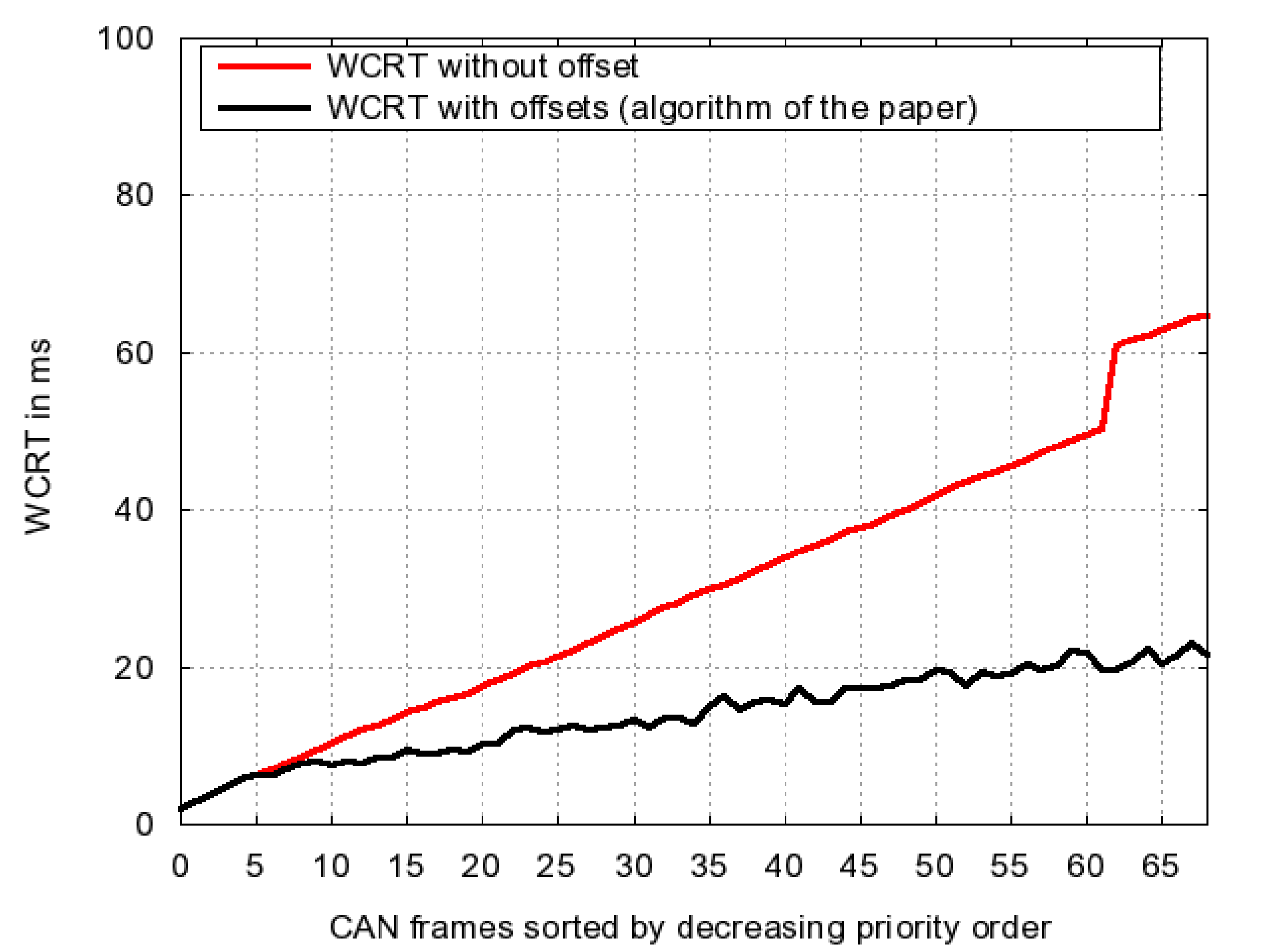
RTaW RealTime-at-Work INRIA

Offsets Algorithm (1/3)

- Ideas:
  - assign offsets in the order of the transmission frequencies
  - release of the first frame is as far as possible from adjacent frames
  - identify “least loaded interval”
- Ex: $f_1=(T_1=10)$, $f_2=(T_2=20)$, $f_3(T_3=20)$

Time	0	2	4	6	8	10	12	14	16	18
Frame			$f_{1,1}$		$f_{2,1}$			$f_{1,2}$		$f_{3,1}$

