The Journal of Systems and Software 99 (2015) 66–84

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Integrating mixed transmission and practical limitations with the worst-case response-time analysis for Controller Area Network

Saad Mubeen[a,b,*], Jukka Mäki-Turja[a,b], Mikael Sjödin[a]

[a] Mälardalen Real-Time Research Centre (MRTC), Mälardalen University, Västerås, Sweden
[b] Arcticus Systems AB, Järfälla, Sweden

ARTICLE INFO

Article history:
Received 9 February 2014
Received in revised form 1 September 2014
Accepted 3 September 2014
Available online 16 September 2014

Keywords:
Controller Area Network
Real-time network
Response-time analysis

ABSTRACT

The existing worst-case response-time analysis for Controller Area Network (CAN) calculates upper bounds on the response times of messages that are queued for transmission either periodically or sporadically. However, it does not support the analysis of mixed messages. These messages do not exhibit a periodic activation pattern and can be queued for transmission both periodically and sporadically. They are implemented by several higher-level protocols based on CAN that are used in the automotive industry. We extend the existing analysis to support worst-case response-time calculations for periodic and sporadic as well as mixed messages. Moreover, we integrate the effect of hardware and software limitations in the CAN controllers and device drivers such as abortable and non-abortable transmit buffers with the extended analysis. The extended analysis is applicable to any higher-level protocol for CAN that uses periodic, sporadic and mixed transmission modes.

1. Extended version

This paper extends our previous works that are published in the conferences as a full paper in Mubeen et al. (2011) and two work-in-progress papers (discussing basic ideas and preliminary work) in Mubeen et al. (2012a,b), respectively. To be precise, the work in this paper generalizes the response-time analysis for Controller Area Network (CAN) (Robert Bosch GmbH, 1991) developed in Mubeen et al. (2011) by extending the proposed analyses in Mubeen et al. (2012a,b). In addition, we conduct a case study to show a detailed comparative evaluation of the extended analyses.

2. Introduction

CAN is a multi-master, event-triggered, serial communication bus protocol supporting speeds of up to 1 Mbit/s. It has been standardized in ISO 11898-1 (ISO, 1993). It is a widely used protocol in the automotive domain. It also finds its applications in other domains, e.g., industrial control, medical equipments and production machinery (Di Natale et al., 2012). There are several higher-level protocols for CAN that are developed for many industrial applications such as CANopen, J1939, Hägglunds Controller Area Network (HCAN) and CAN for Military Land Systems domain (MilCAN). CAN is often used in hard real-time systems that have stringent deadlines on the production of their responses. The need for safety criticality in most of these systems requires evidence that the actions by them will be provided in a timely manner, i.e., each action will be taken at a time that is appropriate to the environment of the system. For this purpose, *a priori* analysis techniques such as schedulability analysis (Audsley et al., 1993, 1995; Sha et al., 2004) have been developed. Response Time Analysis (RTA) (Joseph and Pandya, 1986) is a powerful, mature and well established schedulability analysis technique. It is a method to calculate upper bounds on the response times of tasks or messages in a real-time system or a network respectively. RTA applies to systems (or networks) where tasks (or messages) are scheduled with respect to their priorities and which is the predominant scheduling technique (Nolin et al., 2008).

2.1. Motivation and related work

Tindell et al. (1994) developed RTA for CAN which has been implemented in the industrial tools, e.g., VNA tool (Volcano Network Architect, 2014). Davis et al. (2007) refuted, revisited and revised the analysis by Tindell et al. (1994) The revised analysis is also implemented in an industrial tool suite Rubus-ICE (Rubus-ICE, 2014; Mubeen et al., 2013). The analysis in Tindell et al. (1994) and Davis et al. (2007) assumes that each node picks up the highest

* Corresponding author. Tel.: +46 704 274 019.
E-mail addresses: saad.mubeen@mdh.se, saadmubeen@hotmail.com (S. Mubeen).

http://dx.doi.org/10.1016/j.jss.2014.09.005
0164-1212/

priority message from its transmit buffers when entering into the bus arbitration. This assumption may not hold in some cases due to different types of queueing policies and hardware limitations in the CAN controllers (Di Natale et al., 2012; Khan et al., 2010; Davis et al., 2013). The different types of queueing polices in the CAN device drivers and communications stacks, internal organization, and hardware limitations in CAN controllers may have significant impact on the timing behavior of CAN messages.

Various practical issues and limitations due to deviation from the assumptions made in the seminal work (Tindell et al., 1994; Davis et al., 2007) are discussed in Di Natale and Zeng (2013) and analyzed by means of message traces in Di Natale et al. (2012). A few examples of these limitations that are considered in RTA for CAN are controllers implementing First-In, First-Out (FIFO) and work-conserving queues (Davis and Navet, 2012; Davis et al., 2013), limited number of transmit buffers (Meschi et al., 1996; Natale, 2006), copying delays in transmit buffers (Khan et al., 2011), transmit buffers supporting abort requests (Khan et al., 2010), device drivers lacking abort request mechanisms in transmit buffers (Khan et al., 2011), and protocol stack prohibiting transmission abort requests in some configurations, e.g., AUTOSAR (Transmit, 2014).

