Fine-grained Simulation in the Design of Automotive Communication Systems

Aurélien Monot[1,2], Nicolas Navet[3], Bernard Bavoux[2], Cristian Maxim[1]

1: LORIA - Nancy Université, BP 239, 54506 Vandoeuvre, France
2: PSA Peugeot Citroën, Centre technique de Vélizy, Route de Gisy, 78 943 Vélizy-Villacoublay Cedex, France
3: INRIA - RealTime-at-Work, 615 Rue du Jardin Botanique, 54506 Vandoeuvre-lès-Nancy, France

Abstract: Early in the design cycle, the two main approaches for verifying timing constraints and dimensioning automotive embedded networks are worst-case schedulability analysis and simulation. The first aim of the paper is to demonstrate that both provide complementary results and that, most often, none of them alone is sufficient. In this paper, we present a simulation approach accounting for the clock drifts that occur on the network nodes at run-time and evaluate the extent to which the results obtained with this approach are relevant for the designers in order to validate the performances of a CAN-based communication system. One of the practical outcome of this study is to show that initial phasings between nodes, as well as the values of the clock drifts, do not significantly impact the frame response time distributions that can be observed on the long run.

1 Introduction

Context. Electronic architectures in the automotive domain are defined years in advance and their real-time properties need to be evaluated and validated early in the design process. Amongst numerous design concerns, engineers are driven by two primary objectives which are to ensure safety and to dimension the electronic architecture so as to obtain a good trade-off between cost and performances. In order to achieve these goals two approaches are mainly used when designing an automotive communication system: worst-case analysis and simulation. As CAN is the main communication protocol between ECUs (Electronic Control Units) in cars, CAN frame response times have been studied extensively by the means of analysis and simulation. If Worst Case Response Times (WCRT) were initially the main focus of these research studies in order to ensure the safety requirements, response time distributions were eventually looked into as they provide richer information on the real-time behavior of the system. Experiments suggest that WCRT are rare events and quantiles such as 99,99% might be more relevant when dimensioning the electronic architecture as they can be significantly lower and might still provide enough guarantees to meet the safety requirements for many of the frames. Thus, being able to obtain accurate response time distributions might lead to significant cost optimization on hardware.

Previous work. The timing analysis of CAN has been rather extensively investigated in the past. Bounds on the worst case response times have first been provided in [1] and [2]. Studies have integrated the limitations of hardware [3, 4, 5], and the communication stack, as the first analyses usually overlooked them, and also considered the effect of aperiodic traffic on CAN frame response times [6] and the consequences of transient perturbations [7]. Of course, methods to minimize the response times, or make them more predictable, were investigated too, e.g. in [8] and [9].
In the previous works, some studies provide probabilistic analysis, and specifically response time distributions as a consequence of 1) all the possible phasing configurations between ECUs [10, 11] and 2) CAN bit-stuffing mechanism [12]. However, though they are a subject of interest in the field of wireless sensor networks [13], none of these works assumed the possibility of ECU clock drifts until [14] which this paper builds upon. Clock drifts result from the fact that the clocks of the ECUs which drive the instantiations of the CAN frames do not exactly operate at the same frequency. Due to production tolerances, the oscillators are not exactly identical and their frequency may also change over time because of environmental factors such as the temperature. In practice, the clock drifts are unavoidable and cause the phasings between the communicating ECUs to vary continuously over time. As a direct consequence, the response times of frames vary continuously too.

Paper organization. The paper is organized as follows. Section 2 describes our approach to simulate CAN networks. Then, section 3 describes how simulation can be used along analysis to study the worst case scenario. In the ensuing section 4, we present how to use our simulation approach to derive and interpret frame response time distributions. Finally, section 5 concludes this paper.

2 Fine-grained simulation of a CAN Network

In this section, we describe our simulation model and present the simulation tool. Specifically, we detail the clock model and assumptions made on the CAN nodes as well the kind of results obtained from the simulation.