Offsets Algorithm applied on a typical body network



65 ms

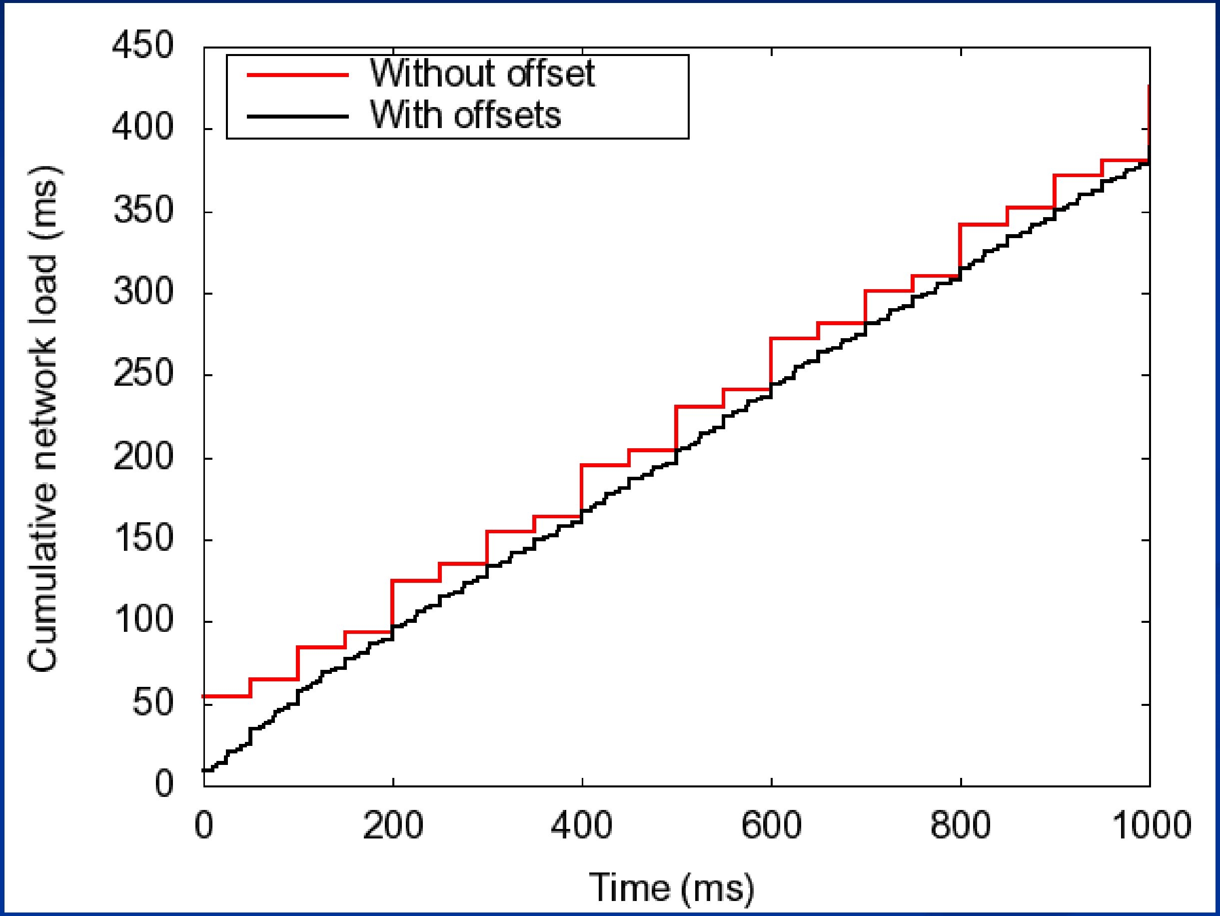
21 ms

Offsets Algorithm (3/3)

- Low complexity and efficient as is but further improvements possible:
  - add frame(s) / ECU(s) to an existing design
  - user defined criteria : optimize last 10 frames, a specific frame,
  - take into account priorities
  - optimization algorithms: tabu search, hill climbing, genetic algorithms
  - …

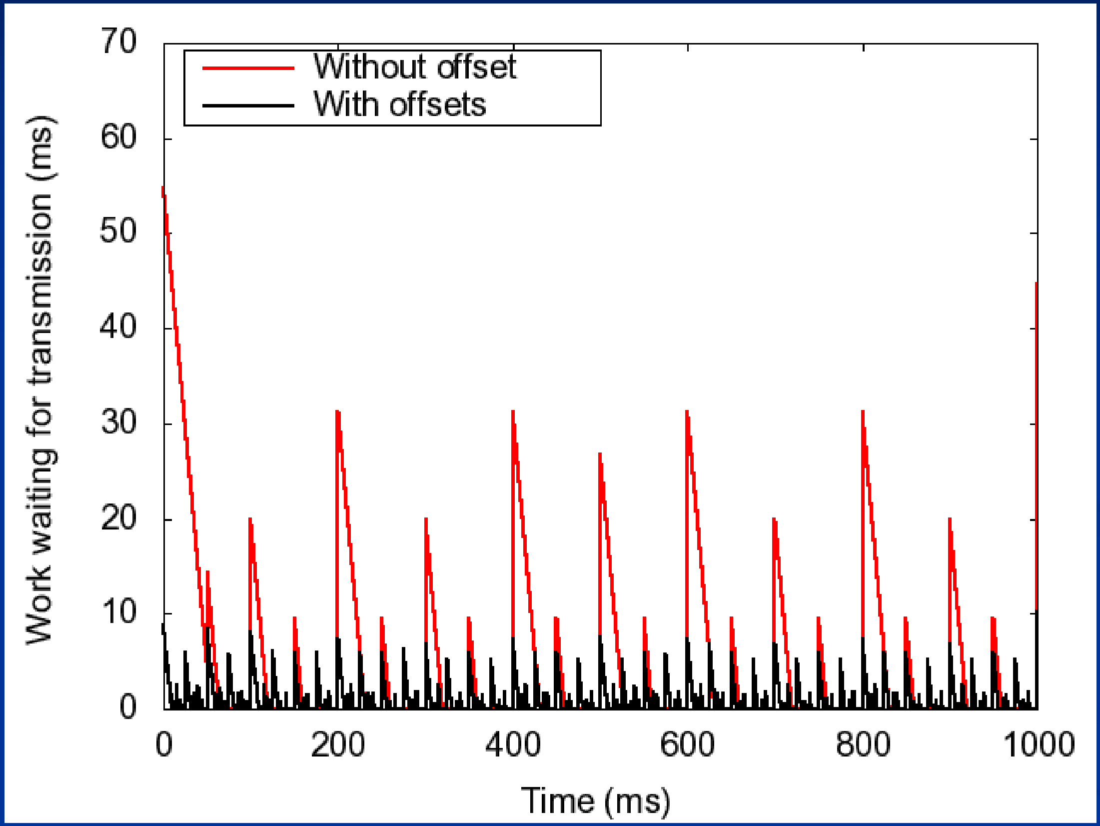
Efficiency of offsets : some insight (1/2)

