Verification of automotive networks - what to expect (and not expect) from each technique

Nicolas NAVET – nicolas.navet@uni.lu

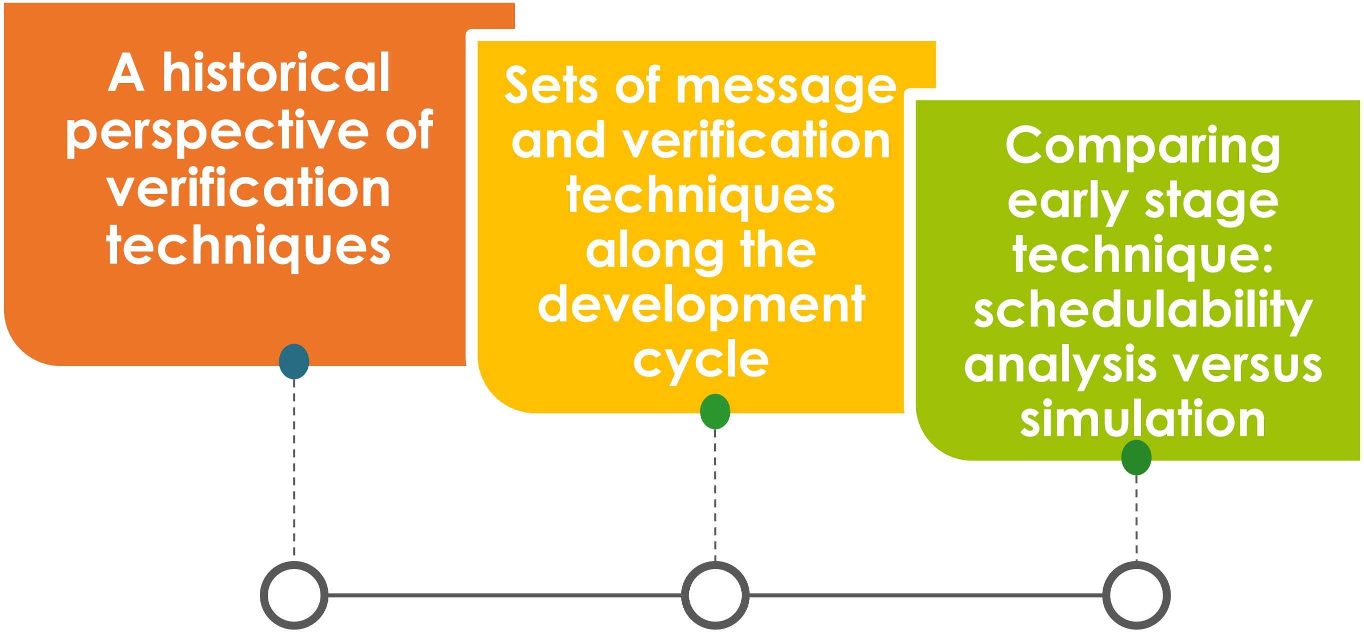
“Automotive Bus systems + Ethernet”

Stuttgart, Germany, December 9-11, 2013.

December 09, 2013

1 Outline

✓ Early-stage timing verification of wired automotive buses – illustration on CAN



1

Verification techniques and their use along the development cycle

If the workload submitted is bounded and the resources are deterministic, then it is always possible to provide timing guarantees

Schedulability analysis
"mathematic model of the worst-case possible situation"

VS

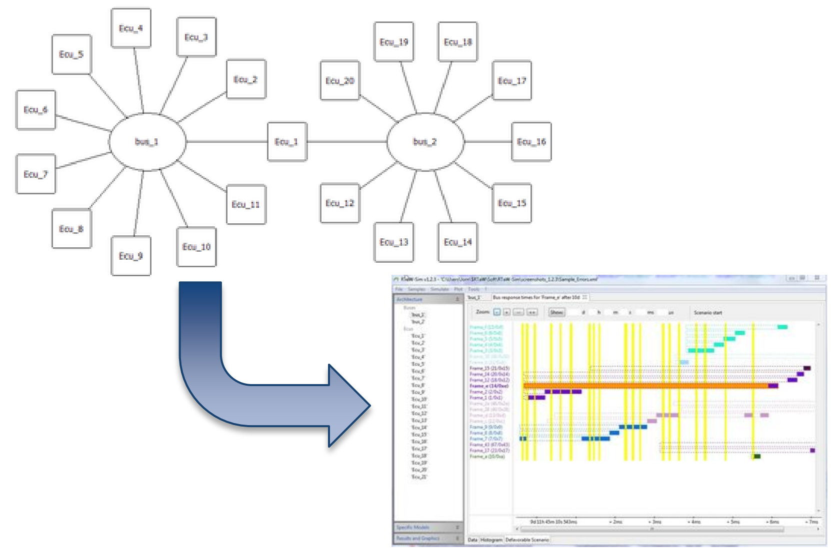
Simulation
"program that reproduces the behavior of a system"

$$K_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor + 1}$$

max number of instances that can accumulate at critical instants

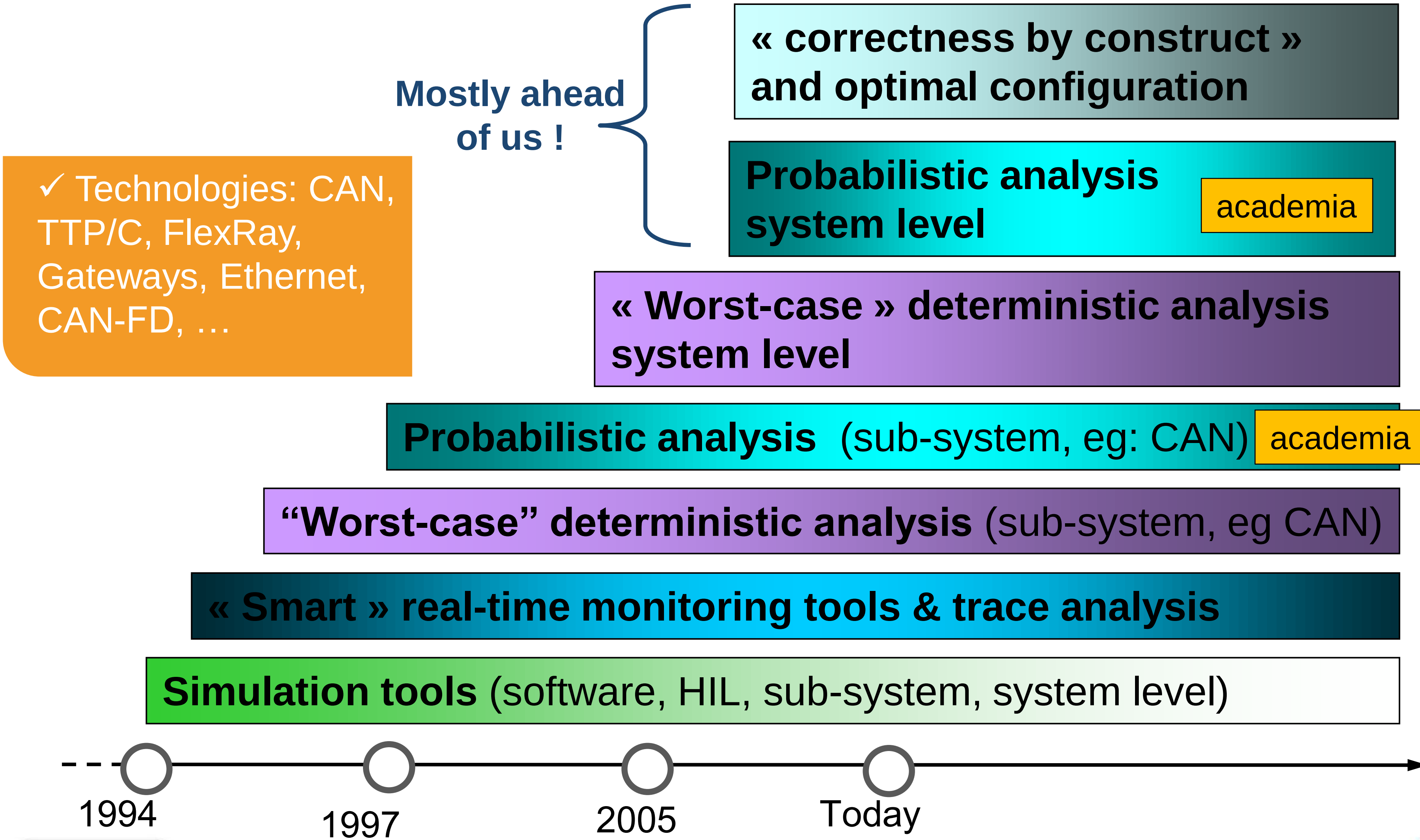
max number of instances arriving after critical instants