2.1 Modeling the CAN network

2.1.1 Clock drifts

In car electronic architectures, ECUs are typically driven by quartz oscillators. If ceramic resonators are considered for the less critical applications, they usually do not meet the requirements on maximum clock drift to drive CAN communication controller. For quartz oscillators, clock drifts result from various factors, the main of them being fabrication tolerance, aging, and temperature. Humidity and vibrations may also influence the clock drift of a quartz oscillator but are considered negligible compared to the ambient temperature and the quality of the quartz. Clock drifts are measured in "parts per million" or *ppm*, which express how slower or faster a clock is compared to a "perfect" clock. For instance, 1 ppm corresponds to a deviation of 1μs every second. Typical values provided by quartz manufacturers for an automotive usage are tolerances of +/- 50 ppm for the fabrication, +/- 5 ppm per year for aging and +/- 150 ppm for temperature in their operating temperature range (*i.e.*, -40℃/125℃). In practice, the observed values tend to be smaller, *e.g.* +/- 20 ppm over the whole temperature range.

In this study, we choose a rather simple and widely applicable drift model, based on fixed deviations of clock speeds (positive or negative) with respect to the nominal bus speed. Here, the variation due to aging is assumed to be negligible for the simulation lengths that we are considering. Furthermore, we assume that the CAN nodes are operating at a constant temperature. For a given clock c driving a CAN node, its local time t_c with respect to a global time t is simply as follows :

$$t_c(t) = \phi_c + \delta_c \cdot t$$

where ϕ_c is the initial start time (the phasing) of the CAN node with regard to the bus time referential, and δ_c is the constant drift value. For instance, a drift rate of +50ppm means that $\delta_c = 1.000050$. For this work, every CAN node j is assigned a clock defined by the tuple (ϕ_j, δ_j).

2.1.2 Hypotheses on the CAN communication system

For this work, no specific hardware characterics, other that the clock drifts, are considered. At each communicating node, the messages are queued in priority order and there is a sufficient number of communication buffers such that the highest priority message is always considered for arbitration. Additionally, the copy time from the message queue to the communication buffers is assumed to be negligible.

The CAN frames are considered to be strictly periodic. A CAN message m_i is described by (T_i, O_i, C_i, P_i) where T_i is its period, O_i its initial offset, C_i its transmission time and P_i its priority. In the context of this study, the worst case of bit-stuffing is always considered for C_i. Offsets serve to desynchronize the streams of frames sent by a node so as to reduce peak loads and improve the response times as typically as done in automotive networks today (see for instance [8]).

Since, the instantiations of the CAN messages are driven by their sending node clock, quantities T_i and O_i are measured by the local clock of the sending node. As a result, messages with equal periods but sent by different CAN nodes may have slightly different transmission periods with respect to the bus time reference. On the other hand, C_i, the transmission time on the CAN bus, depends only of the frame payload and the bit stuffing, and is expressed in the nominal time referential. Simulating this system consists in computing the frame release dates and reproducing the CAN arbitration and transmission process.

2.2 Assessing frame response time distribution

Frame response times essentially depend on the frame characteristics and of the phasings between the communicating nodes. Upper-bounds on the response times, valid whatever the phasings between ECUs, can be computed with analysis tools such as NETCAR-Analyzer [1] and they correspond to the worst-case behavior of the network. Response times in the typical case can be better evaluated by simulation. The novelty of the simulation study performed in this paper is that clock drifts are explicitly modeled and their impact on the network performances are evaluated. The classical simulation approach is to derive statistics on a number of configurations corresponding to distinct initial configurations (*i.e.*, different phasing between ECUs at the system startup). This is needed because a single simulation run only captures a limited set of response times for each frame, as the period of the system is small (typically a few seconds). Here we explore another approach which is to gather statistics on a single trajectory of the system simulated during a long time (*i.e.*, one simulation run). As the phasings between the ECUs

[1] NETCAR-Analyzer is a worst-case response time analysis tool [15] ©INRIA/INPL/RTaW

vary continuously due to the clock drifts, simulating a single trajectory is sufficient to observe a wide range of response times for each frame. Because there is a single simulation run to perform, this approach is more convenient from an experimental point of view. Besides, it is more realistic in the sense that clock drifts, which unavoidably occur at run-time, are accounted for. The aim of the paper is to evaluate the extent to which the results obtained with this approach is relevant for the designers in order to validate the performances of a CAN-based system.