Work = time to transmit the CAN frames sent by the stations



- Almost a straight line, suggests that our algorithm is near-optimal

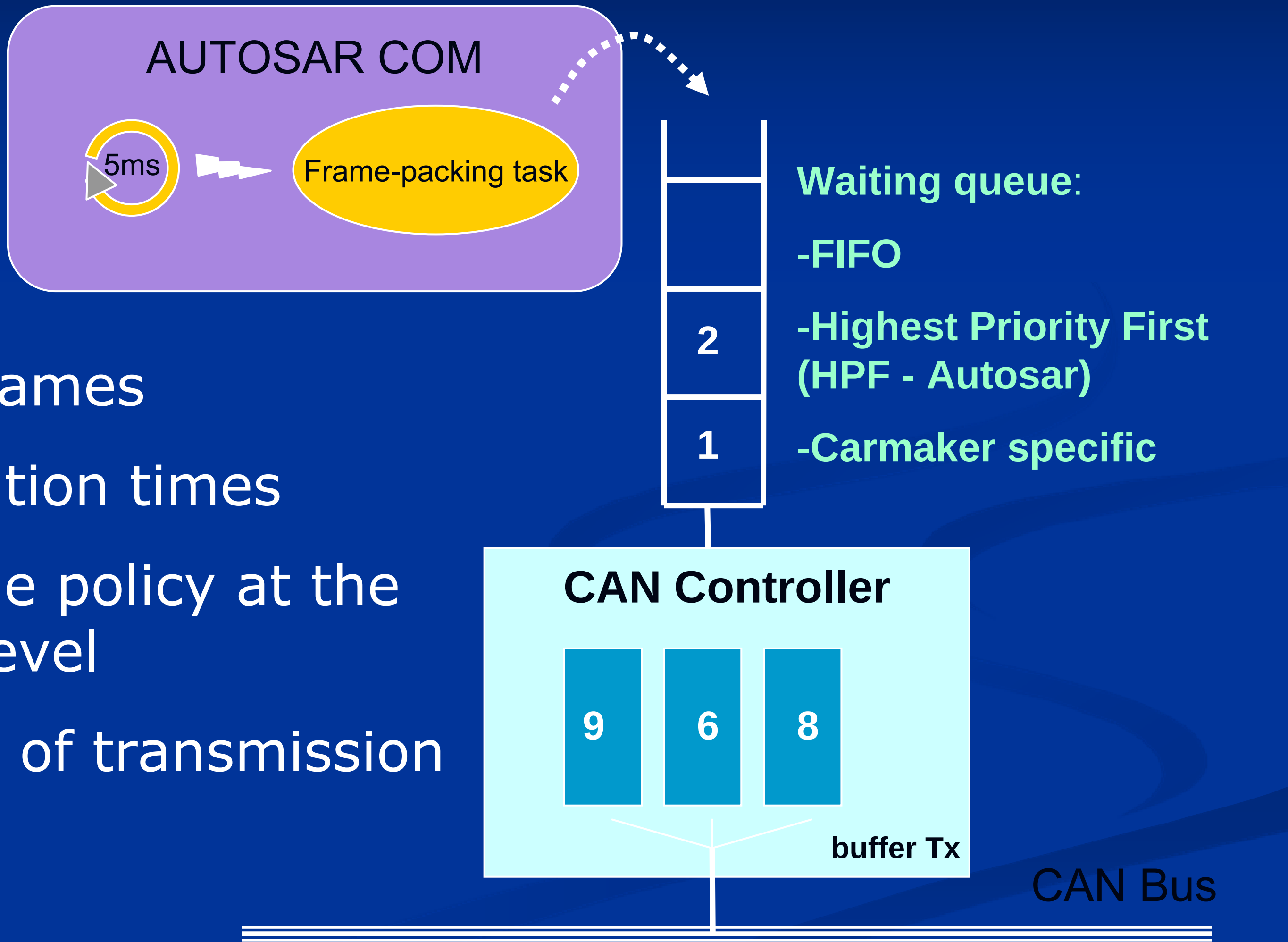
Efficiency of offsets : some insight (2/2)



- A larger workload waiting for transmission implies larger response times for the low priority frames ..

