

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

Patrick Meumeu Yomsi, Dominique Bertrand
INRIA Nancy Grand-Est,
615, rue du Jardin Botanique,
54600 Villers-lès-Nancy (France),
firstname.name@inria.fr

Nicolas Navet
INRIA / RTaW
615, rue du Jardin Botanique,
54600 Villers-lès-Nancy (France),
nicolas.navet@inria.fr

Robert I. Davis
Department of Computer Science,
University of York,
YO10 5DD, York (UK),
rob.davis@cs.york.ac.uk

Abstract

Desynchronizing streams of frames through the means of offsets has today become common practice in automotive CAN networks. This is because this traffic shaping strategy is very beneficial in terms of reducing response times especially at high load levels. However, to the best of our knowledge, there is no result available in the literature that allows the response times of frames with offsets to be calculated for CAN. In this paper, we address this shortcoming of existing CAN schedulability analysis, and propose an extendible framework built upon the transaction model to derive worst-case response times (WCRT) on CAN. As will be shown in the experiments performed on realistic automotive networks, explicitly integrating offsets in the analysis permits a much tighter WCRT evaluation than with the classical synchronous analysis, which ultimately enables the designer to reduce resource over-provisioning.

1. Introduction

Problem definition. Early in the development cycle, schedulability analysis is along with simulation the main approach to verifying timing constraints and dimensioning a real-time system. The collection of schedulability analyses developed over the years for Controller Area Network (CAN) now constitutes a mature body of literature that allow the designer of CAN-based systems to obtain performance guarantees in most practical cases (see for instance [1, 2, 3, 4, 5]). Frames having offsets is one of the cases that is not handled by published analyses, though there are several commercial tools such as NETCAR-Analyzer [6] that have provided response time analysis with offsets for years. Further, many car manufacturers have used offsets for at least one car generation. This paper presents a basic CAN schedulability analysis

with offsets and thus, provides a contribution to the design of safe and optimized CAN-based communication systems.

Scheduling frames with offsets. Transmitting frames with offsets means that the first instance of a stream of periodic frames is released with a delay, called the offset, with regard to a reference point which is the first point in time at which the station becomes ready to transmit. Subsequent frames of the streams are then sent periodically, with the first transmission as the time origin. It should be pointed out that since there is no global synchronization among the stations on a CAN network, each station possesses its own local clock and the de-synchronization between the streams of frames remain local to each station¹. Offsets are efficient because they allow the workload to be spread over time and thus to reduce peak load. The reader may refer to [7] for explanations about why offsets are efficient, the description of a simple offset assignment algorithms, and experiments showing the gain one can expect on automotive networks.

Related work. The timing analysis of CAN has been extensively investigated in the past. Bounds on the worst case response times were first provided in [1], then revisited in [3]. Studies have integrated the limitations of hardware [8, 9, 10, 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 [11] and the consequences of transient perturbations [2]. Response times with offsets have also been studied but, in contrast to the contribution of this work, the literature focuses on approximate but lower-complexity forms of analyses, as first introduced by Tindell in [12]. In

¹In a time-triggered system, nodes would all be synchronized - offsets can be seen as a trade-off between time-triggered and fully event-triggered systems.

this thread of research, one can cite the studies published in [13, 14, 15].

Contribution of the paper. In this paper, we propose a framework built upon the *transaction* model to analyse the WCRT of CAN frames having offset relationships. This work adapts the exact schedulability analysis with offsets developed in the preemptive case by Palencia and Harbour [16] to the non-preemptive case where offsets are local to each sending node and work arrival functions are different because of the non-preemptive nature of the scheduling policy. Another contribution of this work is to analyze the case where jitter can be larger than the transmission period, which occurs in practice when frames are forwarded from one network to another through a gateway (i.e., the response time on the source network is part of the jitter on the destination network and may eventually exceed the transmission period). Finally, we evaluate, on realistic automotive networks, the gain of the analysis with offsets with regard to the simple synchronous analysis.

Paper organization The remainder of this paper is structured as follows. Section 2 describes the model of computation and the notation that will be used in the rest of the paper. This section also recalls the basic notions of transactions and presents the mapping of a set of frames transmitted by a given node into a single transaction. Section 3 presents the worst-case response time analysis for frames with offsets and jitter. Section 4 presents the experimental results. Finally, Section 5 concludes the paper and proposes future work.

2. Model of computation and assumptions

We consider a CAN Communication system \mathcal{N} to be a network which uses the CAN protocol for communication among the nodes. The network is composed of a set of nodes connected to a CAN bus. We model the network as an infinite non-preemptive priority queue where each node is inserting its frames according to their priorities. That is, we consider the ideal CAN behavior where the highest priority frame ready for transmission *always* wins arbitration. In this paper, the hardware limitations due to an insufficient number of transmit buffers [8], frame copy time [10], non-abortable transmission [5], FIFO queue [4] are not considered, but the analysis presented here can be extended to address these issues.

2.1. Frame model

Each node π_i transmits a set of n_i periodic frames $\mathcal{F}_{\pi_i} \stackrel{\text{def}}{=} \{\mathcal{F}_i^1, \mathcal{F}_i^2, \dots, \mathcal{F}_i^{n_i}\}$ to other nodes. Figure 1 illustrates a CAN Communication system. Each frame \mathcal{F}_i^k (with $1 \leq k \leq n_i$) is characterized by a tuple $(\tilde{\phi}_i^k, J_i^k, C_i^k, D_i^k, T_i^k, P_i^k)$ where each parameter is defined as follows:

- ▷ $\tilde{\phi}_i^k$ is the initial offset, i.e., the first release time of \mathcal{F}_i^k on node π_i after the node has started,
- ▷ J_i^k is the *maximum* jitter, i.e., any release of \mathcal{F}_i^k can be delayed by an arbitrary amount of time between 0 and J_i^k ,
- ▷ C_i^k is the worst-case transmission time (including maximal bit-stuffing and inter-frame space),
- ▷ D_i^k is the relative deadline (i.e., deadline relative to the activation),
- ▷ T_i^k is the transmission period,
- ▷ P_i^k is the priority of \mathcal{F}_i^k at the CAN MAC layer, assumed to be unique.

