A Two-Competitive Approximate Schedulability Analysis of CAN

Björn Andersson, Nuno Pereira, Eduardo Tovar
IPP Hurray Research Group
Polytechnic Institute of Porto, Portugal
{bandersson,npereira,emt}@dei.isep.ipp.pt

Abstract

Consider the problem of deciding whether a set of n sporadic message streams meet deadlines on a Controller Area Network (CAN) bus for a specified priority assignment. It is assumed that message streams have implicit deadlines and no release jitter. An algorithm to solve this problem is well known but unfortunately it time complexity is non-polynomial. We present an algorithm with polynomial time-complexity for computing an upper bound on the response times. Clearly, if the upper bound on the response time does not exceed the deadline then all deadlines are met. The pessimism of our approach is proven: if the upper bound of the response time exceeds the deadline then the response time exceeds the deadline as well for a CAN network with half the speed.

1. Introduction

Consider the problem of deciding whether a set of *n* sporadic message streams meet deadlines when scheduled on a CAN bus. A message stream τ_i generates a (potentially infinite) sequence of messages. The time when these messages arrive cannot be controlled by the CAN bus. It is assumed that the time between two consecutive arrivals of messages from the same message stream τ_i is at least T_i. A message from message stream τ_i has a message transmission time C_i; this includes the frame header and the arbitration for the CAN bus. If a message finishes transmission at most T_i time units after its arrival then we say that the message meets its deadline; otherwise it misses its deadline. It is assumed that $0 < C_i$ and $0 < T_i$, and that T_i and C_i are multiples of t_{bit} where t_{bit} denotes the time duration of a single bit in the CAN bus.

The arbitration protocol in CAN implements non-preemptive static-priority scheduling using the dominance protocol [1] which offers a very large number of priority levels. We assume that each message stream is assigned a unique priority and each message contends for the channel with the priority given by the message stream it belongs to.

A large amount of research has been performed on the CAN bus. Unfortunately, it has taken a long time to design a correct exact analysis [2]; it was presented just recently, in the year of 2007. Its time-complexity is currently unknown.

We present an algorithm with polynomial time-complexity for computing an upper bound on the response times. Clearly, if the upper bound on the response time does not exceed the deadline then all deadlines are met. The pessimism of our approach is proven: if the upper bound of the response time exceeds the deadline then the response-time exceeds the deadline as well for a CAN network with half the speed.

Section 2 gives a background on the CAN bus and in particular it presents a method that decides if and only if a set of message streams meets deadlines. Section 3 presents our new approximate schedulability analysis. Section 4 gives conclusions and future work.

2. Background on CAN Analysis

As already pointed out, CAN implements non-preemptive static-priority scheduling, and as a consequence, its timing behavior can be understood by analyzing non-preemptive static-priority scheduling. It is known [2, 3] that the response-time R_i of a message stream τ_i is can be correctly computed by the following steps. Compute B_i:

$$B_i = \max_{k \in lp(i)} C_k \quad (1)$$

where $lp(i)$ is the set of message streams with a lower priority than τ_i.

The length of the level-*i* busy period is computed as follows:

$$t_i^0 = C_i \quad (2)$$

and iterate according to:

$$t_i^{k+1} = B_i + \sum_{\forall j \in hep(i)} \left\lceil \frac{t_i^k}{T_j} \right\rceil \times C_j \quad (3)$$

where *hep*(*i*) is the set of message streams with a priority higher than or equal to the priority of τ_i. When (3) converges with $t_i^{k+1} = t_i^k$ then this is the value of t_i.

We can now compute $w_i(q)$, the queuing time of a the q+1:th message from τ_i in the level-*i* busy period. It is computed iteratively until the we obtain convergence, $w_i^{k+1}(q)=w_i^{k}(q)$ for the following iterative procedure:

$$w_i^0(q) = B_i + q \times C_i \tag{4}$$

and iterate according to:

$$w_i^{k+1}(q) = B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i^k(q) + t_{bit}}{T_j} \right\rceil \times C_j \tag{5}$$

where *hp*(*i*) is the set of message streams with a priority higher than or equal to the priority of τ_i.
When (5) converges with $w_i^{k+1}(q)=w_i^k(q)$ then this is the value of $w_i(q)$. We compute the response-time for the q+1:th message from τ_i in the level-*i* busy period as:

$$R_i(q) = w_i(q) - q \times T_i + C_i \tag{6}$$

This is used to calculate the response time:

$$R_i = \max_{q=0..Q_i-1} R_i(q) \tag{7}$$

where Q_i is defined as:

$$Q_i = \left\lceil \frac{t_i}{T_i} \right\rceil$$

This analysis works for the case that

$$\sum_{i=1}^{n} \frac{C_i}{T_i} < 1 \tag{8}$$

and hence we will make this assumption in the remainder of this paper.