Computing worst-case response times with offsets

Computing frame worst-case response time with offsets



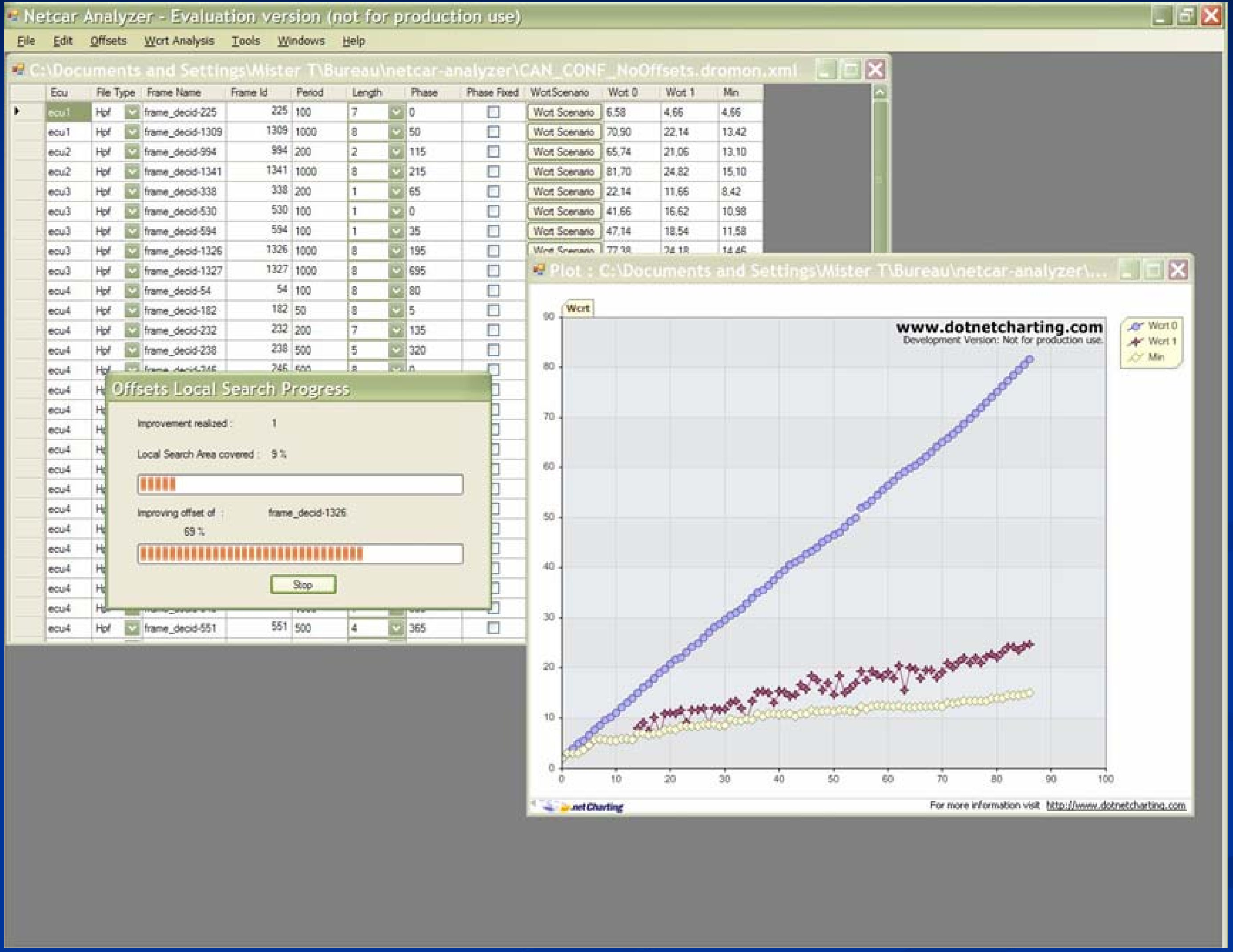
Requirements :

- handle 100+ frames
- very fast execution times
- ≠ waiting queue policy at the microcontroller level
- limited number of transmission buffers