These parameters are given with the interpretation that frame \mathcal{F}_i^k generates and queues an infinite number of successive frame instances $f_i^{k,n}$ ($k = 1, 2, \dots, n_i$; $n = 1, 2, \dots, \infty$) with transmission time of at most C_i^k each, activated at time $\tilde{\phi}_i^k + (n-1) \cdot T_i^k$ and that must complete transmission at the latest at $\tilde{\phi}_i^k + (n-1) \cdot T_i^k + D_i^k$. In addition, the release of \mathcal{F}_i^k may have a maximum jitter equal to J_i^k which means that the actual release time of $f_i^{k,n}$ may occur at any time between $\tilde{\phi}_i^k + (n-1) \cdot T_i^k$ and $\tilde{\phi}_i^k + (n-1) \cdot T_i^k + J_i^k$. Hereafter, we assume that $\tilde{\phi}_i^k < T_i^k, \forall k \in [1, n_i]$, and without any loss of generality, we assume that the smallest offset is zero for each node π_i , i.e., $\min_k(\tilde{\phi}_i^k) \stackrel{\text{def}}{=} 0$.

2.2. Transaction model

In this subsection, we first remind the reader of the basic notions of transactions. Then, we show how transactions can be used for the modelling of the set of frames generated by a given node. Indeed, *all* the periodic frames generated by each node of the CAN Communication system can be gathered into a *single* transaction.

Definition 1 (Transaction) Let \mathcal{N} be a CAN Communication system and $\mathcal{F}_{\pi_i} = \{\mathcal{F}_i^1, \mathcal{F}_i^2, \dots, \mathcal{F}_i^{n_i}\}$ be a set of periodic frames generated by node π_i . A transaction Γ_i is defined as the subset of frame instances $\{f_i^{1,k_1}, f_i^{2,k_2}, \dots, f_i^{n_i,k_i}\}$ (with $k_1, k_2, \dots, k_i \in \mathbb{N}^+$) activated between 0 and the Least Common Multiple (lcm) of the periods of all frames in \mathcal{F}_{π_i} .

From Definition 1, it follows that all instances of a transaction are sent by the same node and possess offset relationships according to the characteristics of the frames. It should be noted that there is no offset relationship between transactions as nodes on CAN are not synchronized. Each transaction Γ_i is periodic with period $T_i \stackrel{\text{def}}{=} \text{lcm}\{T_i^1, T_i^2, \dots, T_i^{n_i}\}$, that is the lcm of the periods of all the frames belonging to Γ_i . We denote by $\Gamma_{i,n}$ the n^{th} release of transaction Γ_i . The j -th frame instance of \mathcal{F}_i^k in the first instance of Γ_i is characterized by an offset $\tilde{\phi}_i^k$ relative to the start of the transaction, an activation

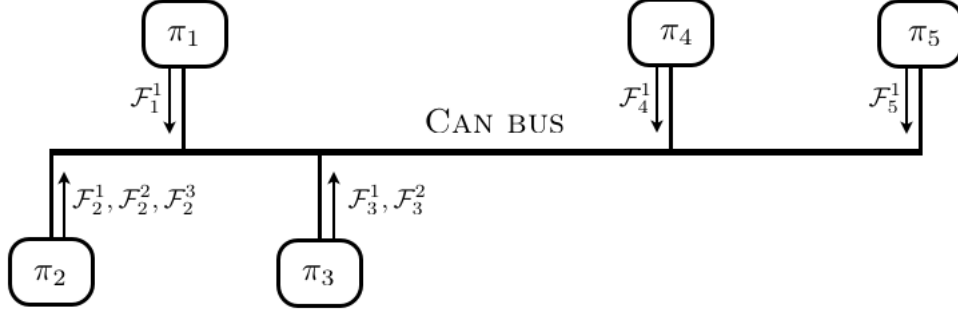


Figure 1. A CAN Communication system where 5 nodes are connected to a CAN bus. $\{\mathcal{F}_1^1, \mathcal{F}_1^2, \mathcal{F}_2^1, \mathcal{F}_2^2, \mathcal{F}_2^3, \mathcal{F}_3^1, \mathcal{F}_3^2, \mathcal{F}_4^1, \mathcal{F}_5^1\}$ is the set of frames transmitted on the network, and $\{\mathcal{F}_2^1, \mathcal{F}_2^2, \mathcal{F}_2^3\}$ is the set of frames generated by node π_2 .

time $\tilde{\phi}_i^k + (j-1) \cdot T_i^k$, a maximum jitter J_i^k , a worst-case transmission time C_i^k , a relative deadline D_i^k , and a priority P_i^k inherited from \mathcal{F}_i^k . By definition of the periodicity of a transaction, this frame instance is activated each T_i time units afterwards.

2.3. From frames to transactions

The framework developed in [16] for the schedulability analysis of periodic tasks with offsets provides us with a sound basis for analyzing a set of transactions in the non-preemptive case. The analysis in [16] cannot be used directly but we will build on it to present in Section 3 an analysis based on notations and concepts (e.g., occupied period) adapted to the non-preemptive case.

The modelling of the set of periodic frames generated by each node π_i into a *single* transaction Γ_i is performed by considering the release times of all the frame instances occurring during the lcm of the periods of all the frames generated by π_i , i.e., all the frame instances within $[0, T_i)$. This transformation is complete in the sense that there is no frame release that will be missed during the transformation into transactions and there will be the same number of instances during any time interval of length lcm as $\tilde{\phi}_i^k < T_i^k, \forall k \in [1, n_i]$. Note that we can generalize this model to handle sporadic² frames sent by each node π_i by making the distinction between periodic and sporadic frames, and then saying that all of the periodic frames of π_i are modelled as being part of a single transaction, whereas each sporadic frame is modelled as belonging to its own separate transaction, thus making it possible to apply the same analysis to all messages.

Figure 2 illustrates the notations used for frames generated by a given node on the one hand, and the mapping of these frames into a single transaction on the other hand: \mathcal{F}_1^1 and \mathcal{F}_1^2 are respectively defined by $(\tilde{\phi}_1^1 = 0, J_1^1 = 2, C_1^1 = 1, D_1^1 = 4, T_1^1 = 4, P_1^1 = 1)$ and $(\tilde{\phi}_1^2 = 1, J_1^2 = 7, C_1^2 = 2, D_1^2 = 12, T_1^2 = 6, P_1^2 = 2)$,

²A periodic frame is equivalent to a sporadic frame where the time elapsed between two successive releases, is always constant.

and are generated by node π_1 . Note that $\tilde{\phi}_1^1 = 0 < 4 = T_1^1$ and $\tilde{\phi}_1^2 = 1 < 6 = T_1^2$. The work arrival pattern is depicted up to the time instant 30.