2.3 Software toolset

The main software used in this study, RTaW-Sim [16] is a fine-grained discrete-event CAN bus simulator providing the frame response time distributions. It is free for all uses and available for download. The granularity of the simulation is $1\mu s$ allowing thus to simulate accurately a CAN bus at the bit level even at the highest possible speed of 1 Mbit/s. RTaW-Sim has various other features such as a precise modeling of the communication buffers, and fault injection capabilities that will not be used here. Also, we are going to use its import feature from NETCAR-Analyzer that allows to replay the phasing scenarios leading to the worst case response times[2].

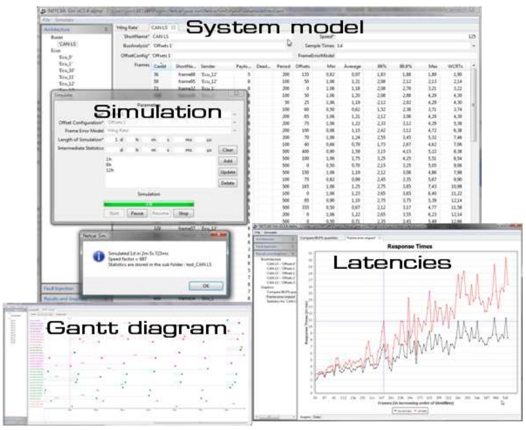

Figure 1: *Screenshot of RTAW-Sim. The largest window in the underground contains the frames properties. The two smallest windows show the simulation options and its progress. The last bottom right window displays simulation statistics after a run.*

In this paper, we present the experimental results obtained by simulation on realistic sets of message. The simulator is able to compute about 700 kilo-events per seconds on a 2 GhZ laptop computer, which means for instance that it can simulate 24 hours of communication on a 60% loaded CAN bus in about 20 minutes. When simulating a CAN bus, the usual approach (see §2.2) is to compute a few hyperperiods for as much phasing configurations possible, resulting often in tens of hours of simulation even using a rather coarse-grained time granularity.

3 Exploring worst case scenario

In order to ensure safety, design engineers have to look at the worst case response time of frames which is typically done by analysis. However, analysis techniques do not tell us for how long the worst cases last, nor how often these situations happen.

3.1 Finding the worst case scenario

Finding the worst case response times using analysis is a problem that can be solved with some COTS software tools such as Netcar-Analyzer[15]. However, being able to find the worst case scenario through simulation alone proved in our experiments to be almost impossible on realistic-size network. Determining a worst case response scenario for a frame consists in identifying the specific phasings between sending ECUs leading to the longest possible response time.

The number of phasing configurations for a CAN network is $(\frac{T}{\tau})^N$ with T being the hyperperiod of the set of periodic frames, τ the bit-time of the bus and N the number of ECUs. For a typical high-speed CAN network at 500 kbit/s made up of 10 ECUs sending a set of periodic frame having a hyperperiod of 2s, this amounts to 10^{60} different phasing configurations. Amongst them, several phasing configurations could possibly lead to the same response time but the number of phasing configurations giving different response times is still very large. As a consequence of the clock drifts, numerous phasing configurations will be explored in a single simulation run but this still does not guarantee to find a phasing configuration triggering a worst case response time. Fortunately, we can use the results of an analysis tool to reproduce the phasing parameters leading to a worst case scenario for a specific frame. This will be done in the next subsection.