This exact schedulability analysis for CAN has been proposed recently (January 2007) and the time-complexity of this procedure is currently not characterized. However, it generally accepted that it is non-polynomial since for message streams with large utilization it appears to be necessary to explore a busy period that has the length of the least-common multiple of T_i. For this reason, we will (in Section 3) propose a faster approximate method; it errs on the safe side though.

3. The New Analysis Method

It can be seen in (1)-(7) that the source of the large time-complexity is that (i) many values of q must be explored and (ii) many iterations needs to be made for computing $R_i(q)$. We will now simplify these aspects. This gives us a new analysis method and we will prove its performance.

Consider Equation 5. For one of its terms it holds that:

$$B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i^k(q) + t_{bit}}{T_j} \right\rceil \times C_j \leq B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left(\frac{w_i^k(q) + t_{bit}}{T_j} + 1 \right) \times C_j \tag{9}$$

Based on this rewriting we define wUB_i as being the minimum number that satisfies:

$$wUB_i = B_i + \sum_{\forall j \in hp(i)} \left(\frac{wUB_i + t_{bit}}{T_j} + 1 \right) \times C_j \tag{10}$$

Observe that (10) is a simplification of (5) in two ways (i) Inequality (10) considers the case that q=0 and (ii) an upper bound on the ceiling function is used. Nonetheless (10) is useful to find an upper bound on the response-time of message streams. Lemma 1 is instrumental for that.

Lemma 1. Let q denote an integer. Then it holds that:

$$\forall q \geq 0 : w_i(q) - q \times T_i \leq wUB_i$$

Proof: Suppose that the lemma was false. Then there must exist a set of message streams for which it is possible to select i and q such that:

$$w_i(q) - q \times T_i > wUB_i \tag{11}$$

Let $\Delta_i(q)$ be defined as:

$$\Delta_i(q) \mathrel{\hat{=}} w_i(q) - q \times T_i - wUB_i \tag{12}$$

It is easy to see from (12) that:

$$\Delta_i(q) > 0 \tag{13}$$

Using (5) and the fact that $w_i(q)$ is obtained when they converge and apply it to on (12) yields:

$$B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j - q \times T_i - wUB_i > 0$$

Using (10) yields:

$$
\begin{aligned}
B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- q \times T_i - B_i \\
- \sum_{\forall j \in hp(i)} \left(\frac{wUB_i + t_{bit}}{T_j} + 1 \right) \times C_j \\
> 0
\end{aligned}
\tag{14}
$$

Using (12) yields:

$$
\begin{aligned}
B_i + q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- q \times T_i - B_i \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - q \times T_i - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> 0
\end{aligned}
$$

Cancelling out the terms B_i yields:

$$
\begin{aligned}
q \times C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- q \times T_i \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - q \times T_i - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> 0
\end{aligned}
$$

We can rewrite it into:

$$
\begin{aligned}
\sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - q \times T_i - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> q \times T_i - q \times C_i
\end{aligned}
$$

We can rewrite it further into:

$$
\begin{aligned}
\sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> q \times T_i - q \times C_i - \sum_{\forall j \in hp(i)} \left(\frac{q \times T_i}{T_j} \right) \times C_j
\end{aligned}
$$

Further rewriting the right-hand side yields:

$$
\begin{aligned}
\sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> q \times T_i \times \left(1 - \left(\frac{C_i}{T_i} + \sum_{\forall j \in hp(i)} \frac{C_j}{T_j} \right) \right)
\end{aligned}
$$

Using the fact that $q \geq 0$ and (8) yields:

$$
\begin{aligned}
\sum_{\forall j \in hp(i)} \left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil \times C_j \\
- \sum_{\forall j \in hp(i)} \left(\frac{w_i(q) - \Delta_i(q) + t_{bit}}{T_j} + 1 \right) \times C_j \\
> 0
\end{aligned}
\tag{15}
$$

From (15) we obtain that there must be a j such that:

$$
\begin{aligned}
\left\lceil \frac{w_i(q) + t_{bit}}{T_j} \right\rceil > \\
\left(\frac{w_i(q) - \Delta_i(q) + t_{bit}}{T_j} + 1 \right)
\end{aligned}
\tag{16}
$$

But this is impossible. Hence the lemma is correct. □

We can now rewrite Equation (10) into:

$$
wUB_i = \frac{B_i + \sum_{\forall j \in hp(i)} \left(\frac{t_{bit}}{T_j} + 1 \right) \times C_j}{1 - \sum_{\forall j \in hp(i)} \frac{C_j}{T_j}}
\tag{17}
$$

Combining Lemma 1 and (6) and (7) yields:

$$
R_i(q) \leq wUB_i + C_i
\tag{18}
$$

Based on (18) we can construct an algorithm for computing an upper bound on the response-time. Theorem 1 does that.