Since $\tilde{\phi}_i^k < T_i^k$, the same instances are released in each interval of length lcm as in a single transaction Γ_i (with period T_i) corresponding to the node π_i . Thus, the work arrival function of the resulting transaction Γ_i and that of the set of periodic frames generated by π_i are the same.

3. Worst-Case Response Time Analysis

We recall that the response time of a frame instance is the time elapsed between its activation time and its reception by all the targeted nodes. Accordingly, the worst-case response time of a frame is the maximum amongst the response times of all the frame instances.

Before going any further in this paper, it should be pointed out that in general it is not possible to compute the worst-case response time of a frame by collecting the response times of all the possible trajectories of the system due to complexity reasons. The classical solution to get around this issue consists of: (i) “building” the smallest set of trajectories amongst which one can guarantee that the worst-case can be found, and (ii) analyzing all trajectories one after another in this set. In other words, this means analyzing time intervals with well-chosen frame activation times. For this purpose, we first provide the principle of our approach by giving the set of scenarios that must be considered. Then, we explain the computation of the worst-case response time for a specific scenario. In the remainder of the paper, we denote the frame under analysis by \mathcal{F}_u^a and we assume that it has been assigned priority p (i.e., ID p at the CAN MAC layer).

3.1. Set of scenarios

We adapt the Response Time Analysis (RTA) developed in [16] for preemptive tasks executed on a uniprocessor platform to the case of non preemptive frames trans-

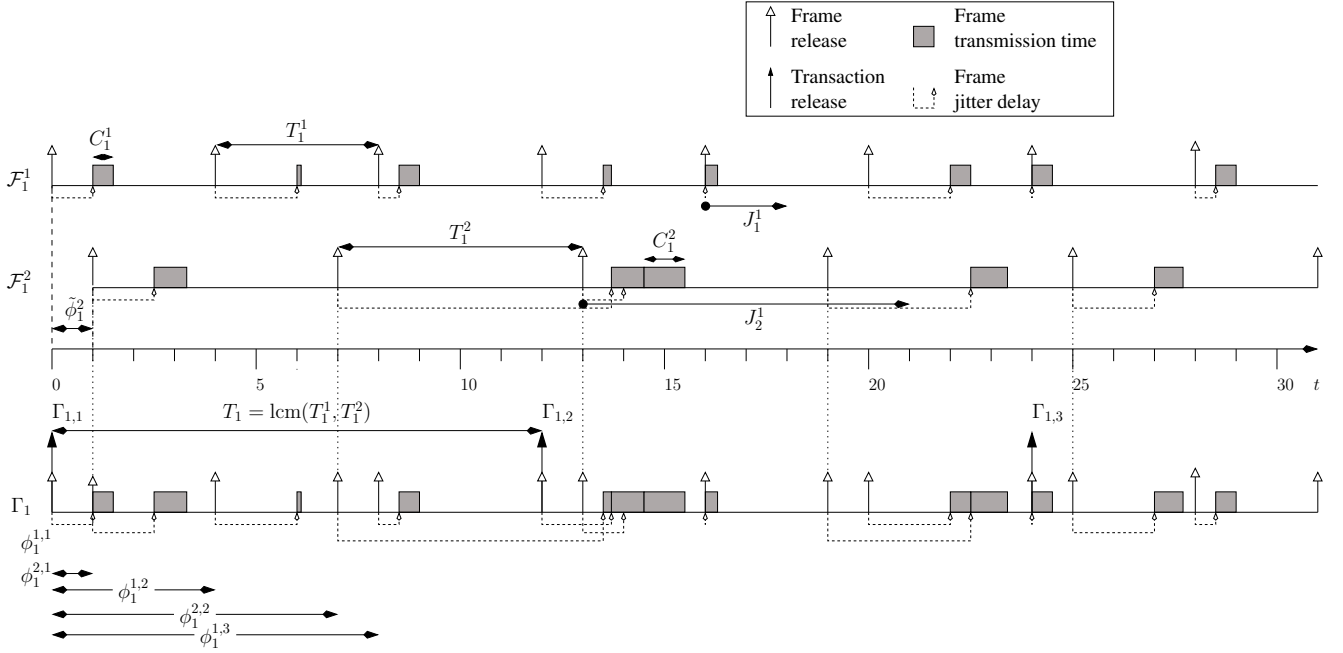


Figure 2. Workload induced by frames \mathcal{F}_1^1 and \mathcal{F}_1^2 generated by node π_1 , and mapping of these frames into a single transaction Γ_1 . The periods of frames \mathcal{F}_1^1 and \mathcal{F}_1^2 are respectively 4 and 6, the maximum jitters are 2 and 7 respectively, and the offsets are 0 and 1 respectively.

mitted on a CAN Communication system. This analysis relies on the concepts of a *level- p busy*, *idle* and *occupied periods*, as defined in [17] and adapted below to frames in Definitions 2, 3 and 5.

Definition 2 (Level- p idle instant) A level- p idle instant is defined as a time instant t at which there are no frames of priority higher than or equal to p , released strictly before t that have not completed transmission.

Definition 3 (Level- p busy period) A level- p busy period, denoted by L_p , is defined as a time interval $[t_1, t_2)$ which; (i) starts at a level- p idle instant t_1 , when a frame of priority p or higher becomes ready to be transmitted, (ii) is a contiguous interval of time during which any frame of priority lower than p is unable to start being transmitting, and (iii) ends at the first level- p idle instant t_2 after its start.

Definition 4 (Level- p idle period) A level- p idle period is defined as a time interval $[t_3, t_4)$ of length greater than zero, during which no frames are ready to be transmitted at priority p or higher, strictly before the end of the idle period at t_4 .

Definition 5 (Level- p occupied period) A level- p occupied period is defined as a time interval $[t_5, t_6)$ which; starts at the end of a level- p idle period and ends at the start of the next level- p idle period. During a level- p occupied period, the CAN bus is occupied transmitting frames at priority p or higher and is thus unable to start transmission of any frame at a priority lower than p .