- ☺ Upper bounds on the perf. metrics → Safe (really?! – TBD)
- ☺ Analysis is known to be correct → Safe (really?! – TBD)
- ☹ Pessimistic → over-dimensioning
- ☹ Gap between models and real systems!
- ☹ Do not provide much information since a single trajectory is studied

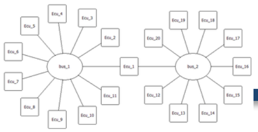


- ☺ Models close to real systems
- ☺ Fine grained information
- ☹ Upper bounds are out of reach! → Unsafe (really?! – TBD)
- ☹ Model correctness is unsure

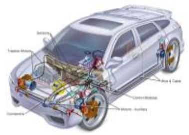
Historical development of verification techniques – personal perspective



Sets of messages and verification techniques along the development cycle



"Early stage" **"Project"** **"Real"**



"Early stage"

- ✓ **"Virtual" set of messages derived from existing ones**
- ✓ **Architecture design & technological choices**
- ✓ Coarse-grained verification
- ✓ System will be able to grow? Add frames, ECU, clusters ?

"Project"

- ✓ **Set of messages as specified by the designer**
- ✓ **Configuration:** offsets, priorities, frame packing, round, routing, etc
- ✓ Fine-grained verification .. but model-based

"Real"

- ✓ **Set of messages as seen in the car**
- ✓ errors, aperiodic, ECU clock drifts,
- ✓ Specifications are met ?
- ✓ Impact of non-conformance ?!

techniques

- ✓ **Workload generator**
- ✓ **Simulation & analysis**

[Netcarbench & RTaW-Sim]

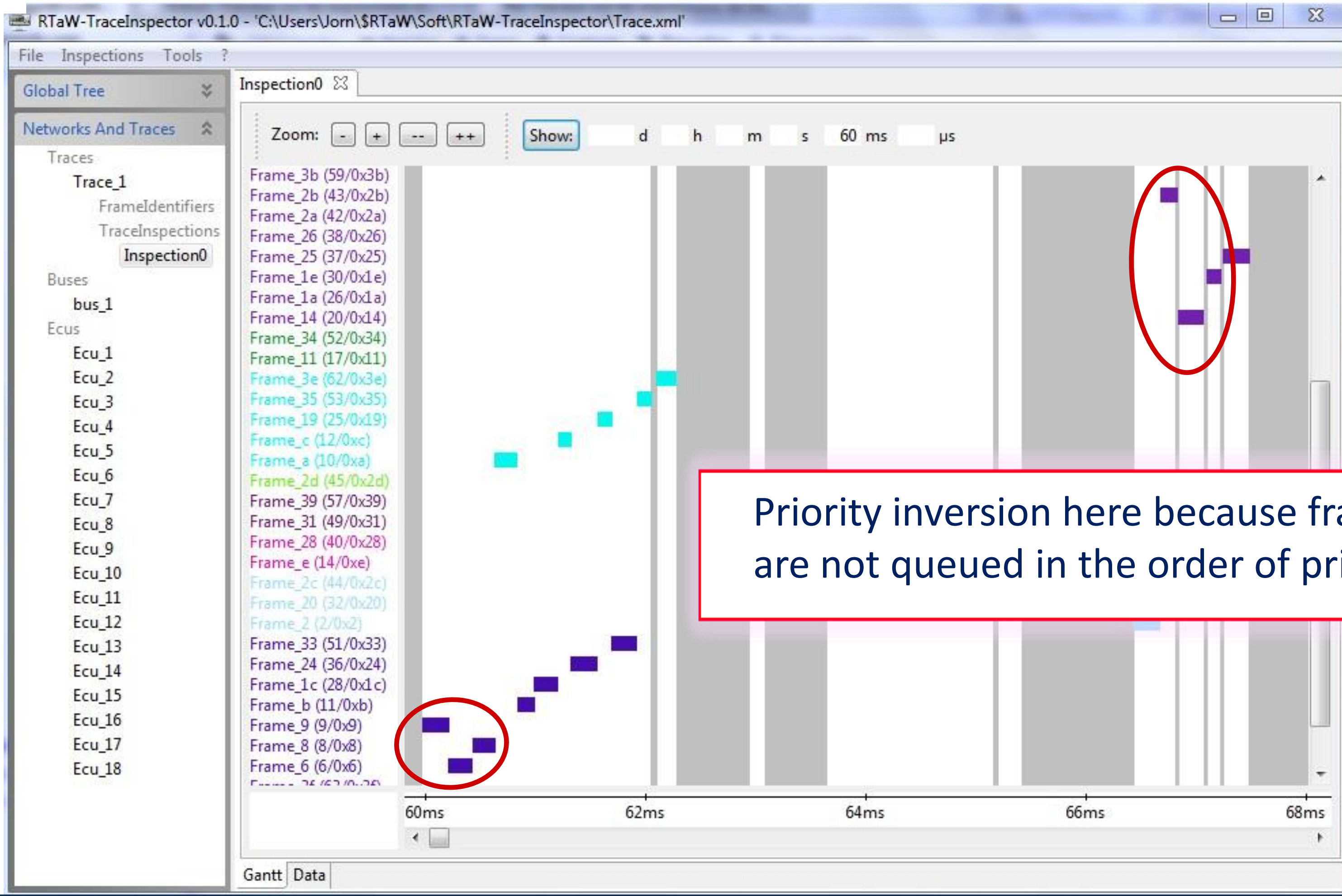
- ✓ **Configuration optimization**
- ✓ **Simulation & analysis**

[RTaW-Sim / RTaW-Pegase]

- ✓ **Monitoring tools**
- ✓ **Trace analysis**
- ✓ **Simulation & analysis with real traffic monitored**

[RTaW-TraceInspector]

Analyzing communication traces : are there departures from the specifications ?