3.2 Reproducing a worst case scenario

In these first experiments, we will replay a configuration leading to the wort-case response time for a specific frame and examine what happens shortly after a worst case response time. In order to do that, we start a simulation with the phasing parameters between ECUs leading to the a frame WCRT that we obtain through analysis. Here, the message set used is the SAE benchmark at a speed of 250 kbit/s corresponding to an average bus load of 62,3%. The frame under study has identifier 49. The offsets for the frames were computed using the SOA algorithm from NETCAR-Analyzer. Simulation statistics are shown in figure 2. We simulated four configurations corresponding to different drift parameters. The reference is represented by the black and green curves giving the average and maximum response time of the frames with no drift. Since there is no drift, and simulation is larger than the hyperperiod of the

[2]For each frame, the initial offset configuration leading to the worst case is usually different.

system, these two curves are identical for the three simulation length. The other curves correspond to statistics obtained with random drift values in +/- 50 ppm, +/- 150 ppm (realistic drift values) and +/- 1000 ppm (very pessimistic with respect to suppliers data sheet).

Figure 2: *Simulation statistics with initial phasings leading to worst case response time for the frame 49. From top to bottom, the graphics show the frame response time for the first 10s, 1mn and 10mn of simulation. The X-axis are the frames sorted by decreasing order of priority. The black curve is the maximum response times recorded without drift. The other curves are the average response times recorded without drift (green) and drift bounds resp. equal to 50 ppm (red), 150ppm (blue), and 1000 ppm (purple).*

The top graph shows that after 10s of simulation (5 hyperperiods), the average response times for all the configurations (except +/- 1000 ppm) are very close to the maximum response times of the case without drift which is the sign that most response times are large. The middle graph shows that after 1mn, the average response time have significantly decreased. It is worth pointing out that configurations with higher drifts lead to lower average values because the system get away from the worst case phasing configuration quicker. Finally, the bottom graph shows that after 10mn the average response time curves stabilize around values that are significantly lower than with smaller simulation lengths, whatever the clock drifts. Considering that 150 ppm is a realistic drift bound value, these experiments suggest that the frames stop experiencing large average response times above 1mn. Interestingly, this means that clock drifts, which are usually seen as being detrimental because they reduce the predictability of the system, might also be beneficial in some cases: they help getting away from phasing configurations leading to large response times.

3.3 Maximum observed response time versus worst-case scenario

In this second set of experiments, we simulated during 4 hours (i.e., a long car trip) different CAN networks using 20 random different initial phasings. For a low priority frame, we recorded the worst observed values among the simulation runs for the response times, average response times and two high quantiles. For this set of experiments, we used the SAE benchmark, the message set introduced in [10] and a message set which was derived from an existing CAN set from PSA Peugeot-Citroën. Results are given in Figure 3.

CAN networks	SAE	Zeng, et al.	PSA
ECU number	6	6	18
Frame number	53	69	101
Bus speed (kbit/s)	250	500	125
Bus load	62,3%	60,25%	33,8%
Worst case (ms)	4,408	4,414	**30,414**
Maximum (ms)	4,408	4,163	**10,195**
99,9% quantile (ms)	4,125	3,675	8,65
99% quantile (ms)	3,875	2,975	5,85
Average (ms)	1,332	0,721	0,96

Figure 3: *Statistics for a low priority frame of each benchmark obtained for random initial offsets and a drift range of +/- 150 ppm. The values in the 4 last rows are the highest values observed in the set of experiments done for the corresponding CAN network.*

Interestingly, we were able to find the worst case scenario for the first network. Though, it should be noted that the number of ECUs is small, thus considerably lowering the number of possible phasing configurations. However, it should be noted that the average and high quantile values of the response time distributions are always noticeably lower than the worst case, especially for the third network where the number of nodes is realistic in view of today's automotive networks.

4 Deriving frame response time distributions

In this section we study the extent to which the results of simulations with drifts relate to the expected behavior of the network.

In other words, we explore the impact of the simulation parameters on the resulting frame response time distributions to determine the relevance of simulating a single trajectory with clock drifts for a long time. The purpose of having a long observation window is to be able to observe the overall response time statistics over long scenarios (a single trip or the car estimated total functioning time, e.g. 7000 hours). The statistics obtained by simulation should indicate what kind of performance can be expected from the studied configuration. This should allow engineers to draw conclusions about the dimensioning of the CAN bus: its speed and set some characteristics of the frames (e.g., priority, offsets, periods).