Note that there exists a subtle difference between a busy period and an occupied period which should be clarified. This difference is explained as follows. A busy period ends once all the frames released strictly prior to the end of the busy period have completed transmission, irrespective of whether further frames are released at the end of the period; whereas, an occupied period only ends once all the frames released prior to, and including the end of the period have completed transmission. According to Definition 3, all the frames with a priority higher than or equal to p that have been released before the start time of the level- p busy period have been transmitted by the beginning of L_p , whereas those released during L_p will have been transmitted by the end of L_p . It thus follows that we are guaranteed to obtain the worst-case response time of each frame \mathcal{F}_u^a by analyzing all the possible level- p busy periods. In the following, a level- p busy period is defined by a *scenario*.

Definition 6 (Scenario) Let \mathcal{F} be the set of frames. A scenario \mathcal{S} is defined by the first release time of each frame in \mathcal{F} occurring at or after the beginning of a level- p busy period.

It should be noted that a scenario fully defines a level- p busy period. Since analyzing all the possible level- p busy periods is intractable due to the potentially huge number of level- p busy periods, we have to restrict ourselves to the “significant” scenarios. For the analysis of frame \mathcal{F}_u^a with the assigned priority p , Tindell proved in [18] that it is sufficient to restrict the analysis to the

subset of scenarios where at least one frame, with a priority higher than or equal to p , is released at the beginning of the level- p busy period in every transaction. For the sake of readability in the upcoming equations, we introduce the notions of critical instant, $hp_i(\mathcal{F}_u^a)$ and $lp_i(\mathcal{F}_u^a)$ in the following definitions.

Definition 7 (Critical instant) Let \mathcal{F}_u^a be a frame with priority p . The critical instant t_c for \mathcal{F}_u^a corresponds to the beginning of the level- p busy period in which \mathcal{F}_u^a is transmitted.

Definition 8 ($hp_i(\mathcal{F}_u^a)$ and $lp_i(\mathcal{F}_u^a)$) Let \mathcal{F}_i be a set of frames generated by node π_i . We define $hp_i(\mathcal{F}_u^a)$ and $lp_i(\mathcal{F}_u^a)$ as the set of frames generated by π_i with a priority higher than that of \mathcal{F}_u^a and the set of frames with a priority lower than that of \mathcal{F}_u^a , respectively.

According to these definitions, the main result of [16] can be adapted to traffic streams and transactions in the *non-preemptive* case as follows.

Theorem 1 (See [16]) Let \mathcal{F} be a set of frames such that $\mathcal{F}_u^a \in \mathcal{F}$. We assume that \mathcal{F}_u^a is assigned priority p . Then, the worst-case scenario for frame \mathcal{F}_u^a necessarily belongs to the set of scenarios $\mathcal{S}^{\text{worst}}$ in which a frame with a priority higher than or equal to p , in each transaction Γ_i , is released at a critical instant, after having experienced its maximum possible jitter.

Proof 1 The proof of this theorem is very similar to the one presented in [16] (Theorem 2) and we refer the interested reader to that paper. The basic idea of the proof consists of considering a scenario for frame \mathcal{F}_u^a in which no frame with a priority higher than or equal to p , in any transaction Γ_i , is released at a critical instant. Then cause the event arrivals of Γ_i to occur earlier while keeping the same release patterns for all its frame instances, until a frame with a priority higher than or equal to p is released at the critical instant. Then show that this process brings more frame instances into the level- p busy and occupied periods, thus increasing the chance of additional interference on the transmission of frame \mathcal{F}_u^a . In the same way, for the maximum possible jitter, the basic idea consists of moving back in time the event arrivals of Γ_i while increasing by the same amount of time the jitter delay of all the frame instances with a priority higher than or equal to p , so that all these frame instances continue to be released at the same time as before. Then perform this process until one or more of the releases that start the busy period have experienced their maximum jitter. Finally, show that the frame instances which previously occurred after the end of the busy period, now occur within the busy period. Due to space limitations, we do not repeat the complete proof here.

We are currently not aware of any technique for deciding *a priori* which frames in each transaction will lead

to the worst-case response time when performing the analysis of a particular frame. Consequently, the only remaining option consists of testing all the possible scenarios in the set $\mathcal{S}^{\text{worst}}$ identified by Theorem 1. The cardinality of $\mathcal{S}^{\text{worst}}$ is given by the following Expression (1).

$$\prod_{i \neq u} \left(\sum_{\mathcal{F}_i^k \in hp_i(\mathcal{F}_u^a)} \#\{\tau_i^{k,j}\} \right) \cdot \sum_{\mathcal{F}_u^k \in hp_u(\mathcal{F}_u^a) \cup \mathcal{F}_u^a} \#\{\tau_u^{k,j}\} \quad (1)$$

where $\#\{\tau_i^{k,j}\}$ denotes the number of frames brought by frame \mathcal{F}_i^k to transaction Γ_i . Since \mathcal{F}_i^k generates at most $\frac{T_i}{T_i^k}$ frames in transaction Γ_i (see Section 2.3), then it holds that:

$$\#\{\tau_i^{k,j}\} \leq \frac{T_i}{T_i^k}$$

Note that the quantity given by Expression (1) grows rapidly with the number of frame instances generated by the nodes. Thus in practice, the framework proposed in this paper applies to systems with a small number of generated frames and transactions with relatively small periods (i.e., sets of frames where the lcm is not prohibitively large). The optimization problem addressing the reduction of Expression (1) remains out of the scope of this paper and is left for future work.

3.2. Worst-Case Response Time analysis of frame \mathcal{F}_u^a under a particular scenario

Let \mathcal{S}^p be a particular scenario defining a level- p busy period for the worst-case response time analysis of frame \mathcal{F}_u^a with the assigned priority p . We denote by $\tau_u^{a,n}$ the n^{th} frame instance of \mathcal{F}_u^a occurring after the beginning of the level- p busy period. Note that $\tau_u^{a,n}$ will differ from $f_u^{a,n}$, defined in Section 2.1, as the level- p busy period will in general begin later than the offset $\tilde{\phi}_u^a$ of frame \mathcal{F}_u^a . Indeed, $\tau_u^{a,n}$ is the n^{th} frame instance of \mathcal{F}_u^a relative to the beginning of the level- p busy period whereas $f_u^{a,n}$ is the n^{th} frame instance of \mathcal{F}_u^a relative to its very first release time, i.e., at its offset $\tilde{\phi}_u^a$. Focusing on the level- p busy period, we know that the start time of the transmission of $\tau_u^{a,n}$ occurs when no frame with a priority higher than p is pending (we assume a FIFO ordering among all frames with priority p). Consequently, we first need to compute the interference due to the transmission of other frames on the transmission of $\tau_u^{a,n}$ in order to derive the worst-case response time of frame \mathcal{F}_u^a .