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

Check comm. stack implementation, periods, offsets, jitters, model for aperiodic traffic and transmission errors, clock drifts, etc .. [*RTaW-Trace Inspector* screenshot]

2

Early-stage verification techniques : schedulability analysis versus simulation

Main performance metric: frame response time ≈ communication latency

“Time from transmission request until frame received by consuming nodes”

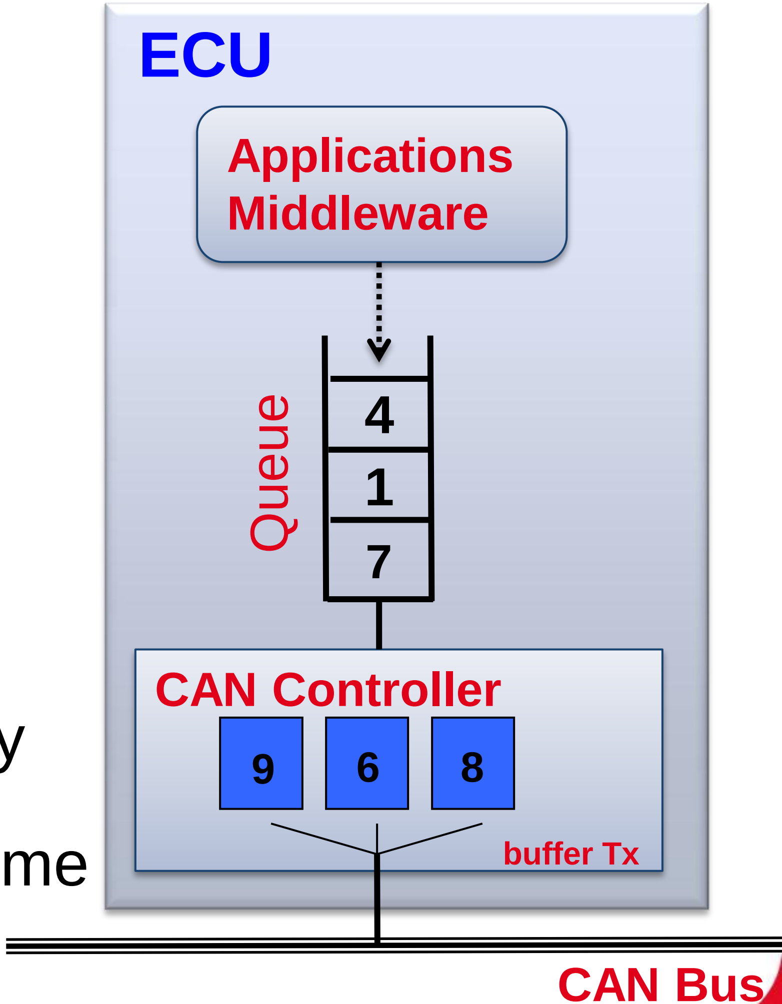
✓Synthetic metrics at the bus level : eg. Max (response time / deadline)

Software delay

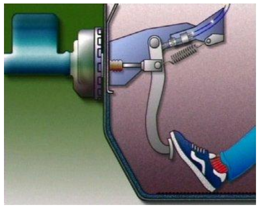
Waiting time in software queue

Arbitration delay

Transmission time



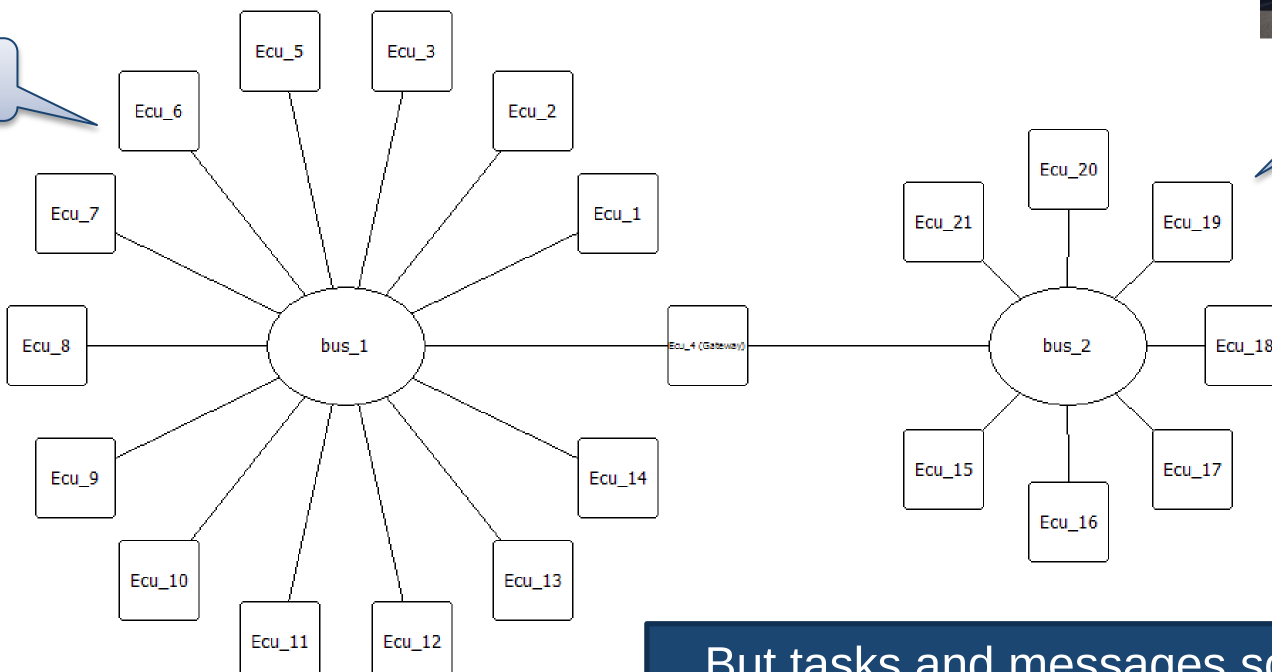
End-to-end response time verification has to handle for heterogeneous networks, task scheduling, gateways, etc