4.1 Impact of the simulation length

Here we study how the statistics stabilize over time. While doing preliminary simulations with large simulation length, we observed that frame response time statistics vary at the begining but eventually stabilize. In fact, with drifts the system is still periodic but the period is however considerably larger.

Supposing that the hyperperiod of the frames sent by a node i is T_i, the hyperperiod of the arrival of the frames from this node becomes $T_i' = T_i \cdot \delta_i$, with δ_i being the drift value for the sending node. If we do that for every node, we are still simulating a periodic system though the hyperperiod of the system became much higher (ppcm of the $T_i \cdot \delta_i$). For instance, assuming hyperperiods equal to T for N sending nodes and drift value close to δ leads to a period that can be up to $(T \cdot \delta)^N$ which leads to approximately $(2s \cdot 1.00015)^{10} \approx 17mn$ for $T = 2s$, 10 nodes and drift values in the range of +/-150ppm. The scheduling of the CAN frames become thus periodic after an initial phase (assuming the load of the system is less than 100%). As a consequence, the distribution of response times corresponding to the periodic phase of the scheduling will be repeated as the simulation lasts, making the response times recorded during the initial phase negligible.

This means that it is not needed to simulate the CAN networks more than for a certain time as the response time distributions will no longer change after some point. It should be noted that this only explains the fact that the response time distributions stabilize as the simulation goes on but this does not mean that the resulting distributions should be identical. In the next experiments we compare the distributions resulting from different initial offsets and initial drift values.

4.2 Impact of initial offsets between sending nodes

In these experiments, we study the overall behavior of the CAN frames response time statistics over long simulations. We provide here results for a simulated time of 24h. The message set is the one introduced in [10] which is made up of 69 CAN frames sent by 6 different nodes over a 250Kbit/s CAN bus. The total load of this message set is 60.25%. The offsets for the frame were computed using the SOA algorithm from NETCAR-Analyzer [15].

Figure 4 shows response times statistics for three configurations corresponding to the same clock drift parameters but different initial phasing configurations. The initial phasing and clock drift values were chosen randomly (clock drifts are in +/- 150 ppm range). The first thing to point out is that the maximum observed response times (the green curves) are slightly different depending on the initial configuration. This is especially true for the lowest priority frames (right-hand side of the graphic). As expected, the minimum response time curves are all equal because the phasings between sending nodes changes in such a way that every frame will be sent without delay at least once during the 24h. What is more surprising is that the average response times are equal and that the 99% and the 99.9% quantiles are almost identical. This is a meaningful result because it suggests that, for a long simulation time, the initial phasings are not significant for the final response time distributions statistics (except of course for the maximum observed response times).

4.3 Impact of the drift values

The aim here is to evaluate the impact of the clock drift values on the resulting frame response time statistics. Our first experiments showed similar results as seen with the initial phasings and differences were hardly noticeable. However, since using different drift values effectively affects the bus load, it should affect the response time distributions too. As a consequence, we tried to use a larger value range for the drifts : +/- 1000 ppm. We also picked values so as to impact the average load while staying within the tolerance bounds.

The figure 5 shows the resulting cumulated frame response time distributions of the lowest priority frame from the SAE benchmark for 24h simulation runs corresponding to different drift configurations. The simulation length is chosen large enough so that response time statistics have stabilized. The purple and green curves correspond to configurations with random drift value respectively in the range +/- 1ppm (almost no drift) and +/- 1000 ppm (high possible drift range). The blue and red curve correspond to configurations with close but different drift values (so that

