Computer Standards & Interfaces 45 (2016) 13–25

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

A reference model for the timing analysis of heterogeneous automotive networks

Rodrigo Lange [a,*], Romulo S. de Oliveira [b], Francisco Vasques [c]

[a] Federal Institute of Rio Grande do Sul, Porto Alegre, Brazil
[b] DAS — Federal University of Santa Catarina, Florianópolis, Brazil
[c] IDMEC — Faculty of Engineering, University of Porto, Portugal

ARTICLE INFO

Article history:
Received 18 September 2014
Received in revised form 23 October 2015
Accepted 24 October 2015
Available online 10 November 2015

Keywords:
FlexRay
Automotive
Real-time

ABSTRACT

The complexity of electronic systems embedded in modern vehicles has led to the adoption of distributed implementations where different communication protocols are used. Although literature addressing vehicular networks presents several methods for the timing analysis of automotive systems, there is not a reference model for the holistic time analysis of heterogeneous systems where FlexRay/CAN protocols are used. In this work we propose a reference model for timing and schedulability analysis of heterogeneous FlexRay/CAN networks. The proposed reference model can be used to compute end-to-end response times and to analyze local components, such as response times in a specific network segment.

1. Introduction

Typically, a modern vehicle embedded system is sub-divided into functional domains, each one having its own characteristics and constraints. Two of these domains are specifically related to the vehicle real-time control: the power train, which is related to engine and transmission control, and the chassis, which is related to the suspension control, steering and brakes. Safety issues are fundamental in these domains. In systems currently being used in vehicles, these functions are distributed across multiple electronic control units (ECUs) that are connected by a data communication network. Within this context, networks and communication protocols are of utmost importance since they support the integration of distributed functions, reducing the cost and complexity of cabling and providing means for the implementation of advanced fault tolerance techniques.

The Controller Area Network (CAN) has been a de facto standard for communication in automotive applications. More recently, new protocols for automotive applications were developed [37]. The FlexRay Communication System [10] was designed for communications in vehicular domains with high requirements for determinism, synchronization and bandwidth such as chassis and powertrain [37,7,22].

Although CAN has low data rate and frame payload when compared to FlexRay, it has low cost and meets the requirements of several vehicular applications. Even more, there is a whole class of existing CAN systems for vehicular applications with low performance requirements that can be reused even in modern high-end vehicles. Examples of such potentially reusable systems are sensors and actuators manufactured by Bosch GmbH, such as their ABS M4 Kit or their HP Injection Power Stage [5]. FlexRay, on the other hand, is designed for use in backbones and/or in applications with high requirements of bandwidth, determinism and synchronization. Therefore, the coexistence of FlexRay and CAN protocols in the same vehicle is expected, with several advantages such as costs reduction and reuse of legacy components [25].

If ECUs connected to a network segment exchange information with ECUs connected to a network segment running a different protocol, it is necessary for a network gateway to manage such exchange [16]. Literature addressing vehicular networks presents several methods for the timing analysis of ECUs (including gateways) and CAN and FlexRay networks. To the best of our knowledge, there is not a reference model for the holistic time analysis of heterogeneous FlexRay/CAN systems. Although vehicular standards such as AUTOSAR [3] define architectures for complete vehicular systems, they don't address the issues related to the timing analysis.

Therefore, the objective of this work is to propose a reference model for timing and schedulability analysis of heterogeneous FlexRay/CAN networks interconnected by gateways. The proposed reference model uses the concept of standardized components and the concept of a flow from a sender task to a receiver task used in techniques for deadline partition. It can be used to compute not only end-to-end response times, but also to analyze local components, such as response times of messages in a specific network segment. Even more, when combined with techniques such as deadline partitioning [15], the proposed reference model can be used in

* Corresponding author.
E-mail addresses: rodrigo.lange@restinga.ifrs.edu.br (R. Lange), romulo.deoliveira@ufsc.br (R.S. de Oliveira), vasques@fe.up.pt (F. Vasques).

http://dx.doi.org/10.1016/j.csi.2015.10.004

the design of an automotive system in such a way that the design of a sub-network can be done with a certain degree of independence from the overall system.

Although there are many papers describing methods for the timing analysis of the different elements of an automotive network, such as, CAN and FlexRay buses, in an automotive network all the elements must work together, and the timing requirements are end-to-end, and not associated with specific components of the embedded electronic solutions. In a real scenario, there may not be global clock synchronization among all components, and then the timing analysis must consider the interplay among the components. To the best of our knowledge, this is the first paper putting together the several aspects of a heterogeneous automotive network and providing a general framework for its timing analysis, combining previously published specific analysis.

The remaining of this document is divided as follows. Sections 2, 3 and 4 present, respectively, the basic concepts of FlexRay, CAN and automotive gateways. Section 5 presents the proposed reference model, Section 6 presents relevant techniques for time analysis, Section 7 presents an example illustrating the use of the proposed reference model and, finally, Section 8 presents the conclusions and final remarks of this work. In Appendix A we include a list of symbols, parameters and variables defined or used through this paper.

2. FlexRay Communication System

FlexRay is a communication protocol for automotive systems with high communication requirements for determinism, synchronization, bandwidth and reliability in automotive systems, such as those related to X-by-Wire. FlexRay provides static and dynamic methods for the transmission of messages, incorporating the advantages of well-known synchronous and asynchronous protocols. Among other features, it provides fault-tolerant clock synchronization, collision-free medium access, two communication channels with transmission rates up to 10 Mbit/s per channel and a data payload up to 254 bytes per frame [10].