Constraint :
brake light on < 50ms

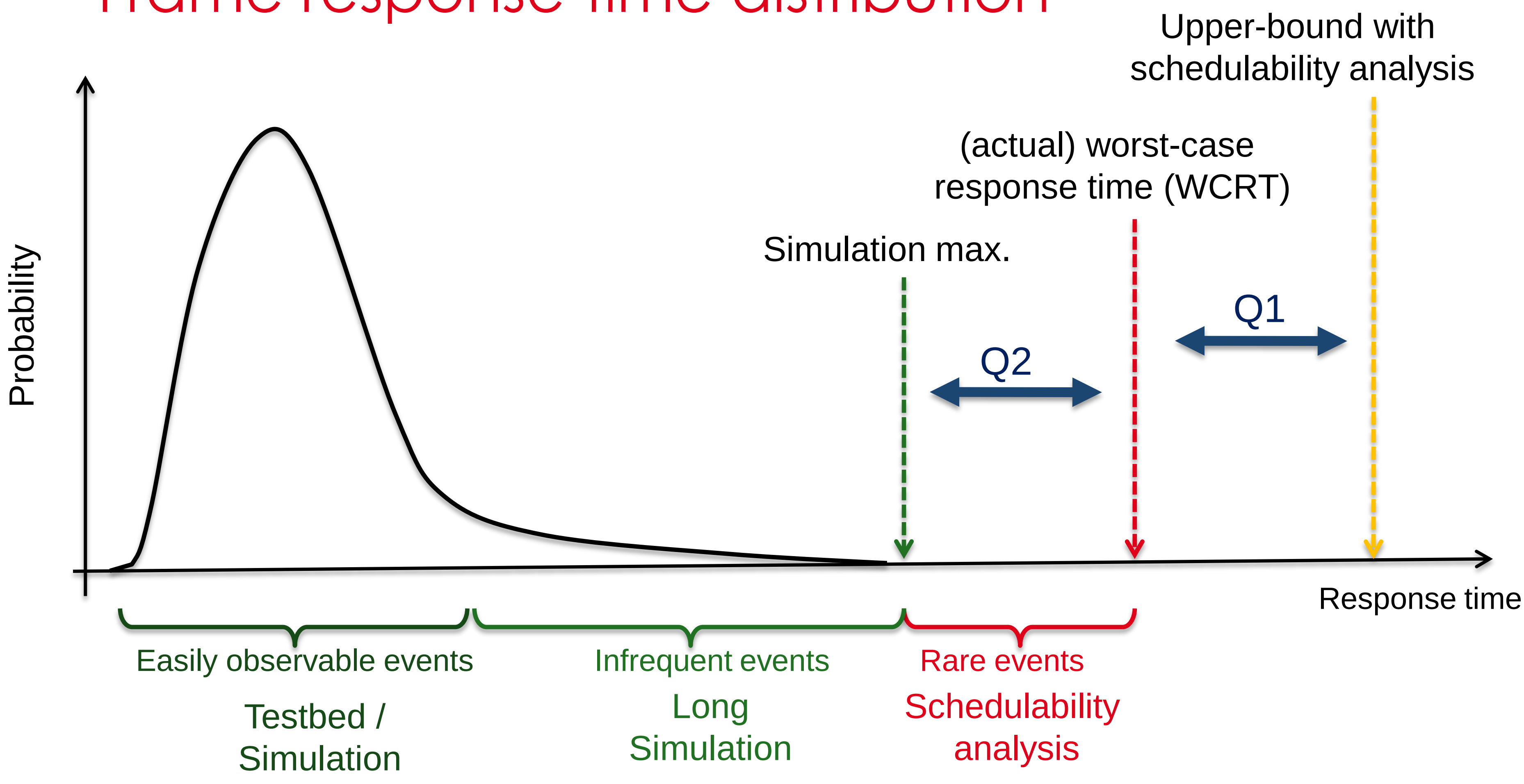
Stimulus

Response



But tasks and messages scheduling are often decoupled in the design …

Frame response time distribution

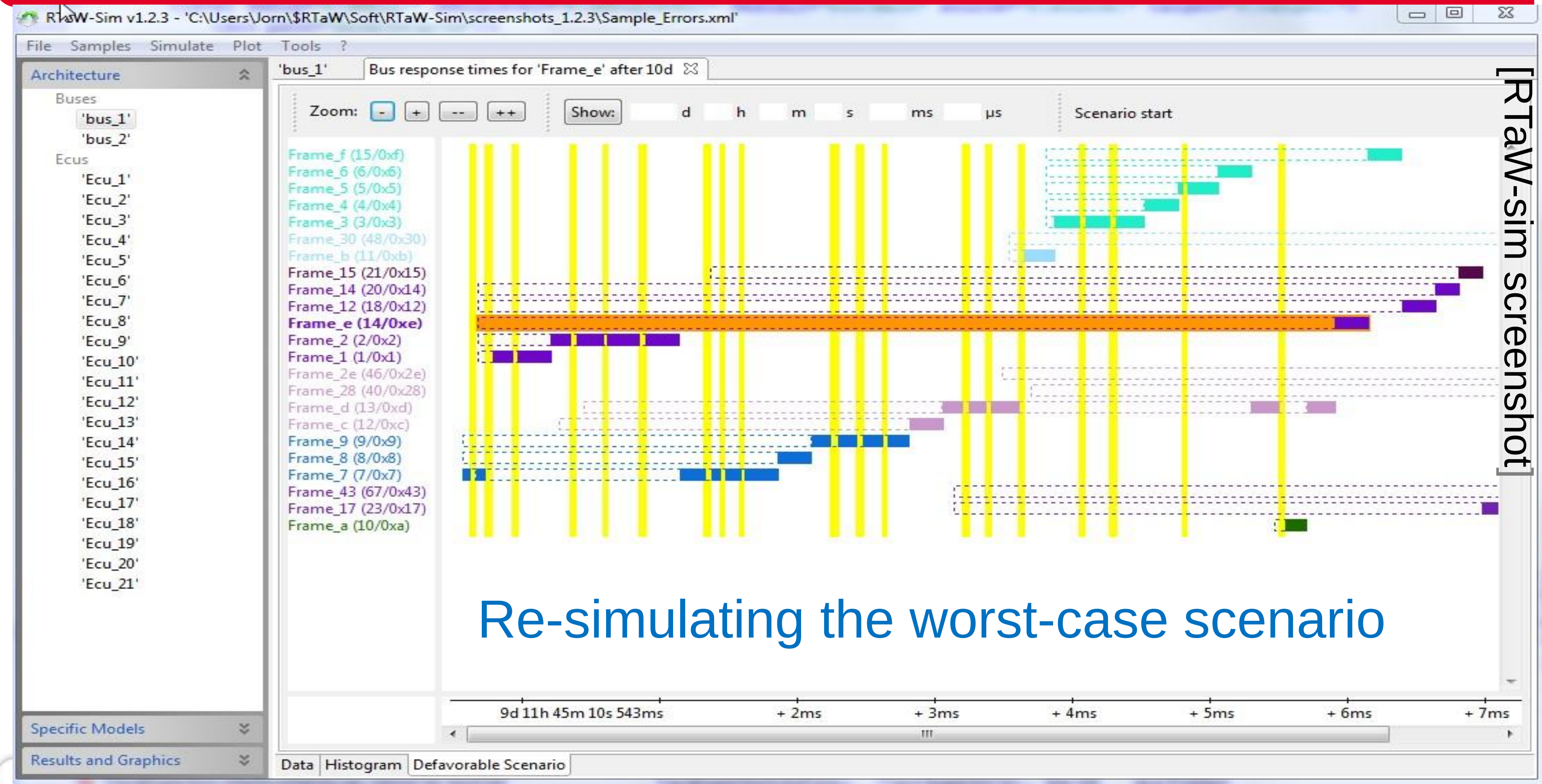


Q1: pessimism of schedulability analysis ?!

