Evaluation of a Hybrid Real-time Bus Scheduling Mechanism for CAN

Mohammad Ali Livani, Jörg Kaiser
University of Ulm, Department of Computer Structures, D-89069 Ulm, Germany
{Mohammad, Kaiser}@informatik.uni-ulm.de

Abstract. As a common resource, the CAN bus has to be shared by all computing nodes. Access to the bus has to be scheduled in a way that distributed computations meet their deadlines in spite of competition for the communication medium. The paper presents an evaluation of a resource scheduling scheme for CAN, which is based on time-slot reservation and dynamic priorities. The processing overhead as well as the schedulability parameters of the hybrid bus scheduling scheme are analyzed and compared with TDMA and the fixed priority assignment.

1 Introduction

The Controller Area Network (CAN) [8] is a field bus widely applied in industrial and automotive distributed real-time systems. One of the central issues in distributed real-time systems is scheduling the access to the communication medium so that distributed computations meet their deadlines. There exist several alternative approaches to solve this problem in CAN, which commonly exploit the priority-based bus arbitration mechanism of the CAN protocol[1].

The deadline-monotonic priority assignment [10] achieves meeting deadlines as ensured by an off-line feasibility test for a static system with periodic and sporadic tasks. By applying the dual priority scheme [4] a more flexible scheduling of hard and soft real-time communication is possible. The fixed priority assignment has been applied in the most common CAN-based communication systems implicitly, e.g. CAL [1], SDS [3], and DeviceNet [7]. The CANopen standard [2] defines a periodic communication scheme which is coordinated by a certain node (the SYNC master). Obviously, the SYNC master constitutes a single point of failure for the whole system. In [12] a combination of fixed and dynamic priority scheduling is approached. But this approach fails to schedule messages in a bus with 3 or more sender nodes due to a too short *time horizon*. The term *time horizon* is explained later in the paper.

In this paper, we first give a short presentation of a hybrid bus scheduling algorithm that combines mechanisms of TDMA and dynamic priority scheduling. A detailed description of the algorithm is given in [6]. In the rest of the paper the performance of the algorithm is evaluated. The evaluation consists of an analysis of

[1] Due to this arbitration technique, the identifier of a CAN-message serves as its priority (lower value = higher priority), and the bus itself acts as a priority-based dispatcher. The CAN arbitration mechanism requires that different nodes never start simultaneously sending messages with equal identifiers. This requirement must be satisfied by higher-level protocols.

the hybrid mechanism, and a comparison of the real-time communication schedulability under this scheme, TDMA, and the Deadline-Monotonic scheme.

2 The Hybrid Scheduling Mechanism

The bus scheduling scheme presented in this paper, is a combination of TDMA, dynamic priority, and user-defined fixed priority assignment. Unlike traditional TDMA schemes, our TDMA protocol uses dynamic priorities to ensure exclusive access rights of a sender during a reserved time-slot.

2.1 The Dynamic Priority Scheme

The dynamic priority scheme of the hybrid bus scheduling algorithm implements the Least-Laxity-First Scheme. Since CAN messages are non-preemptive, and their length varies in the same order of magnitude, the LLF scheme achieves a schedulability comparable with the Earliest-Deadline-First (EDF) scheduling, which is known to be optimal for the scheduling of a single resource. We chose LLF instead of EDF for its simpler implementation. In order to realize LLF in CAN, a mapping of the transmission laxity into the message priority has to be defined, such that the message with a shortest laxity gains the highest priority. Since the CAN identifier must provide uniqueness (cf. Footnote 1 on page 1) and information about the message subject, our algorithm uses only the most significant byte of the identifier as message priority.

Having a range $\{P_{min} .. P_{max}\}$ for the priority field, a laxity ΔL is mapped to a priority P, where $P = \lfloor \Delta L / \Delta tp \rfloor + P_{\min}$ if $\Delta L < (P_{max}–P_{min})*\Delta tp$ and $P = P_{max}$ if $\Delta L \geq (P_{max}–P_{min})*\Delta tp$. The period Δtp is called the priority slot. Since any laxity value $\Delta L \geq (P_{max}–P_{min})*\Delta tp$ is mapped to P_{max}, the priority-based dispatcher (i.e. the CAN arbitration) cannot distinguish different laxity values greater or equal $(P_{max}–P_{min})*\Delta tp$. We denote $\Delta H = (P_{max}–P_{min})*\Delta tp$ as the *time horizon* of our LLF-scheduler. A correct deadline-based scheduling of messages from $N+1$ senders requires a time horizon greater or equal to the maximum transmission time of N arbitrary messages.