Davis et al. extended the analysis of CAN with FIFO and work-conserving queues while supporting arbitrary deadlines of messages (Davis et al., 2013; Davis and Navet, 2012). In Meschi et al. (1996), it is proved that the priority inversion due to limited buffers can be avoided if the CAN controller implements at least three transmit buffers. However, RTA in Meschi et al. (1996) does not account the timing overhead due to copying delay in abortable transmit buffers. Khan et al. (2010) integrated this extra delay with RTA for CAN (Tindell et al., 1994; Davis et al., 2007). RTA for CAN with non-abortable transmit buffers is extended in Khan et al. (2011) and Natale (2006). However, none of the above analyses support messages that are scheduled with offsets. The worst-case RTA for CAN messages with offsets is developed in several works including (Szakaly, 2003; Chen et al., 2011; Yomsi et al., 2012).

However, all these analyses assume that the messages are queued for transmission either periodically or sporadically. They do not support mixed messages which are simultaneously time (periodic) and event (sporadic) triggered. Mixed messages are implemented by several higher-level protocols for CAN that are used in the automotive industry. Mubeen et al. (2011) extended the seminal RTA (Tindell et al., 1994; Davis et al., 2007) to support mixed messages in CAN where nodes implement priority-based queues. Mubeen et al. (2012) further extended the RTA to support mixed messages in the network where some nodes implement priority queues while others implement FIFO queues. Mubeen et al. also extended the existing RTA for CAN to support periodic and mixed messages that are scheduled with offsets (Mubeen et al., 2012, 2013). In Mubeen et al. (2012a,b) we presented the basic idea for analyzing mixed messages in CAN with controllers implementing abortable and non-abortable transmit buffers respectively.

2.2. Paper contributions

We extend and generalize the RTA for periodic, sporadic and mixed messages in CAN by integrating it with the effect of buffer limitations in the CAN controllers namely abortable and non-abortable transmit buffers. The relationship between the existing and extended RTA for CAN is shown in Fig. 1. The analyses enclosed within the dashed-line box in Fig. 1 are the focus of this paper. The extended analysis is able to analyze network communications in not only homogeneous systems, but also heterogeneous systems where:

1 CAN-enabled Electronic Control Units (ECUs) are supplied by different tier-1 suppliers such that some of them implement abortable transmit buffers, some implement non-abortable transmit buffers, while others may not have buffer limitations because they implement very large but finite number of transmit buffers;
2 any higher-level protocol based on CAN is employed that uses periodic, sporadic and mixed transmission modes for messages.

It should be noted that the main contribution in this paper, compared to the contributions in Mubeen et al. (2011, 2012a,b), is that the extended analysis is also applicable to the heterogeneous systems. Moreover, we conduct a case study to show the applicability of the extended analyses. We also carry out a detailed comparative evaluation of the extended analyses.

2.3. Paper layout

The rest of the paper is organized as follows. In Section 3, we discuss the mixed messages. Section 4 describes the system model. In Section 5, we present the extended RTA for mixed messages without buffer limitations. Sections 6 and 7 discuss the extended RTA for mixed messages in the case of abortable and non-abortable transmit buffers respectively. Section 8 presents a case study and evaluation. Section 9 concludes the paper.

3. Mixed messages implemented by the higher-level protocols

Traditionally, it is assumed that the tasks queueing CAN messages are invoked either periodically or sporadically. However, there are some higher-level protocols for CAN in which the task that queues the messages can be invoked periodically as well as sporadically. If a message can be queued for transmission periodically as well as sporadically then the transmission type of the message is said to be mixed. In other words, a mixed message is simultaneously time (periodic) and event triggered (sporadic). We identified three types of implementations of mixed messages used in the industry.

Consistent terminology. We use the terms message and frame interchangeably because we only consider messages that fit into one frame (maximum 8 bytes). We term a CAN message as periodic, sporadic or mixed if it is queued by an application task that is invoked periodically, sporadically or both (periodically and sporadically) respectively. If a message is queued for transmission at periodic intervals, we use the term “Period” to refer to its periodicity. A sporadic message is queued for transmission as soon as an event occurs that changes the value of one or more signals contained in the message provided the Minimum Update Time (*MUT*) between queueing of two successive sporadic messages has elapsed. We overload the term “*MUT*” to refer to the “Inhibit Time” in the CANopen protocol (CANopen Application Layer, 2002) and the “Minimum Delay Time (MDT)” in AUTOSAR communication (AUTOSAR, 2014).

3.1. Method 1: implementation in the CANopen protocol

A mixed message in the CANopen protocol (CANopen Application Layer, 2002) can be queued for transmission at the arrival of an event provided the Inhibit Time has expired. The Inhibit Time is the minimum time that must be allowed to elapse between the queueing of two consecutive messages. The mixed message can also be queued periodically at the expiry of the Event Timer. The Event Timer is reset every time the message is queued. Once a mixed message is queued, any additional queueing of it will not take place during the Inhibit Time (CANopen Application Layer, 2002).