Q2: distance between simulation max. and WCRT ?!

Q1 : Pessimism of CAN schedulability analysis ?
Q2: distance with simulation ?

Case 1: ideal communication stacks + no gateway →
the computed upper-bound can occur (and be re-simulated)

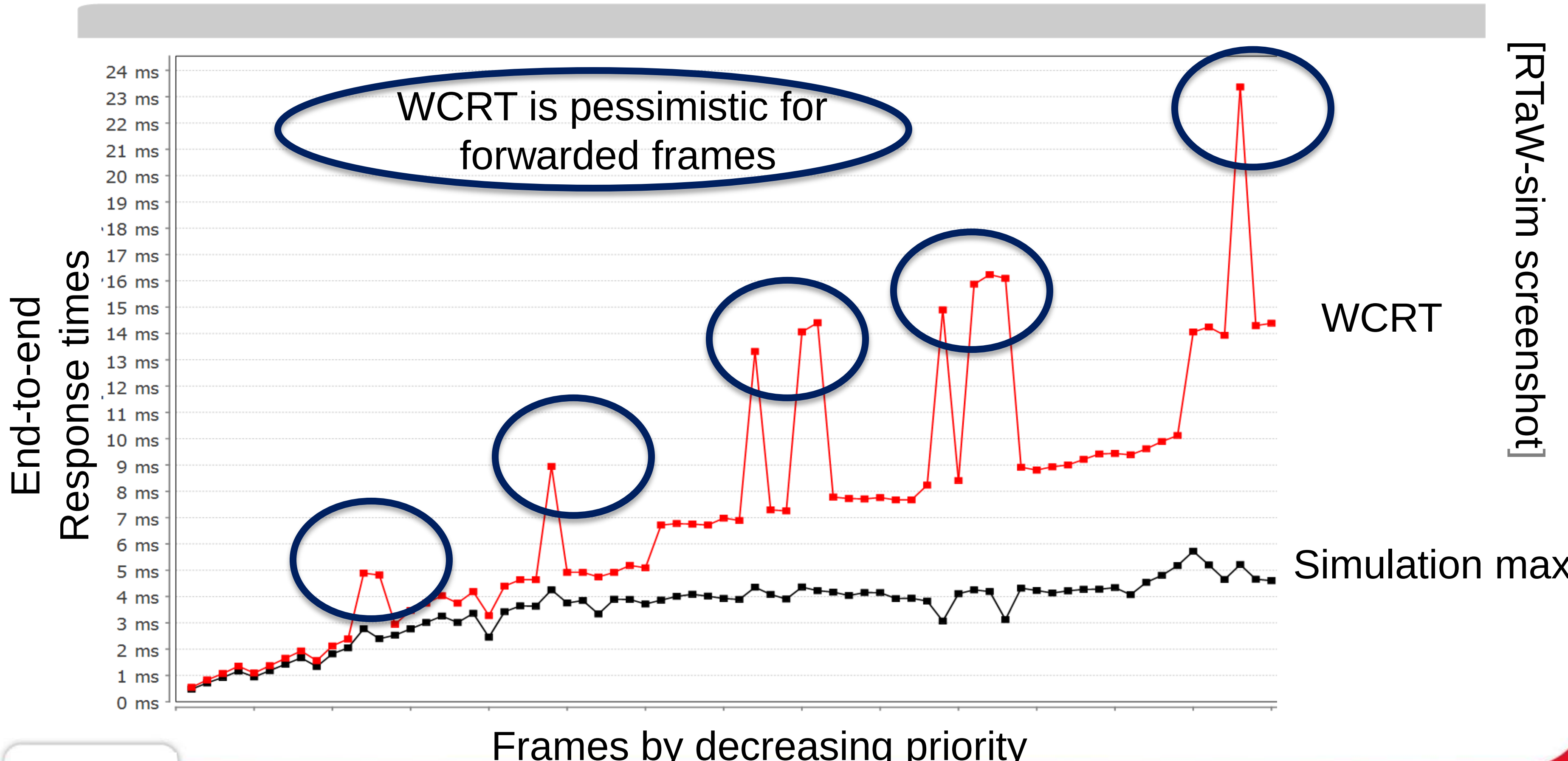


Re-simulating the worst-case scenario

[RTaW-sim screenshot]

Q1 : Pessimism of CAN schedulability analysis ?
Q2: distance with simulation ?

Case 2: perfect communication stacks + **gateway** →
the computed upper-bounds do not occur for forwarded frames in the general case



Beware of verification models !

"Schedulability analysis ensures safety!"
Our view: it might not be so…

1. Analytic models are pessimistic (except in the "ideal" case)
2. Analytic models are unrealistic (except in the "ideal" case)
3. Analytical models and their implementation can be flawed

"Simulation cannot provide firm guarantees"
Our view: it might not be so…

4. It is possible to verify correctness of simulation models
5. User- chosen guarantees can be enforced with proper methodology, e.g. with quantiles

Assumptions made by analytical models may not always be realistic

UNIVERSITÉ DU LUXEMBOURG

Possible departures from assumptions made : communication stack – illustration on CAN