Given a maximum transmission time of ΔT_{max} (in CAN: 151 bits), an inter-frame space of 3 bits, a maximum failure rate of λ_{max}, and a maximum time loss of ΔT_{fail} per failure (in CAN: 168 bits), the time horizon ΔH must satisfy the condition: $\Delta H \geq N * (\Delta T_{\max} + 3) + \lceil \Delta H * \lambda_{\max} \rceil * \Delta T_{fail}$. Thus $\Delta H \geq \frac{N * (\Delta T_{\max} + 3)}{1 - \lambda_{\max} * \Delta T_{fail}} + \Delta T_{fail}$ is a sufficient time horizon for correct deadline-based scheduling of real-time messages.

2.2 Scheduling Hard Real-time Messages by the Dynamic TDMA Scheme

Due to the high criticality of hard real-time messages, all their occurrences have to be predicted and respective transmission times have to be scheduled in advance. The scheduled transmission times must include the worst-case error handling delays and retransmission times under anticipated fault conditions. In [6] we have shown that every hard real-time message will be transmitted timely under anticipated fault conditions, if following requirements are fulfilled (see also Fig. 1):

(R1) for each hard real-time message (HRTM), an exclusive time-slot is reserved,
(R2) the length ΔT_h of the reserved time-slot of an HRTM *h* is greater or equal to the worst-case transmission time of *h* under worst-case failure occurrence,
(R3) the minimum gap ΔG_{min} between consecutive reserved time-slots is equal to the maximum time skew between any two non-faulty clocks in the system,
(R4) the latest ready time of an HRTM is at least ΔT_{fail} before its reserved time slot.
(R5) After its latest ready time, an HRTM has the highest priority in the system. This is satisfied by the condition: $P_{\min}^{SRT} > P_{\min}^{HRT} + \lfloor (\max\{\Delta T_h\} + \Delta T_{fail}) / \Delta tp \rfloor$

The requirements are fulfilled by the hybrid scheduling algorithm presented in [6].

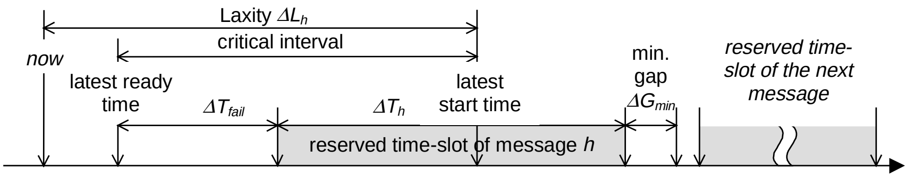

Fig. 1 . The Critical Interval of a Hard Real-Time Message *h*

2.3 Scheduling Soft and Non-Real-Time messages

Soft real time messages have transmission deadlines which are considered by the system but no guarantees are given to meet them. Thus, the transmission resources for soft real-time messages are not reserved in advance. However, the Least-Laxity-First (LLF) scheduling scheme (cf. section 2.1) is used for optimal resource utilization. For non-real-time activities, messages are sent with low (user-defined) priorities. Different priority values for non-real-time messages can be used to express the user defined importance of a message.

3 Performance Analysis of the Algorithm

3.1 Computing overhead

The main additional overhead of the hybrid bus scheduling algorithm results from the necessary periodic modification of the dynamic priority of the ready-to-send message. In our current prototype, this task takes about 13 µ-seconds on a C167 micro-controller [9] at 20MHz. Since this is a periodic task which is triggered at each priority tick, the processor overhead can be calculated as $Overhead = 13\mu s / \Delta tp$. For example, in our current prototype Δtp has the same length (500µs) for both soft and hard real-time messages, resulting in a constant processing overhead of 2.6% during the transmission of any kind of real-time message. The processing overhead of the hybrid bus scheduling approach is similar to that of TDMA [5]. Both protocols need a globally synchronized time reference, and perform the correct bus scheduling using periodic tasks. In contrast, the fixed priority approach needs no additional processing effort for message scheduling.

3.2 Evaluation of the Schedulability

Since the dynamic TDMA scheme guarantees the timely transmission of every hard real-time message by reserving a time-slot, the maximum amount of schedulable hard real-time messages depends on the length of the time-slots and the length of the gaps between time-slots, which must be twice the maximum clock inaccuracy in the system. Table 1 contains examples for the maximum schedulability at a bus speed of 1 Mbit/s and assuming a maximum clock inaccuracy of ±20μs. For a message *h* with *b* bytes of data, the maximum length of the message including header and bit-stuffing is $L_h = 75 + \lfloor b * 9.6 \rfloor$. Under the assumption of *f* transmission failures due to bus/controller errors, the required minimum time-slot length is:

$$\Delta T_h = (L_h + 18) * f + L_h + 3 \ \textit{bit-times.}$$

Table 1. Maximum Schedulability of HRTM at 1 Mbit/s under Various Fault Conditions and a Maximum Clock Inaccuracy of ±20 μsec