3.3. Interference from other frames

The sources of interference caused by the transmission of other frames on the response time of $\tau_u^{a,n}$ consists of the following contributions:

- ▷ The transmission of the instances of \mathcal{F}_u^a occurring prior to the release time of $\tau_u^{a,n}$ in the current level- p busy period.
- ▷ The transmission of a frame with a lower priority than p whose transmission can be ongoing at the start

time of the level- p busy period (called the *blocking factor*).

- ▷ The transmission of all the frames with a higher priority than p which are released since the beginning of the level- p busy period, some of them having been delayed by their maximum jitter [16].

The first two sources of interference are easily evaluated. Indeed, since $\tau_u^{a,1}$ is the first released frame instance of \mathcal{F}_u^a in the level- p busy period and $\tau_u^{a,n}$ is the n^{th} released instance, then the workload of the instances of \mathcal{F}_u^a occurring prior to the release time of $\tau_u^{a,n}$ in the non-preemptive case remains the same as in the preemptive case, i.e.,

$$(n-1) \cdot C_u^a \quad (2)$$

The blocking factor B_u^a due to the transmission of frames with a lower priority than p is defined by Equation (3).

$$B_u^a \stackrel{\text{def}}{=} \max_{\mathcal{F}_i^k \in lp_i(\mathcal{F}_u^a)} C_i^k \quad (3)$$

Note that this blocking factor is pessimistic and such blocking may not be able to occur at the same time as the maximum amount of interference due to the frames involved being in the same transaction. More precise analysis to refine this point is left for future work.

Now, it remains to determine the value of the maximum contribution due to the transmission of all frames with a higher priority than p . First, let ϕ^i denote the time interval between the latest release of transaction Γ_i that occurred prior to the critical instant t_c , and that critical instant in the scenario \mathcal{S}^p (see Figure 3 for illustration, in this case $\phi^1 = 8$ time units). Based on the results presented in [16] in the preemptive case, we recall that the contribution of a single frame $\mathcal{F}_i^k \in hp_i(\mathcal{F}_u^a)$ in scenario \mathcal{S}^p to the interference on the transmission of \mathcal{F}_u^a before the time instant $t_c + t$ with $t \geq 0$ is given by Equation (4) as follows.

$$W(\mathcal{F}_i^k, \phi^i, t) \stackrel{\text{def}}{=} M_i^k(t) \cdot C_i^k \quad (4)$$

where

$$M_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\substack{\text{max. number of instances} \\ \text{that may accumulate at } t_c}} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\substack{\text{max. number of instances} \\ \text{in } [t_c, t_c + t)}} \quad (5)$$

and $\varphi_i^k(\phi^i)$ is the length of the time interval between the critical instant t_c and the first release time of frame \mathcal{F}_i^k occurring at or after the critical instant (see Figure 3 for illustration, here $\phi^1 = 8$ and $\varphi_1^2(\phi^1) = 5$). A mathematical expression for the computation of $\varphi_i^k(\phi_i)$ is adapted from [16] as follows.

$$\varphi_i^k(\phi^i) \stackrel{\text{def}}{=} (T_i^k - (\phi^i - \tilde{\phi}_i^k) \bmod T_i^k) \bmod T_i^k \quad (6)$$

The adaptation of Equation (4) to the non-preemptive case is obtained by replacing $M_i^k(t)$ by $K_i^k(t)$ defined as follows.

$$K_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\substack{\text{max. number of instances} \\ \text{that may accumulate at } t_c}} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor + 1}_{\substack{\text{max. number of instances} \\ \text{in } [t_c, t_c + t)}} \quad (7)$$

In Equation (7), only the second factor has been modified due to the non-preemptive transmission of the frames. Indeed, once a frame has started its transmission upon the network, it cannot be preempted, i.e., interrupted during its transmission and resumed later. Therefore, we are interested in the release time of each frame. In Equation (7), the factor $\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor$ corresponds to the number of release times of frame \mathcal{F}_i^k within time interval $[t_c, t_c + t]$ and the factor (+1) corresponds to a release of frame \mathcal{F}_i^k that may occur at time instant $t_c + t$. Consequently, the maximum interference due to all frames of transaction Γ_i with a priority higher than p is given by Equation (8).

$$W_u^a(\Gamma_i, \phi^i, t) = \sum_{\mathcal{F}_i^k \in hp_i(\mathcal{F}_u^a)} W(\mathcal{F}_i^k, \phi^i, t) \quad (8)$$

where $W(\mathcal{F}_i^k, \phi^i, t) = K_i^k(t) \cdot C_i^k$ and $K_i^k(t)$ is given by Equation (7).

3.4. Worst-case response time of frame \mathcal{F}_u^a (R_u^a)

Let Γ denote the set of all transactions and let $\mathcal{S}^{\text{wc}} \in \mathcal{S}^{\text{worst}}$ denote the worst-case scenario for \mathcal{F}_u^a . In the scenario \mathcal{S}^{wc} one has a maximum contribution from all other frames in terms of the interference on the transmission of frame \mathcal{F}_u^a . By using the same approach as developed in [16] for the schedulability analysis of a preemptive periodic task set with offset constraints, we can derive the worst-case response time R_u^a of frame \mathcal{F}_u^a in scenario \mathcal{S}^{wc} by considering the maximum among the response times $R_u^{a,n}$ ($n \in \mathbb{N}^+$) of all the frame instances $\tau_u^{a,n}$ of \mathcal{F}_u^a within the level- p busy period.