1 Non-prioritized waiting queues [5,6]

2 Frame queuing not done in priority order by communication task

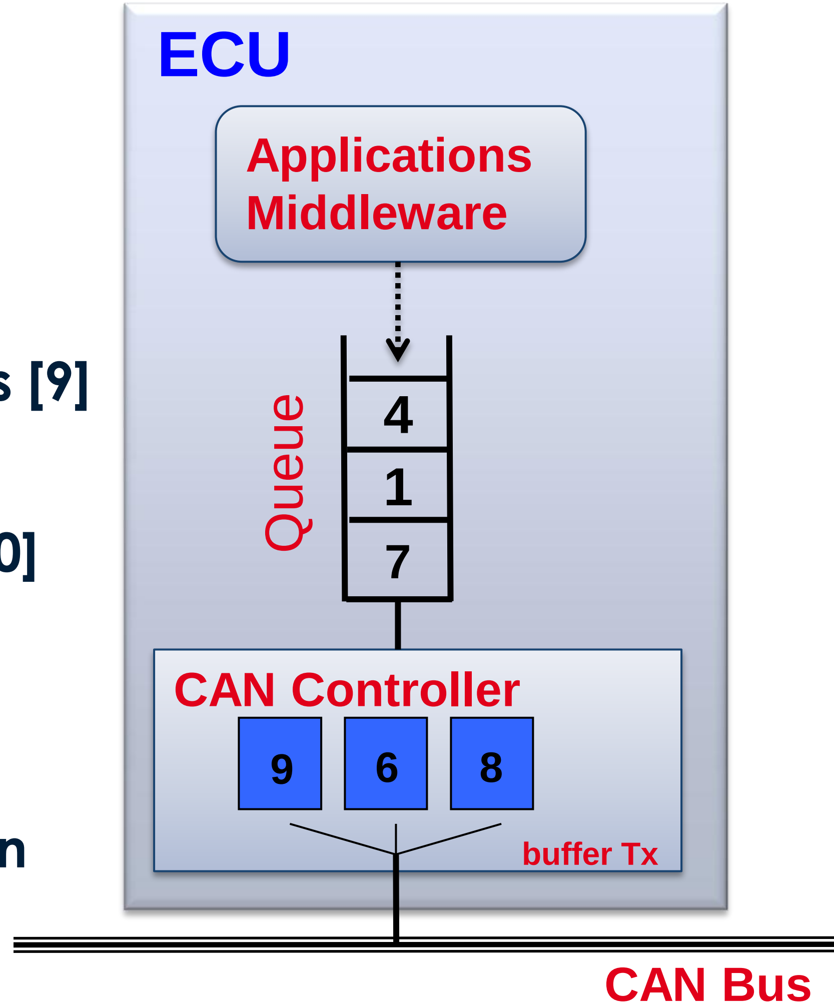
3 Non abortable transmission requests [9]

4 Not enough transmission buffers [8,10]

5 Delays in refilling the buffers [11]

6 Delay data production / transmission request

• • •

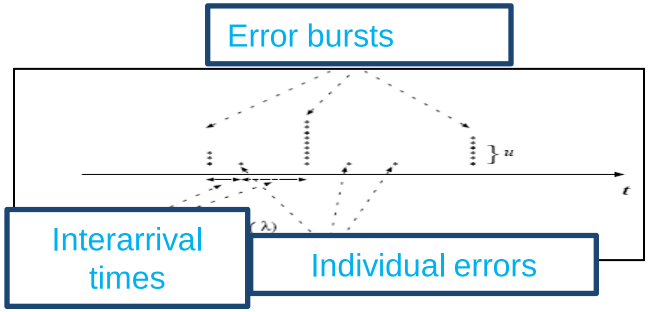


Possible departures from assumptions made: frame transmission patterns

7 **code upload or segmented messages**

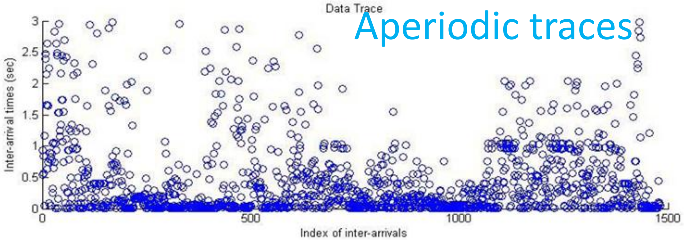
8 **Autosar-like mixed transmission models**

9 **Diagnostics requests**



10 **Transmission errors (probabilistic model ?! [1])**

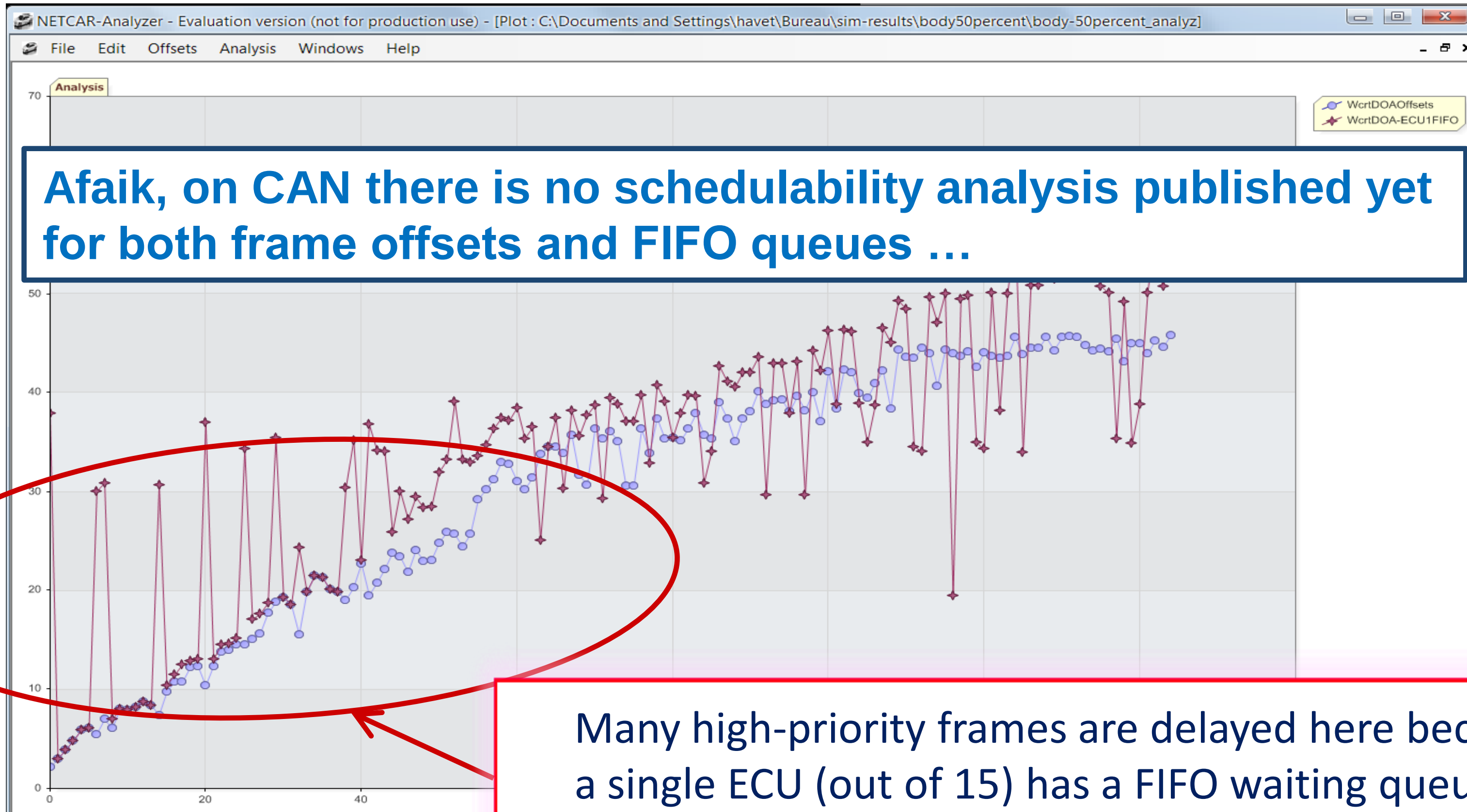
11 **Aperiodic traffic (probabilistic model ?! [2])**



12 **Gatewayed traffic**

...