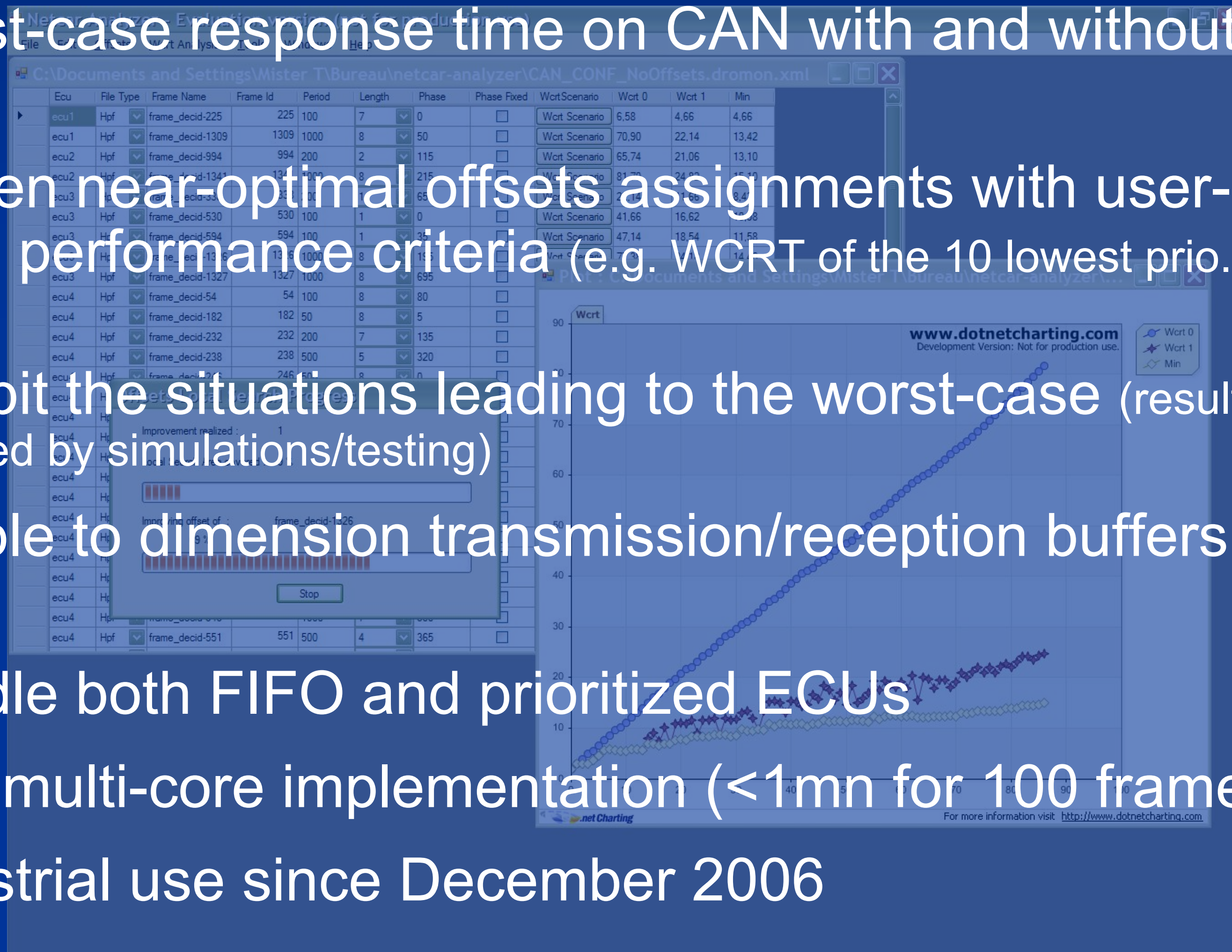
WCRT : State of the art

- Scientific literature:
  - Complexity is exponential
  - No schedulability analysis with offsets in the distributed non-preemptive case
  - Offsets in the preemptive case : not suited for > 10-20 tasks
  - WCRT without offsets: infinite number of Tx buffers and no queue at the microcontroller level
- Our software: NETCAR-Analyzer

NETCAR-Analyzer : developed at INRIA, then RealTime-at-Work



NETCAR-Analyzer : an overview



- ✓ Worst-case response time on CAN with and without offsets
- ✓ Proven near-optimal offsets assignments with user-defined performance criteria (e.g. WCRT of the 10 lowest prio. frames)
- ✓ Exhibit the situations leading to the worst-case (results can be checked by simulations/testing)
- ✓ Enable to dimension transmission/reception buffers (RAM)
- ✓ Handle both FIFO and prioritized ECUs
- ✓ Fast multi-core implementation (<1mn for 100 frames)
- ✓ Industrial use since December 2006

Performance evaluation :

- Experimental Setup
- WCRT of the frames wrt random offsets and lower bound
- WCRT reduction ratio for chassis and body networks
- Load increase : add new ECUs / add more traffic