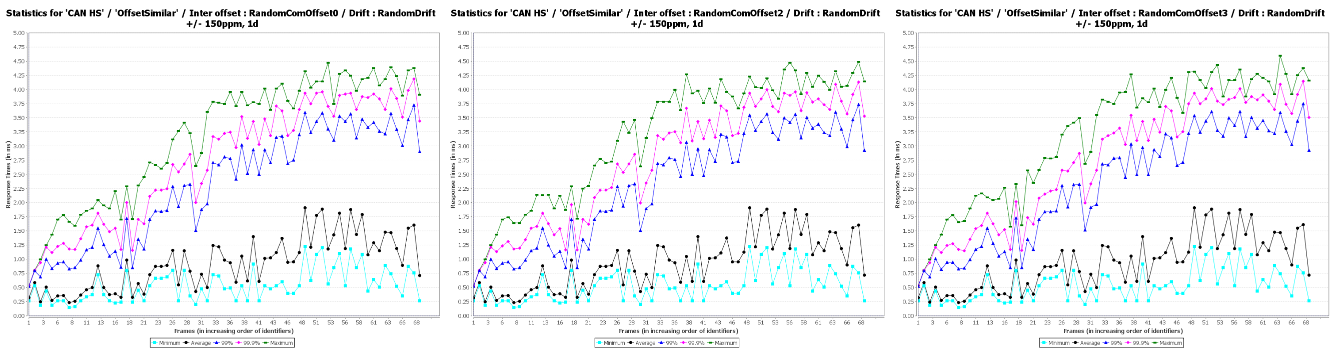

Figure 4: *Response time statistics of a CAN message set for three different initial phasings. The X-axis are the frames sorted by decreasing priorities (*i.e.*, increasing IDs) while the Y-axis shows the response times. The different curves correspond to different response time statistics. From the lowest to the highest curve: minimum, average, 99% quantile, 99.9% quantile and maximum response times.*

stations do not keep the same synchronization) that are chosen near the limit of the +/- 1000 ppm range (*e.g.*, *-1000 ppm, -999 ppm, -998 ppm*, etc, for the red curve and *+1000 ppm, +999 ppm, +998 ppm*, etc, for the blue curve).

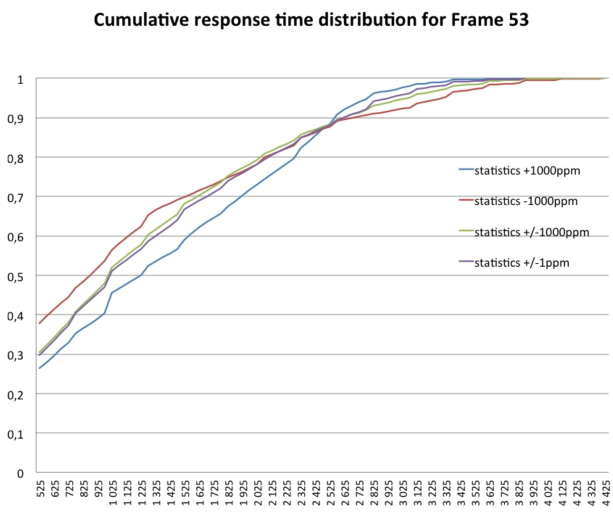

Figure 5: *Cumulative response time distributions of frame 49 of the SAE benchmark for different drift values recorded for a simulation length of two days. The X-axis is the response times in μs. The different curves correspond to different clock drift configurations.*

We clearly observe that the resulting frame response time distributions are different which indicates that the clock drift values do have an impact on the response times. Indeed, choosing clock values close to +1000 ppm and -1000 ppm makes a difference of 0,2% of the initial load which may have an impact on the response times of the lower priority frames. Because the load is lower with -1000 ppm, the red curve is above the others until 2,5 ms and below the others afterward because the response time distribution contains more lower response times. The blue curve (*i.e.*, -1000 ppm) is the exact opposite. However, the green and purple curve are close to each other as their average load are most likely close to each other because their drift values were chosen randomly around the same center value.

This experiment points out that the response times are ultimately impacted by the clock drifts. The next experiment aims at exploring how clock drifts affect practical industrial cases where the tolerated drift values are smaller.