Theorem 1. Compute wUB_i as:

$$
wUB_i = \frac{B_i + \sum_{\forall j \in hp(i)} \left(\frac{t_{bit}}{T_j} + 1 \right) \times C_j}{1 - \sum_{\forall j \in hp(i)} \frac{C_j}{T_j}}
$$

then it holds that $R_i \leq wUB_i + C_i$.
Proof: Follows from the discussion above. □

The algorithm presented in Theorem 1 provides only an upper bound on the response-time; it is not tight. The following examples illustrate the performance of the algorithm.

Example 1. Consider three message streams τ_1, τ_2 and τ_3 to be scheduled on a CAN bus with speed 1 Mbit/s. We measure all time units in µs. Consequently we obtain t_{bit}=1. Message streams are characterized as T_1=200, C_1=90, T_2=300, C_2=90, T_3=400, C_3=90.

With the exact analysis (1)-(7) we obtain R_1=180, R_2=270, R_3=270. But with the analysis of Theorem 1 we obtain: $R_1 \leq 180$, $R_2 \leq 418$, $R_3 \leq 813$. □

The previous example show that the schedulability analysis used in Theorem 1 is not tight. It then natural to ask how much pessimism it causes. Theorem 2 answers that question.

Theorem 2. Compute wUB_i as:

$$wUB_i = \frac{B_i + \sum_{\forall j \in hp(i)} \left(\frac{t_{bit}}{T_j} + 1 \right) \times C_j}{1 - \sum_{\forall j \in hp(i)} \frac{C_j}{T_j}}$$

If there is an i such that $wUB_i + C_i > T_i$ then there is a message stream that misses a deadline if the CAN bus is given half its speed.

Proof: If the theorem was false then there must be a set *TF* of message streams such that $wUB_i + C_i > T_i$ but when the set of message streams is scheduled on a CAN bus with half the speed it met their deadlines. Since the message streams meet deadlines on a CAN bus of half the speed, it must hold for every $q \geq 0$ that the value of convergence $w_i(q)$ that is obtained from (5) - $q \times T_i + 2 \times C_i$ does not exceed T_i. In particular, this must hold for q=0. Let t denote the convergent value for q=0. We have:

$$t = 2B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{t + 2 \times t_{bit}}{T_j} \right\rceil \times 2C_j \tag{19}$$

where

$$t + 2C_i \leq T_i \tag{20}$$

Let us consider what happens if we insert this value of t in (10). Since $wUB_i + C_i > T_i$, know that

$$t \neq B_i + \sum_{\forall j \in hp(i)} \left(\frac{t + t_{bit}}{T_j} + 1 \right) \times C_j \tag{21}$$

Let t' denote the value of t in (21). If the left-hand side of (21) would be greater than the right-hand side of (21) then we can decrease t' until the left-hand side of (21) becomes equal to the right-hand side of (21) and then value of t' can be used as wUB in (10) and it would satisfy $wUB_i + C_i \leq T_i$, which is impossible. This reasoning can be performed because we assume that the utilization is strictly less than 100% (as expressed by (8). Because of this reasoning, we know that t must satisfy:

$$t < B_i + \sum_{\forall j \in hp(i)} \left(\frac{t + t_{bit}}{T_j} + 1 \right) \times C_j \tag{22}$$

We know that for an arbitrary real value x>0, it holds that:

$$x + 1 < \lceil x \rceil \times 2 \tag{23}$$

Applying (23) on (22) yields:

$$t < B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{t + t_{bit}}{T_j} \right\rceil \times 2C_j \tag{24}$$

Combining (24) with (19) yields:

$$2B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{t + 2 \times t_{bit}}{T_j} \right\rceil \times 2C_j < B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{t + t_{bit}}{T_j} \right\rceil \times 2C_j \tag{25}$$

But (25) is impossible. Hence the theorem is correct. □

4. Conclusions and Future work

We have proposed a polynomial-time algorithm for deciding if a set of message streams meet their deadline when scheduled on CAN. This algorithm offers a sufficient test and the pessimism is quantified. The question whether it is possible to design an algorithm with a polynomial time-complexity and with a competitive ratio that is less than two was left open.

Acknowledgements

We are grateful to the reviewers for suggested improvements of the paper. This work was partially funded by the Portuguese Science and Technology Foundation (Fundação para Ciência e Tecnologia - FCT) and the ARTIST2 Network of Excellence on Embedded Systems Design.

References

[1] A. Mok, "Fundamental Design Problems of Distributed Systems for the Hard Real-Time Environment", PhD thesis, MIT, Cambridge, Mass., 1983. http://www.lcs.mit.edu/publications/specpub.php?id=865.

[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 Syst.*, vol. 35, pp. 239-272, 2007.

[3] L. George, N. Rivierre, and M. Spuri, "Preemptive and Non-Preemptive Real-Time UniProcessor Scheduling", INRIA, Technical Report RR-2966, September 1996.