Data and fault model characteristics	Message length	Errors	Time loss due errors	Time-slot length	Schedulable messages
Zero bytes, single bus error	≤75 μs	1	≤93 μs	≥171 μs	≤4739
8 bytes, single bus error	≤151 μs	1	≤169 μs	≥323 μs	≤2754
Zero bytes, severe controller error	≤75 μs	16	≤1488 μs	≥1566 μs	≤622
8 bytes, severe controller error	≤151 μs	16	≤2704 μs	≥2858 μs	≤345
8 bytes, controller + single bus error	≤151 μs	17	≤2873 μs	≥3027 μs	≤326

Soft real-time messages are scheduled according to the Least-Laxity-First (LLF) scheme. Although LLF in CAN performs like EDF (which is an optimal scheduling policy), a soft real-time message will miss its deadline whenever the total waiting time (due to one non-preemptive low-priority transmission, more critical messages, and communication errors) exceeds its laxity.

The schedulability of hard real-time messages in our approach is similar to the schedulability in classic TDMA systems. In both cases, enough transmission time must be reserved to guarantee the successful transmission of the message under maximum number of tolerable consecutive faults. The advantage of our approach is that as soon as a hard real-time message is transmitted successfully, less critical messages may be transmitted in the rest of the reserved time-slot by LLF scheme. Hence, in a system with a mixture of hard and soft real-time communication, the total utilization of the bus bandwidth with the hybrid scheduling is considerably higher than with classic TDMA, especially at low error rates. However, while in classic TDMA systems a message may be transmitted in its reserved time-slot immediately, in our approach every message may be delayed after its ready time for up to ΔT_{max}, because the non-preemptive transmission of a less critical message may be already in progress.

To compare the schedulability of real-time messages with the Deadline-Monotonic scheme, we refer to the results of the analysis made by Tindell et al [11]. Although there is a feasibility test for the DM scheduling scheme, the timeliness of hard real-time communication is not monitored by the scheduling scheme. Hence, under exceptional fault and load conditions, system failures due to late hard real-time messages are possible. In contrast, both classic and dynamic TDMA approaches allow for monitoring the timeliness of the real-time messages, and detecting timing failures.

Another weak-point of the DM scheme is its inability to make difference between hard and soft real-time messages. If the DM scheme assigns a lower priority range to soft real-time messages, unnecessary soft timing failures are possible. If a soft real-time message *s* has a higher priority than a hard real-time message *h*, the higher priority of *s* may cause a late transmission of *h*, which is a fatal timing failure. This weak-point, however, can be eliminated by applying the dual priority scheduling [4].

References

[1] CiA: "CAN Application Layer (CAL) for Industrial Applications", *CiA Draft Standards 201..207, May 1993*.

[2] CiA: "CANopen Communication Profile for Industrial Systems" *CiA Draft Standard 301, Oct. 1996*.

[3] R.M. Crovella: "SDS: A CAN Protocol for Plant Floor Control", *Proc. 1st Int'l CAN Conference, 1994.*

[4] R. Davis: "Dual Priority Scheduling: A Means of Providing Flexibility in Hard Real-time Systems", *Report No. YCS230, University of York, UK, May 1994*.

[5] H. Kopetz and G. Grünsteidl, "TTP - A Time-Triggered Protocol for Fault-Tolerant Real-Time Systems", *Research Report No. 12/92, Institut für Technische Informatik, Technical University of Vienna, 1992.*

[6] M.A. Livani, J. Kaiser, W. Jia: "Scheduling Hard and Soft Real-Time Communication in Controller Area Network (CAN)", *23rd IFAC/IFIP Workshop on Real Time Programming, Shantou, China, June 1998.*

[7] D. Noonen, S. Siegel, P. Maloney: "DeviceNet Application Protocol", *Proc. 1st Int'l CAN Conference, 1994.*

[8] ROBERT BOSCH GmbH, "CAN Specification Version 2.0", *Sep. 1991*.

[9] Siemens AG: "C167, CMOS,User's Manual", *Published by Semiconductor Group, Siemens AG, 1996.*

[10] K. Tindell and A. Burns: "Guaranteeing Message Latencies on Control Area Network (CAN)", *Proceedings of the 1st International CAN Conference, 1994.*

[11] K. Tindell, A. Burns, A. Wellings: "Analysis of Hard Real-Time Communications", *J. Real-Time Systems 9(2), Sep. 1995.*

[12] K.M. Zuberi and K.G. Shin: "Non-Preemptive Scheduling of messages on Controller Area Network for Real-Time Control Applications", *Proc. Real-Time Techn. and Appl. Symposium, 1995.*