4.4 Case study

This last experiment aims to evaluate the impact of clock drifts with sets of parameters corresponding to realistic cases. Since initial phasings depend of the relative start-up times of the nodes which cannot be predicted and since clock drifts are small, we study the relevance of a single simulation run to assess the expected performances of a CAN network.

In order to do that, we compare the distances between the resulting distributions for different simulations with random initial phasings and random drift values within pre-established drift ranges (respectively +/- 100 ppm and +/- 1000 ppm for the two sets of experiments). Each set of experiments correspond to 10 simulations of 48h of the SAE benchmark. We study and compare the response time distributions of the lowest priority frame at different times in the simulation. It is shown in Figure 6 how the distances between these simulations evolve by computing the standard deviations of the different response time intervals and compare their average. Here the response time range is subdivided into 100 intervals of equal size.

The graphic clearly shows that the standard deviations decrease significantly over time which indicates that the response time distributions are getting closer to each other. Another observation is that this happens faster with larger clock drift values as the set of experiments with +/- 1000 ppm give lower standard deviation values. In combination with the fact that each distribution stabilizes over time, these curves indicate that all distributions are getting close to some limit distribution as they stabilize whatever their starting point and clock drift values. It should be noted that the results are more clear-cut with higher priority frames for which standard deviations are even closer to zero.

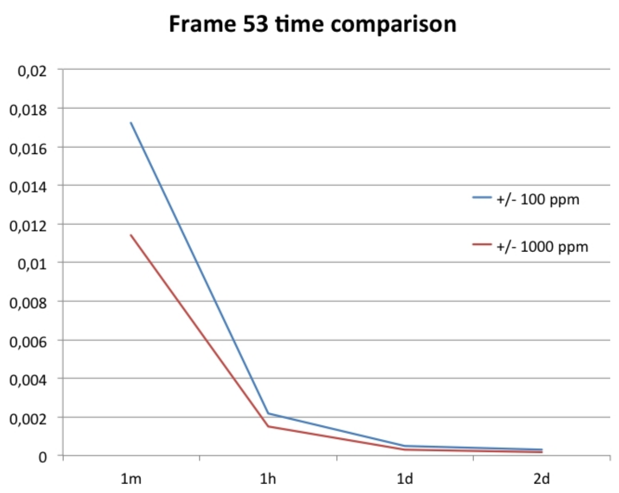

Figure 6: *Evolution over time of the standard deviation of the response time distributions for sets of 10 simulations with random initial phasings and drift values. The frame under study is the lowest priority frame of the SAE benchmark. The Y-axis correspond to the standard deviation value and the X-axis to the simulation time.*

This suggests that, assuming realistic clock drifts values, we can capture the behavior of a CAN network over a long time with a single limit distribution. This also suggests that a single long simulation run is sufficient to assess the expected behavior of a CAN network over time. Additionally, other experiments have shown that higher clock drifts help to stabilize the statistics faster which means that it can reduce the simulation times to obtain long-term CAN frame response time distributions. However, we have seen in section 3 that this simulation approach is not appropriate to capture corner cases: *i.e*, worst case scenario and other rare and large response times.

5 Conclusion

In this paper we present a simulation approach for CAN networks taking into account the clock drifts affecting the sending nodes which causes response times to vary over time. In section 3 of this paper we have seen experimentally that simulation and analysis alone are not enough. On the one hand, these experiments suggest that simulation alone is not appropriate to find the worst case scenarios because they are too rare. On the other hand, worst-case analysis cannot help to quantify how rare these events are, nor how long they last, nor what the average (or any other relevant statistics) of the response times are. However, it is possible to find by analysis the phasing conditions, specific to each frame, that provoke its worst case response time. Then, using a simulation tool, it becomes possible to observe for how long this situation lasts and where the clock drifts lead from there. Such simulations also contribute to validate the results obtained from the analysis tool, which is needed because these tools are usually black boxes and, though progresses are steadily being made [3, 4], have to make simplifications about the hardware and the communication stack [17].
We have then established that the frame response time distributions stabilize over time and that the initial phasings between the communicating nodes do not seem to have a significant impact on the resulting distributions. However, other experiments have shown that clock drifts may have an impact on the frame response time distributions when their values are far from realistic parameters. A noteworthy outcome of the last study is that, for typical ECU clock drift values, the response time distributions converge whatever the initial phasings between the sending nodes. This suggests that a single simulation is enough to capture the relevant statistics.
As future work, we plan to experiment with variable clock drifts (due for instance to temperature variation) which are already implemented in the simulation tool. As a result of using variable clock drift, the system will no longer be periodic meaning that the stabilization of the response time distribution is no longer a given.