In FlexRay, the media access control (MAC) is ruled by a communication cycle with a predetermined size FC_{bus}. A FlexRay cycle (FC) runs periodically from the beginning to the shutdown of the network. The cycle number is controlled by the variable *vCycleCounter* that is incremented by one at the beginning of each communication cycle. *vCycleCounter* ranges from 0 up to 63, and when the maximum value is reached *vCycleCounter* is reset to zero in the next communication cycle instead of being incremented. According to [22,18] the fixed size of a FlexRay cycle imposes a restriction upon the periods of message streams in a FlexRay system: FC_{bus} must be equal or less than the value of the minimum period among the streams of the system plus the size of one static slot (Eq. (1), where $T_{min_{FR}}$ is the minimum period among the periods of FlexRay message streams and *gdStaticSlot* is the size of a static slot), or a FlexRay message may miss its deadline.

$$FC_{bus} \leq P_{min_{FR}} + gdStaticSlot \tag{1}$$

Each FlexRay cycle is structured as a sequence of four segments: a *Static Segment* (ST), a *Dynamic Segment* (DN) and two control segments, the *Symbol Window* (SW) and the *Network Idle Time* (NIT) (Fig. 1).

The ST is based on the Time Division Multiple Access (TDMA) technique. It has size ST_{bus} and is composed of a number *gNumberOfStaticSlots* of static slots that are incrementally counted at each communication cycle. Each static slot is composed of the same number of macroticks whose value is set by *gdStaticSlot*. In a FlexRay cluster, *gNumberOfStaticSlots* and *gdStaticSlot* are global values that must be defined during the design phase. Arbitration in the ST is performed by assigning one or more frame identifiers (*FrameIDs*) to each network node. When the *FrameID* assigned to a specific node matches the static slot counter, that node is allowed to transmit. The ST supports a deterministic communication environment, since it is exactly defined when each frame will be transmitted in a given channel [37].

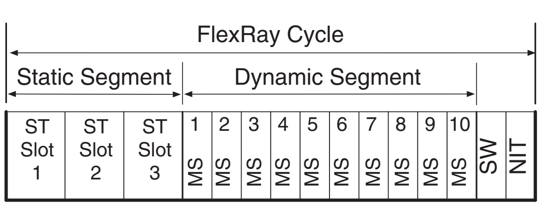

Fig. 1. FlexRay cycle [10].

The length $C^{ST}_{h,j}$ in seconds of a FlexRay static frame $m_{h,j}$ with $lm^{ST}_{h,j}$ data bytes, including frame overheads, is given by:

$$C^{ST}_{h,j} = \left(88\,\text{bit} + lm^{ST}_{h,j} \cdot 10\,\text{bit}\right)\tau^{FR}_{bit} \tag{2}$$

where τ^{FR}_{bit} is the time required to transmit one bit.

The FlexRay Dynamic Segment is also divided into dynamic slots, and the identifiers of the dynamic slots (*FrameIDs*) are allocated to the network nodes. The DN MAC is based on the *Flexible TDMA* (FTDMA) [22]: the segment is divided into *gdNumberOfMinislots* minislots (MS) that have equal size *gdMinislot*, and frames are transmitted within dynamic slots. Dynamic slots can have variable sizes and are superimposed on the minislots. We will abbreviate *gdNumberOfMinislots* as *gdNofMS*.

Each node has a parameter *pLatestTx*. According to the FlexRay specification, *pLatestTx* is defined for each node and represents the latest transmission instant at which the node can transmit in the DS. The value for *pLatestTx* is given by $pLatestTx = gdNofMS - aMinislotPerDNamicFrame + 1$, where *aMinislotPerDNamicFrame* is the number of MS needed for the longest message of the node [27].

When a node is allowed to transmit in a certain dynamic slot one of the following situations may occur: *I*) If there is no data to transmit and *zMinislot* haven't reached the end of DN, the dynamic slot has the size of only one MS; *II*) If there is a frame to be transmitted, and *zMinislot* has not reached *pLatestTx* of the node, the dynamic slot uses a number of MS large enough to accommodate the transmission; or *III*) If there is a frame to be transmitted, *zMinislot* hasn't reached *gdNofMS*, but *zMinislot* has reached *pLatestTx*, the dynamic slot uses only one MS.

The length $C^{DN}_{h,j}$ of a dynamic frame $m_{h,j}$ with $lm^{DN}_{h,j}$ data bytes, including frame overheads, is given by:

$$C^{DN}_{h,j} = \left(lm^{DN}_{h,j} \cdot 16\,\text{bit} + lm^{DN}_{h,j} \cdot 4\,\text{bit} + O_F\right)\tau^{FR}_{bit} \tag{3}$$

where $lm^{DN}_{h,j} \cdot 16$ bit is the signal data in multiples of two byte words, $lm^{DN}_{h,j} \cdot 4$ bit $+ O_F$ is the framing overhead and τ^{FR}_{bit} is the transmission time of a single bit in the FlexRay bus [10].Since a FlexRay DN frame is superimposed over the minislots of the dynamic segment, its size is usually represented as integer multiples of MS [24]. Therefore, the length $C^{DN}_{m}(MS)$ in integers of MS is given by:

$$C^{DN}_{h,j}(MS) = \frac{C^{DN}_{h,j}}{T_{MS}} \tag{4}$$

where T_{MS} is the length in seconds of a minislot.

FlexRay allows the use of slot multiplexing, which means that a *FrameID* can be associated with multiple message streams from a same node, provided that the association occurs only at specific FlexRay cycles [26,36,14]. The AUTOSAR Specification [3] restricts cycle repetition to be values that are a power of 2 with the exponent ranging from 0 to 6, with that assignments that repeat every 1, 2, 4,…, 64 FCs can be defined [26].

More information about FlexRay and its segments can be found in [10,22,24].