Experimental Setup

- Body and chassis networks

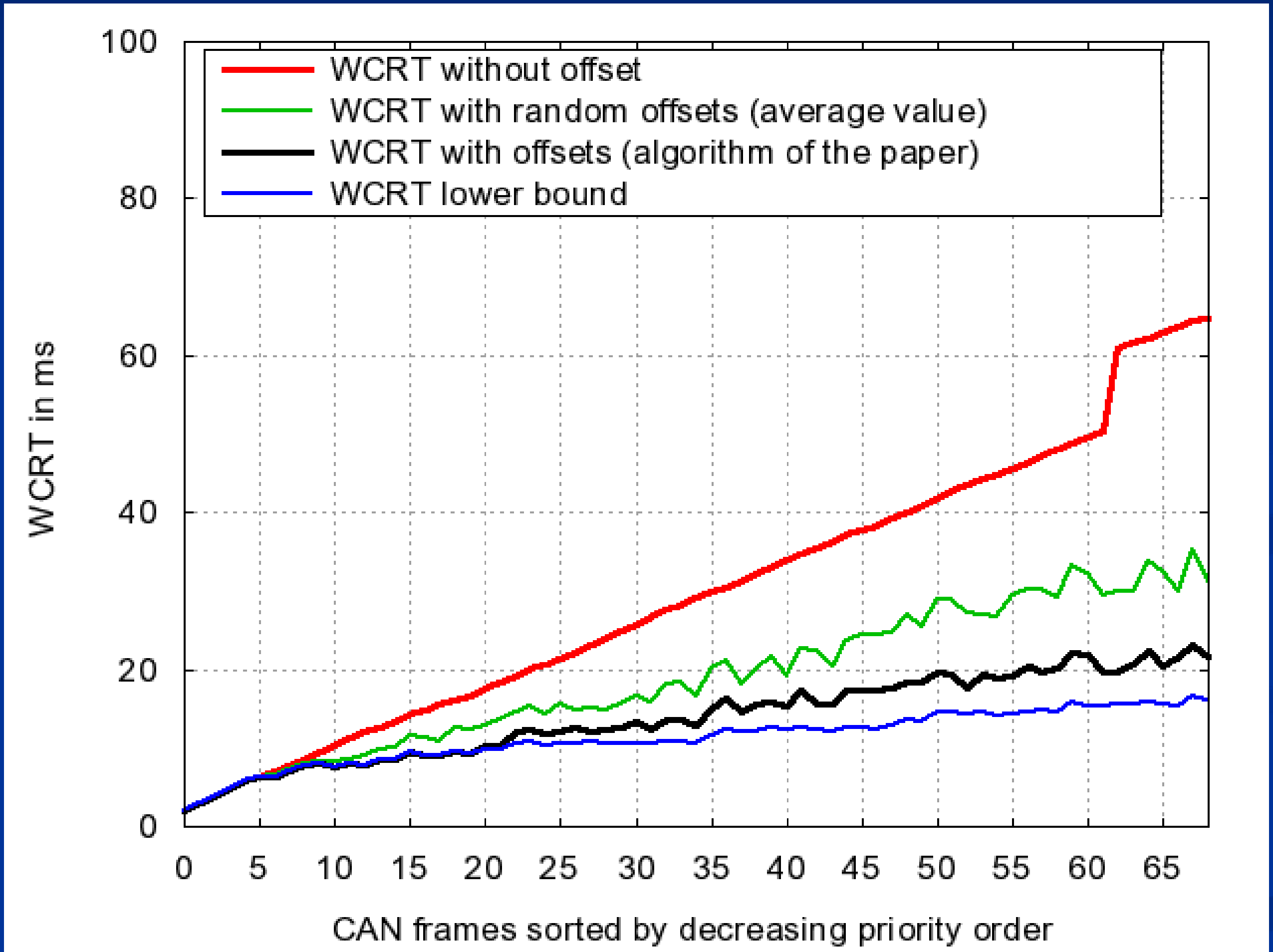
Network	#ECUs	#Messages	Bandwidth	Frame periods
Body	15-20	≈ 70	125Kbit/s	50ms-2s
Chassis	5-15	≈ 60	500Kbit/s	10ms-1s

With / without load concentration: one ECU generates 30% of the load

- Set of frames generated with NETCARBENCH (GPL-licenced)

RTaW RealTime-at-Work INRIA

Offsets in practice : large response time improvements (1/2)