Since the transmission of $\tau_u^{a,n}$ starts as soon as the blocking factor has ended and no frame with a priority higher than p is pending, then the response time of $\tau_u^{a,n}$ is easily obtained by deriving the start time of its transmission. For this purpose, we number the frame instances of \mathcal{F}_u^a using the letter ξ , with consecutive numbers ordered w.r.t. the release time that they would have with no jitter, i.e., $J_u^a = 0$. We assign the value $\xi \stackrel{\text{def}}{=} 1$ to the release of \mathcal{F}_u^a that occurs in the interval $(0, T_u^a]$, (0 being the beginning of the level- p busy period). Accordingly, the release that occurs in the interval $(T_u^a, 2 \cdot T_u^a]$ gets the value $\xi = 2$, and so on. In the same way, the release that would have occurred in the interval $(-T_u^a, 0]$ but was delayed to the critical instant gets the value $\xi = 0$, the one that occurred in the interval $(-2 \cdot T_u^a, -T_u^a]$ gets the value $\xi = -1$, and

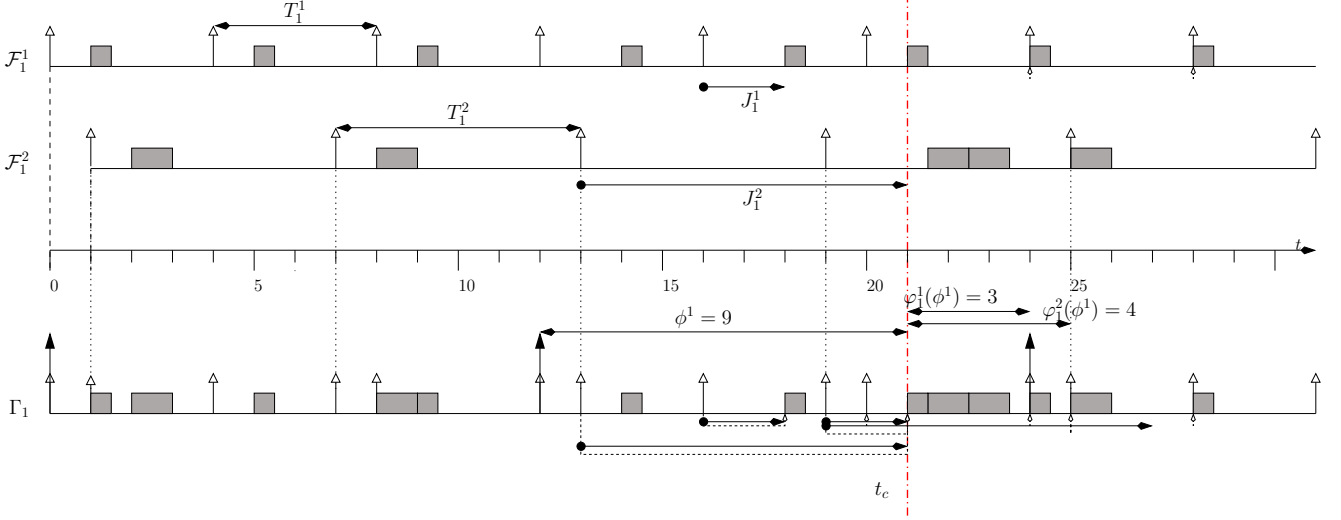


Figure 3. Values of time intervals ϕ^1 , $\varphi_1^1(\phi^1)$ for the frame \mathcal{F}_1^1 and $\varphi_1^2(\phi^1)$ for the frame \mathcal{F}_1^2 of transaction Γ_1 . Same set of frames as in Figure 1 but here the frames always request their worst-case transmission time.

so on. The start time $s_u^{a,n}$ of $\tau_u^{a,n}$ is obtained by solving Equation (9) where t is the variable.

$$t = B_u^a + (n - \xi_{u,0}^a) \cdot C_u^a + \sum_{\Gamma_i \in \Gamma} W_u^a(\Gamma_i, \phi^i, t) \quad (9)$$

In Equation (9), $\xi_{u,0}^a$ corresponds to the lowest-numbered frame instance of \mathcal{F}_u^a , i.e.,

$$\xi_{u,0}^a \stackrel{\text{def}}{=} - \left\lfloor \frac{J_u^a + \varphi_u^a(\phi^i)}{T_u^a} \right\rfloor + 1 \quad (10)$$

The length L_p of the level- p busy period is obtained by solving Equation (11) where t is the variable.

$$t = B_u^a + M_u^a(t) \cdot C_u^a + \sum_{\Gamma_i \in \Gamma, \mathcal{F}_i^k \in \text{hp}_i(\mathcal{F}_u^a)} M_i^k(t) \cdot C_i^k \quad (11)$$

In Equation (11), $M_u^a(t)$ and $M_i^k(t)$ are provided by Equation (5). Once L_p is computed, the maximum value of n , denoted by ξ_{u,L_p}^a , that we need to check can be derived as follows.

$$\xi_{u,L_p}^a \stackrel{\text{def}}{=} \left\lceil \frac{L_p - \varphi_u^a(\phi^i)}{T_u^a} \right\rceil \quad (12)$$

Note that Equations (9) and (11) are solved in an iterative manner by using a fixed-point algorithm until the same result is obtained in two successive iterations. Finally, the worst-case response time R_u^a of frame \mathcal{F}_u^a in scenario \mathcal{S}^{wc} is computed as follows.

$$R_u^a \stackrel{\text{def}}{=} \max \left\{ R_u^{a,n} : n = \xi_{u,0}^a, \dots, \xi_{u,L_p}^a \right\} \quad (13)$$

where

$$R_u^{a,n} = s_u^{a,n} + C_u^a - (\varphi_u^a(\phi^i) + (n - 1) \cdot T_u^a) \quad (14)$$

4. Experimental Results

In this section, we examine the performance gain that can be achieved using offsets. We study a set of CAN networks generated with NETCARBENCH [19], that correspond to medium-size automotive body networks. The WCRT algorithm that is presented in this paper is of exponential complexity and is not suited to analyzing larger networks (e.g., more than 12 nodes) and networks with less regular characteristics (e.g., random jitters and offsets). Compared to NETCAR-Analyzer V1.7.x [6] which implements undisclosed algorithms, the analysis of this paper is clearly much slower but it supports jitter. The reader interested in a more comprehensive study on the performance gain brought by offsets in the case where frames do not have jitter can refer to [7].