If the analytical model does not capture accurately all the characteristics of the system, then the results will be wrong … in an unpredictable manner



Afaik, on CAN there is no schedulability analysis published yet for both frame offsets and FIFO queues …

Response times

NETCAR-Analyzer screenshot

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

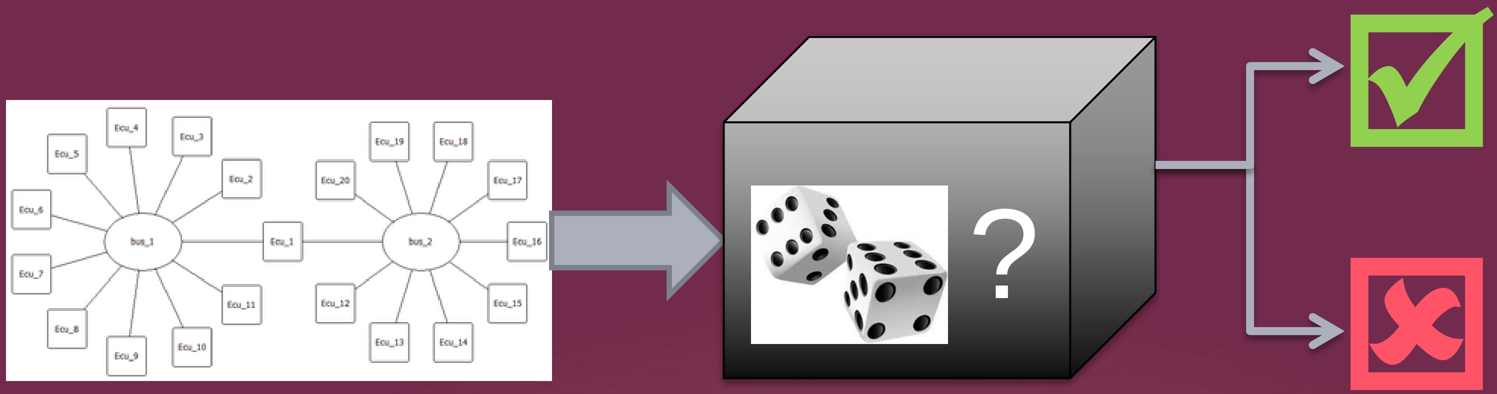
Frames by decreasing priority

Good news: many works try to bridge the gap between analytic models and real systems [Ref.1 to 12]

- ✓ However – not everything is covered, no integrated framework (first step in [6])
- ✓ And - many existing analyses are conservative (= inaccurate), thus hardly usable for highly-loaded systems.
- ✓ Alas - comprehensive and exact analysis would be overly complex (e.g. as in [9]) and intractable!

Personal view : both accurate and comprehensive analyses are out of reach … if you need analysis, you have to conceive the systems accordingly

Why should we trust verification models ?



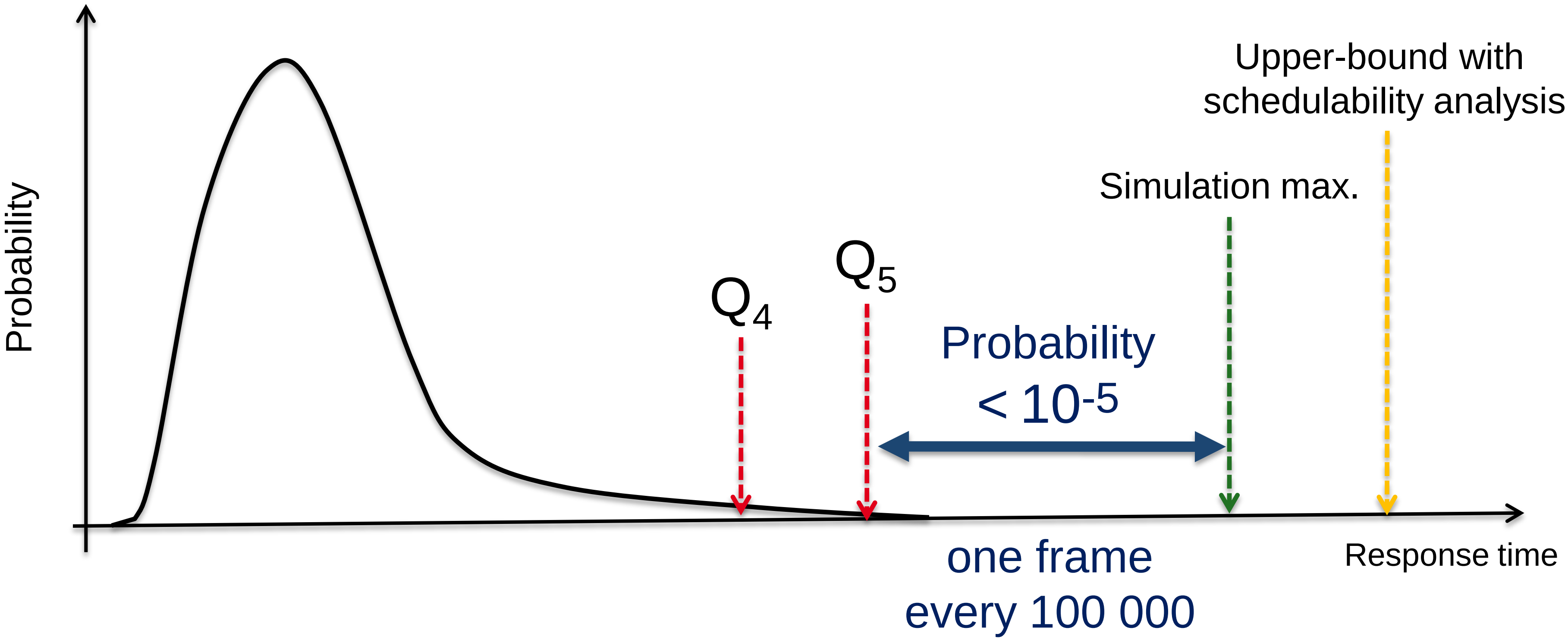
Models and software can be flawed …

- ✓ Schedulability analyses are complex and error prone. remember “CAN analysis refuted, revisited, etc” [14] ?! → peer-review of the WCRT analyses and no black-box software
- ✓ Schedulability analysis implementations are error prone: analyses complexity, floating-point arithmetic!, how to check correctness?, not many end-users, cost-pressure, etc …
- ✓ Easier to validate a simulator ? Yes …
  - Cross-validation by re-simulating worst-case situation from schedulability analysis (when possible)
  - Cross-validation by comparison with real communication traces: e.g., comparing inter-arrival times distribution
  - Checking a set of correctness properties on simulation traces

Simulation can provide guarantees with proper methodology

Using quantiles means accepting a ***controlled*** risk

Quantile Q_n: $P[\text{ response time} > Q_n\,] < 10^{-n}$



✓ No extrapolation here, won't help to say anything about what is too rare to be in simulation traces

1) How often performance objectives can be violated regarding frame criticality ?