65 ms

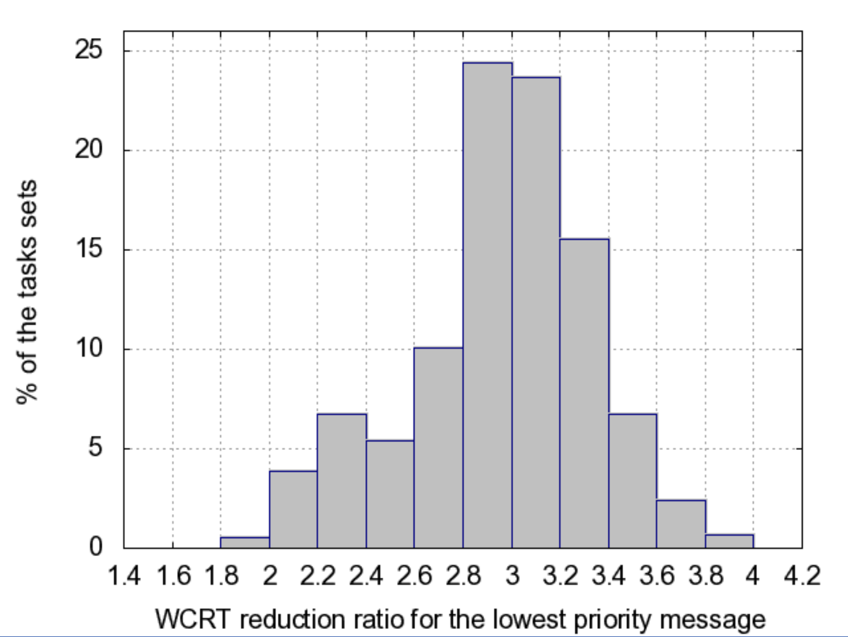
32

21

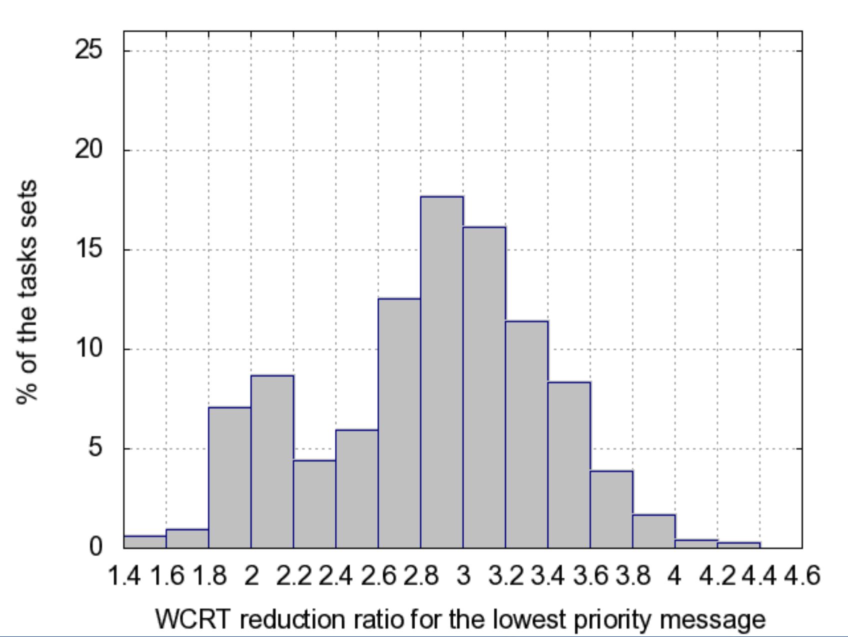
17

WCRT Reduction Ratio

Body Networks



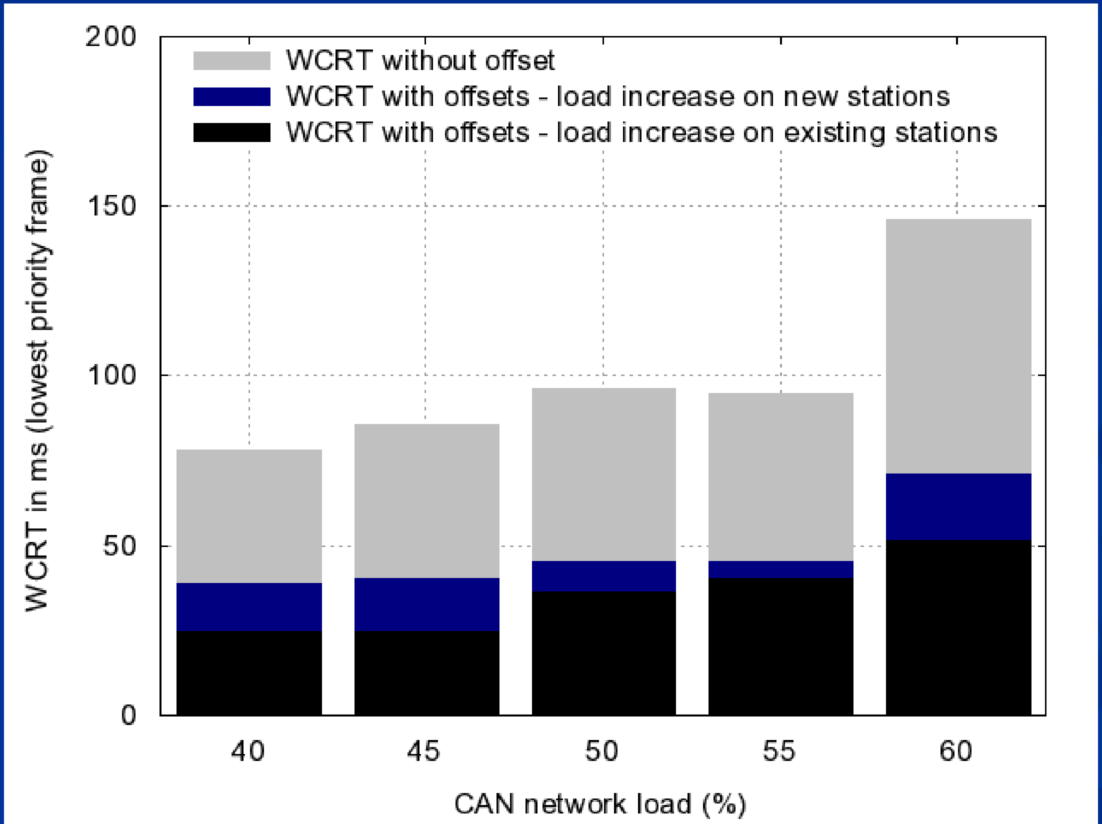
Chassis Networks



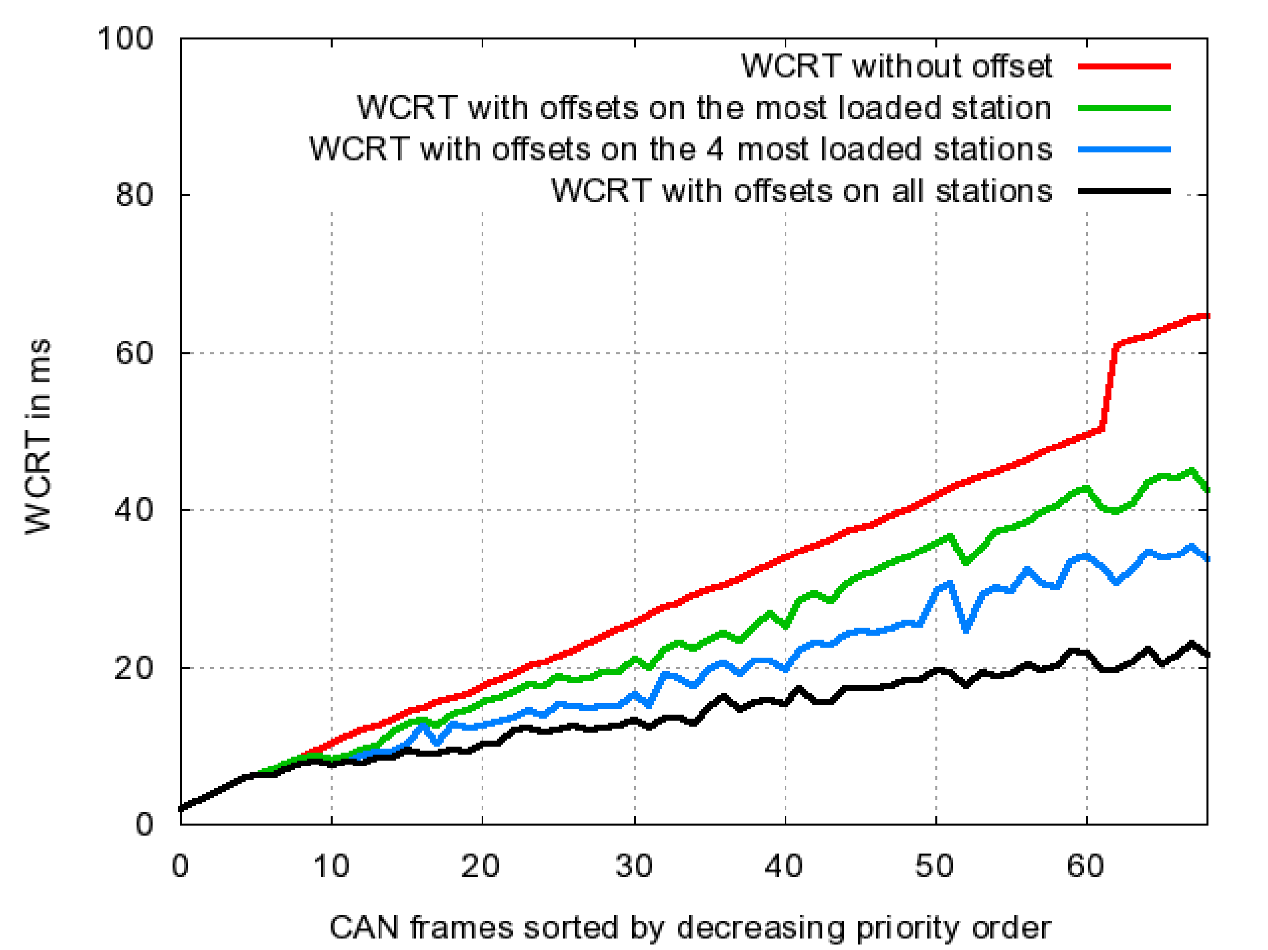
- Results are even better with loaded stations

Offsets allow higher network loads

- Typically: WCRT at 60% with offsets ≈ WCRT at 30% without offsets



Partial offset usage



65 ms

42

34

17

Conclusions

- Offsets provide an cost-effective short-term solution to postpone multiple CANs and FlexRay
- Tradeoff between Event and Time Triggered

- Further large improvements are possible by synchronizing the ECUs …

Questions, feedback?
please contact me at

nicolas.navet@realtimeatwork.com