Each network studied runs at 250kbit/s and is made up of 8 to 10 nodes. The periods of the frames are chosen from the set $\{20, 50, 100, 200, 500\}$ and the size of the data payload in the frames is between 1 and 8 bytes. The priority assignment for the CAN identifier is Rate-Monotonic. The total load on each network is between 38% and 42%. As is often the case in automotive networks, we assume that there is one station (e.g., a gateway) that generates more load than the others. Here this station transmits 20% of the total traffic. The offset assignment algorithm that is used in this study is the implementation of the Dissimilar Offset Assignment (DOA) algorithm [20] that is available in NETCAR-Analyzer. It is known not to be the most efficient offset assignment algorithm when periods are not arbitrary, but we decided nevertheless to use it because it is published. The experiments were done with and without frame queuing jitter, in the former case 10% of the frames are assigned a random

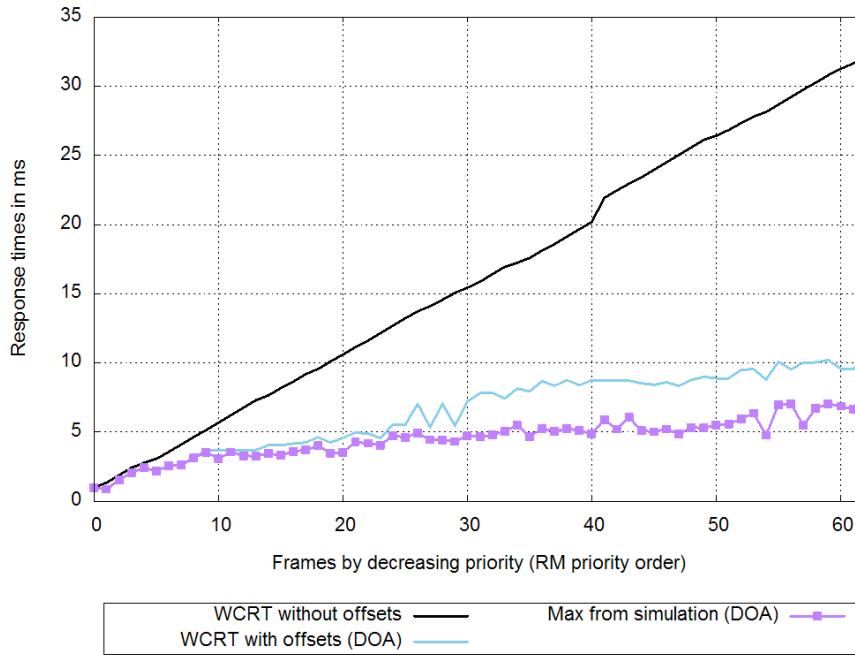


Figure 4. Worst-case response times of the CAN frames with and without offsets, along with the maximum values with offsets observed during a long simulation run. The steep increase of the WCRT at the end can be explained because several high priority frames have a period equal to 20ms, and two instances of these frames delay the lowest priority frames with a WCRT larger than 20ms.

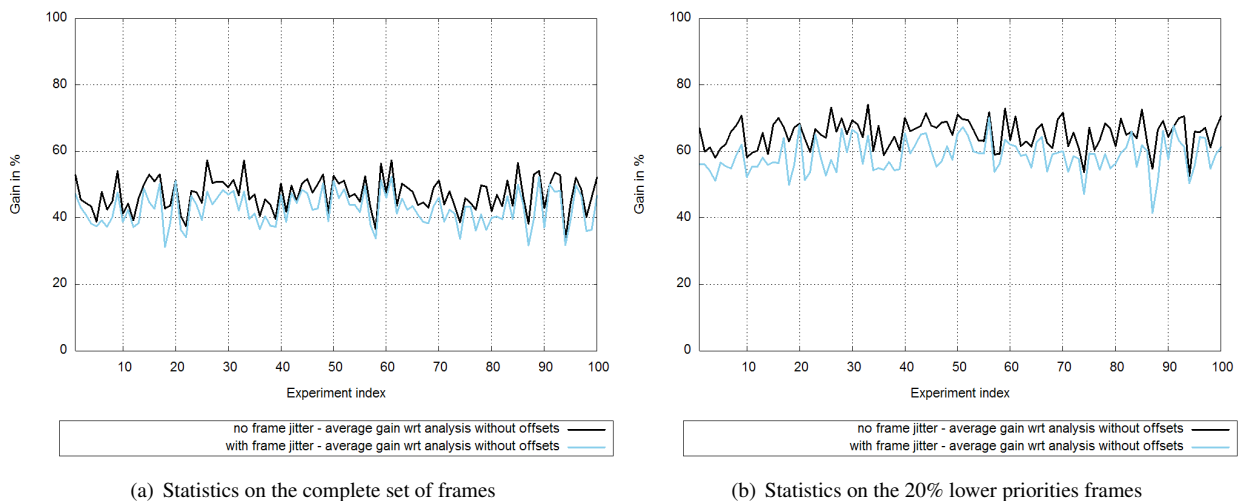


Figure 5. WCRT improvement by using offsets over 100 random configurations.

jitter in emission less than their period (integer values in ms). For the sake of reproducibility, the configuration file (with jitter) for NETCARBENCH V3.4 that is used for generating random networks is shown in Figure 6.

We first evaluate the performance gain with offsets on a single random configuration. As can be seen in Figure 4, the WCRT is improved for all frames but the improvements become more pronounced as the priority decreases. For the lowest priority frame of this example, the

WCRT with offsets is decreased by 22 ms, which represents a reduction of 68% compared to results without offsets. To get some insight in the possibility of determining the WCRT by simulation alone, we ran a long simulation run (corresponding to 7 days of functioning) with RTaW-Sim [21]. The difference between the maximum observed response times and WCRT with offsets is on average 25% and at most 45%. This suggests that WCRT analysis is really needed and confirms experiments presented in [22].


```

<netcarbench-data version="3.4" >
  <can-network name="body_config" granularity="5" bandwidth="250" >
    <network-load min="0.38" max="0.42" />
    <nb-network-interfaces min="8" max="12" />
    <fixed-station-loads>
      <station id="1" value="0.20" />
    </fixed-station-loads>
    <frame-periods>
      <period value="20" weight="2" margin="1" prio_low_range="1" prio_high_range="200"/>
      <period value="50" weight="5" margin="2" prio_low_range="201" prio_high_range="400"/>
      <period value="100" weight="10" margin="3" prio_low_range="401" prio_high_range="600"/>
      <period value="200" weight="5" margin="2" prio_low_range="601" prio_high_range="800"/>
      <period value="500" weight="1" margin="1" prio_low_range="801" prio_high_range="1000"/>
    </frame-periods>
    <frame-payloads>
      <payload value="1" weight="1" margin="1"/>
      <payload value="2" weight="1" margin="1"/>
      <payload value="3" weight="1" margin="1"/>
      <payload value="4" weight="2" margin="1"/>
      <payload value="5" weight="3" margin="2"/>
      <payload value="6" weight="4" margin="2"/>
      <payload value="7" weight="5" margin="2"/>
      <payload value="8" weight="6" margin="3"/>
    </frame-payloads>
    <jitters mode="RANDOM" ratio="0.1" />
  </can-network>
</netcarbench-data>

```

Figure 6. NETCARBENCH configuration file.