Quantile	One frame every …	Mean time to failure Frame period = 10ms	Mean time to failure Frame period = 500ms
Q3	1000	10 s	8mn 20s
Q4	10 000	1mn 40s	≈ 1h 23mn
Q5	100 000	≈ 17mn	≈ 13h 53mn
Q6	1000 000	≈ 2h 46mn	≈ 5d 19h
…	…	…	

Warning : successive failures in some cases might be temporally correlated, this must be ruled out …

2) Determine the minimum simulation length

- ✓time needed for quantile convergence
- ✓ reasonable # of values: a few tens …

Min	Average	Q2	Q3	Q4	Q5	Q6	Max	Bound
0,236 ms	0,272 ms	0,466 ms	0,474 ms	0,477 ms	0,477 ms	0,477 ms	[illegible],477 ms	0,550 ms
				ms	0,719 ms	0,719 ms	[illegible]719 ms	0,830 ms
				ms	0,925 ms	0,925 ms	0[illegible]925 ms	1,074 ms
				ms	1,167 ms	1,167 ms	1,[illegible]67 ms	1,354 ms
				ms	0,943 ms	0,943 ms	0,[illegible]43 ms	1,092 ms
				ms	1,185 ms	1,185 ms	1,1[illegible]5 ms	1,372 ms
				ms	1,414 ms	1,427 ms	1,4[illegible]7 ms	1,652 ms
				ms	1,669 ms	1,669 ms	1,6[illegible]9 ms	1,932 ms
				ms	1,328 ms	1,339 ms	1,3[illegible]9 ms	1,564 ms
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] ms	1,791 ms	1,811 ms	1,8[illegible]2 ms	2,124 ms
0,218 ms	0,313 ms	1,061 ms	1,481 ms	1,750 ms	1,875 ms	2,009 ms	2,0[illegible]5 ms	2,386 ms
0,522 ms	0,686 ms	1,490 ms	1,897 ms	2,116 ms	2,267 ms	2,388 ms	2,50[illegible] ms	4,890 ms
0,450 ms	0,615 ms	1,398 ms	1,811 ms	2,104 ms	2,293 ms	2,402 ms	2,67[illegible] ms	4,818 ms
0,720 ms	0,929 ms	1,832 ms	2,128 ms	2,280 ms	2,374 ms	2,486 ms	2,5[illegible]5 ms	2,946 ms
					2,573 ms	2,710 ms	2,7[illegible]6 ms	3,470 ms
					2,618 ms	2,710 ms	2,8[illegible]3 ms	3,750 ms
					2,989 ms	3,166 ms	3,2[illegible]4 ms	4,030 ms
					2,773 ms	2,854 ms	2,[illegible]41 ms	3,750 ms
					2,854 ms	2,989 ms	3,[illegible]03 ms	4,186 ms
					2,092 ms	2,153 ms	2[illegible]238 ms	3,276 ms
					2,854 ms	2,971 ms	[illegible],060 ms	4,396 ms
					3,277 ms	3,373 ms	3,460 ms	4,640 ms
					3,076 ms	[illegible] ms	3,239 ms	4,640 ms
					3,698 ms	[illegible]06 ms	3,871 ms	8,946 ms
					3,412 ms	3,483 ms	3,483 ms	4,920 ms
					3,491 ms	3,864 ms	3,864 ms	4,920 ms
					3,129 ms	3,181 ms	3,181 ms	4,744 ms
					3,451 ms	3,548 ms	[illegible],548 ms	4,920 ms
					3,392 ms	3,532 ms	[illegible],532 ms	5,182 ms
					3,315 ms	3,336 ms	[illegible],336 ms	5,094 ms
					3,431 ms	3,817 ms	[illegible],817 ms	6,718 ms
					3,511 ms	3,733 ms	[illegible],733 ms	6,772 ms
					3,471 ms	3,587 ms	[illegible],587 ms	6,754 ms
0,182 ms	0,391 ms	2,068 ms	2,726 ms	3,148 ms	3,412 ms	3,578 ms	3,578 ms	6,718 ms
0,166 ms	0,383 ms	2,080 ms	2,805 ms	3,184 ms	3,416 ms		3,416 ms	6,982 ms

Tool support can help here: e.g. numbers in gray should not be trusted

Reasonable values for Q5 and Q6 (with periods <500ms) are obtained in a few hours of simulation (with a high-speed simulation engine) – e.g. 2 hours for a typical automotive setup

[RTaW-sim screenshot]

3

Concluding remarks

UNIVERSITÉ DU LUXEMBOURG

Simulation vs analysis

1 There might be a gap between assumptions made for analytic models and the real system

- ✓ pessimistic at best, can be unsafe
- ✓ no dramatic improvements in sight
- ✓"analyzability" should be a design constraint if needed

2 Simulation is a practical alternative even for critical systems .. with the proper methodology

- ✓ Determine quantile wrt criticality, and simulation length wrt to quantile
- ✓ Simulator and models validation
- ✓ High-performance simulation engine needed for higher quantiles

Increasingly complexity & higher load level calls for

1. More constraining specifications, or conservative assumptions → a single node can jeopardize the system
2. Combined use of verification techniques:
   - Refinement of traffic knowledge over time
   - Simulation and/or analysis, and trace inspection
   - none of them alone is sufficient

✓No verification model & tool
can be trusted blindly – always question assumptions

✓ If schedulability analysis is required,
the (sub-)system should be conceived accordingly,
otherwise simulation is - in our view - a better option

Interested in this talk and simulation methodology?

Please consult our appear at ERTSS'2014: "Timing verification of automotive communication architectures using quantile estimation" co-authored with Shehnaz LOUVART (Renault), Jose VILLANUEVA (Renault) and Jörn MIGGE (RealTime-at-Work).

4 References

References

Most available from http://nicolas.navet.eu

[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.

[2] D. Khan, N. Navet, B. Bavoux, J. Migge, “Aperiodic Traffic in Response Time Analyses with Adjustable Safety Level“, IEEE ETFA2009, Mallorca, Spain, September 22-26, 2009.

[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.

[4] P. Meumeu-Yomsi, D. Bertrand, N. Navet, R. Davis, "Controller Area Network (CAN): Response Time Analysis with Offsets", 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", 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.

[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.

[12] R. Davis, N. Navet, “Traffic Shaping to Reduce Jitter in Controller Area Network (CAN)“, ACM SIGBED Review, Volume 9, Issue 4, pp37-40, November 2012.

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

[14] R.I. Davis, A. Burns, R.J. Bril, J.J. Lukkien. “Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised”, Real-Time Systems, Volume 35, Number 3, pp. 239-272, April 2007.