Acknowledgement

We would like to thank Rob Davis (University of York) who helped us to find the experimental set-up showing the impact of clock drift in §4.3.

References

[1] K. Tindell, A. Burns, and A. J. Wellings, "Calculating Controller Area Network (CAN) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163 – 1169, 1995.

[2] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[3] D. Khan, R. Davis, and N. Navet, "Schedulability Analysis of CAN with Non-abortable Transmission Requests," in *16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*. Toulouse, France: IEEE, Sep. 2011.

[4] R. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Controller area network (can) schedulability analysis with fifo queues," in *23rd Euromicro Conference on Real-Time Systems (ECRTS)*, july 2011, pp. 45 –56.

[5] D. Khan, R. Bril, and N. Navet, "Integrating hardware limitations in CAN schedulability anal-

ysis," in *8th IEEE International Workshop on Factory Communication Systems (WFCS2010)*, 2010, pp. 207 –210.

[6] D. Khan, N. Navet, B. Bavoux, and J. Migge, "Aperiodic traffic in response time analyses with adjustable safety level," in *IEEE Conference on Emerging Technologies Factory Automation (ETFA 2009)*, 2009.

[7] N. Navet, Y. q. Song, and F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over CAN (Controller Area Network)," *J. Syst. Archit.*, vol. 46, pp. 607–617, 2000.

[8] M. Grenier, L. Havet, and N. Navet, "Pushing the limits of CAN - scheduling frames with offsets provides a major performance boost," in *European Congress of Embedded Real-Time Software (ERTS 2008)*, 2008.

[9] M. Nahas, M. J. Pont, and M. Short, "Reducing message-length variations in resource-constrained embedded systems implemented using the Controller Area Network (CAN) protocol," *J. Syst. Archit.*, vol. 55, pp. 344–354, May 2009.

[10] H. Zeng, M. D. Natale, P. Giusto, and A. L. Sangiovanni-Vincentelli, "Stochastic Analysis of CAN-Based Real-time Automotive Systems," *IEEE Trans. Industrial Informatics*, vol. 5, no. 4, pp. 388–401, 2009.

[11] ——, "Using statistical methods to compute the probability distribution of message response time in Controller Area Network," *IEEE Trans. Industrial Informatics*, vol. 5, no. 4, pp. 678–691, 2010.

[12] T. Nolte, H. Hansson, and C. Norström, "Probabilistic worst-case response-time analysis for the Controller Area Network," in *the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '03)*, 2003.

[13] M. De Biasi, C. Snickars, K. Landernas, and A. Isaksson, "Simulation of process control with WirelessHART networks subject to clock drift," in *32nd Annual IEEE International Computer Software and Applications (COMPSAC '08)*, 2008, pp. 1355 –1360.

[14] A. Monot, N. Navet, and B. Bavoux, "Impact of clock drifts on CAN frame response time distributions," in *16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, 2011.

[15] RealTime-at-Work, "NETCAR-Analyzer: Timing analysis and resource usage optimization for Controller Area Network," available at http://www.realtimeatwork.com/software/netcar-analyzer/, 2009.

[16] ——, "RTAW-Sim: Fine-grained simulation of Controller Area Network with fault-injection capabilities," available at http://www.realtimeatwork.com/software/rtaw-sim/, 2009.

[17] N. Navet, A. Monot, and J. Migge, "Frame latency evaluation: when simulation and analysis alone are not enough," 8th IEEE International Workshop on Factory Communication Systems (WFCS2010), Industry Day, 2010.