In the next series of experiments, we evaluated the performance of offset assignments over 100 random sets of messages. The performance metric used is the average WCRT reduction over the whole set of message (see Figure 5(a)) then over the 20% of the frames with the lowest priorities (see Figure 5(b)). This is done with and without jitter. We observe that the gain with offsets does depend on the specific configuration, but not to a large extent. The minimum, average and maximum gains without jitter, over the whole set of frames, are 33%, 47% and 57% respectively. With jitter, these values are 31%, 42% and 52% respectively. As already observed on the single experiment of Figure 4, the performance improvement of using offsets is larger for lower priority frames. Indeed, the minimum, average and maximum gains without jitter, over the 20% of the frames with the lowest priorities, are 53%, 65% and 74% respectively. With jitter, these values become 41%, 59% and 70%.

5. Conclusion and Future Work

The main contribution of this paper is an effective worst-case response time analysis³ in the non-preemptive case with offsets and jitters. Though the algorithm is of exponential complexity, experiments show that it can be used on medium-size networks with automotive characteristics in terms of periods, payloads and offsets. Future work will be devoted to the optimization of the algorithm for instance by identifying dominance relationships between scenarios, that would allow us to discard some of them. Also, along the lines of [13, 14, 15], we would like to develop and experiment with approximate analysis of lower algorithmic complexity. The analysis developed here assumes an ideal CAN network. However, in practice

³It is only the potentially pessimistic treatment of blocking that means the analysis presented is not exact.

many systems do not match this assumption because of a limited number of transmission buffers [8], delays in re-filling the transmit buffers [10], FIFO queues [4] and non-abortable transmit buffers [5]. This response time analysis remains to be extended to handle these cases.

Acknowledgments. The authors would like to thank Mathieu Grenier (INRIA/INPL) and Lionel Havet (INRIA) for their contribution. This work was partially funded by the UK EPSRC funded Tempo project (EP/G055548/1), the EU funded ArtistDesign Network of Excellence.

References

- [1] K. Tindell, H. Hansson, and A. Wellings, "Analyzing Real-Time Communications: Controller Area Network (CAN)", in *Proc. of the 15th IEEE Real-Time Systems Symposium (RTSS 1994)*, 1994.
- [2] N. Navet, Y.-Q. Song, and F. Simonot, "Worst-Case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network)", *Journal of Systems Architecture*, vol. 46, no. 7, pp. 607–618, 2000.
- [3] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refined, revisited and revised", *Real-Time Systems*, vol. 35, pp. 239–272, April 2007.
- [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)*, 5-8th July 2011, pp. 45–56.
- [5] D. A. Khan, R. I. Davis, and N. Navet, "Schedulability Analysis of CAN with Non-abortable Transmission Requests", in *Proc. of the 16th IEEE International Conference on Emerging Technologies and Factory Automation*, September 2011.
- [6] RealTime-at-Work, "NETCAR-Analyzer: Timing Analysis and Resource Usage Optimization for Controller Area Network", Available

- from <http://www.realtimetrwork.com/software/netcar-analyzer/>, 2009.
- [7] M. Grenier, L. Havet, and N. Navet, "Pushing the limits of CAN-Scheduling frames with offsets provides a major performance boost", in *Proc. of the 4th European Congress Embedded Real Time Software*, January 2008.
- [8] A. Meschi, M. D. Natale, and M. Spuri, "Priority Inversion at the Network Adapter when Scheduling Messages with Earliest Deadline Techniques", in *8th Euromicro Workshop on Real-Time Systems*, June 1996, pp. 243–248.
- [9] M. D. Natale, "Evaluating message transmission times in Controller Area Networks without buffer preemption", in *8th Brazilian Workshop on Real-Time Systems*, 2006.
- [10] D. Khan, R. Bril, and N. Navet, "Integrating hardware limitations in CAN schedulability analysis", in *8th IEEE International Workshop on Factory Communication Systems (WFCS2010)*, 2010, pp. 207–210.
- [11] 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.
- [12] K. Tindell, "Adding Time-Offsets to Schedulability Analysis", Technical report, University of York, Computer Science Dept, 1994.
- [13] A. Szakaly, "Response time analysis with offsets for CAN", Master's thesis, Department of Computer Engineering, Chalmers University of Technology, November 2003.
- [14] L. Du and G. Xu, "Worst Case Response time analysis for CAN messages with offsets", in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, November 2009, pp. 41–45.
- [15] Y. Chen, R. Kurachi, G. Zeng, and H. Takada, "Schedulability Comparison for CAN Message with Offset : Priority Queue Versus FIFO Queue", in *Proc. of the 19th International Conference on Real-Time and Network Systems (RTNS 2011)*, September 2011.
- [16] 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.
- [17] R. Davis and A. Burns, "Response Time Upper Bounds for Fixed Priority Real-Time Systems", in *Real-Time Systems Symposium, 2008*, 30 2008–dec. 3 2008, pp. 407–418.
- [18] K. Tindell, "Using Offset Information to Analyse Static Priority Pre-Emptively Scheduled Task Sets", Technical report, University of York, Computer Science Dept, 1994.
- [19] C. Braun, L. Havet, and N. Navet, "NETCARBENCH: a benchmark for techniques and tools used in the design of automotive communication systems", in *7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, 2007, pp. 321–328, Available at <http://www.netcarbench.org>.
- [20] M. Grenier, J. Goossens, and N. Navet, "Near-Optimal Fixed Priority Preemptive Scheduling of Offset Free Systems", in *Proc. of the 14th International Conference on Network and Systems (RTNS'2006)*, Mai 2006.
- [21] RealTime-at-Work, "RTAW-Sim: Fine-Grained Simulation of Controller Area Network with Fault-Injection Capabilities", available at <http://www.realtimetrwork.com/software/rtaw-sim/>, 2012.
- [22] 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.