This work-in-progress paper was presented as part of the main technical program at IEEE ETFA'2011

Extending Response-Time Analysis of Controller Area Network (CAN) with FIFO Queues for Mixed Messages

Saad Mubeen*, Jukka Mäki-Turja*† and Mikael Sjödin*
*Mälardalen Real-Time Research Centre (MRTC), Mälardalen University, Västerås, Sweden
†Arcticus Systems, Järfälla, Sweden
{saad.mubeen, jukka.maki-turja, mikael.sjodin}@mdh.se

Abstract

Existing response-time analysis for Controller Area Network (CAN) messages in networks where some nodes implement FIFO queues while others implement priority queues, assumes that at every node, CAN messages are queued for transmission periodically or sporadically. However, there are a few high level protocols for CAN such as CANopen and Hägglunds Controller Area Network (HCAN) that support the transmission of mixed messages as well. A mixed message can be queued for transmission both periodically and sporadically. The existing analysis of CAN with FIFO queues does not support the analysis of mixed messages. We extend the existing response-time analysis of mixed-type CAN messages. The extended analysis can compute the response-times of mixed (periodic/ sporadic) messages in the CAN network where some nodes use FIFO queues while others use priority queues.

1. Introduction

Controller Area Network (CAN) [5] is a real-time, event-triggered, serial communication bus protocol. It supports bus speeds of up to 1 mega bits per second. CAN is a largely used real-time network in automotive domain. There are many high level protocols and commercial extensions of CAN developed for many industrial applications. These include CAN Application Layer (CAL), CANopen, Hägglunds Controller Area Network (HCAN), CAN for Military Land Systems domain (MilCAN), etc.

1.1. Related Work

The schedulability analysis of CAN was developed by Tindell et al. [9] by adapting the theory of fixed priority preemptive scheduling for uniprocessor systems. This analysis has been implemented in the analysis tools that are used in the automotive industry [2]. Moreover, it has served as the basis for many research projects. Later on, the analysis was refuted, revisited and revised by Davis et al. [3]. The communication model used in the analysis [9, 3] supports CAN messages that are queued for transmission periodically (with a period) or sporadically (with a minimum inter-arrival time). The analysis does not support the response-times computation of mixed-type CAN messages, i.e., the messages that are simultaneously time (periodic) and/or event triggered.

Moreover, the analysis assumes that CAN device drivers in every node in the network implement priority-based queues. This means that the highest priority message at each node enters into arbitration on the network. Davis et al. [4]pointed out that this assumption may become invalid when a node, in a CAN network, implements FIFO queues. They extended response-time analysis of CAN messages in the networks where some nodes implement priority-based queues and some implement FIFO queues. However, the extended analysis does not support mixed messages.

In [7], we extended the analysis developed by Tindell et al. [9] and revised by Davis et al. [3] for mixed-type CAN messages. The extended analysis supports response-time computation of CAN messages that are queued for transmission periodically, sporadically and both periodically/ sporadically (mixed). However, the extended analysis uses the same assumption as in [9, 3], that all nodes in the CAN network implement priority-based queues.

1.2. Paper Contribution

We extend the response-time analysis of CAN messages by integrating CAN analysis with FIFO queues [4] and CAN analysis for mixed messages [7]. The extended analysis is able to compute the response times of periodic, sporadic and mixed CAN messages in networks where some nodes implement priority-based queues while others implement FIFO queues.

2. Mixed Transmission of a CAN Message

If a message can be queued periodically as well as at the arrival of an event then the transmission type of a message is called mixed (periodic/ event) or simply mixed transmission. We identified two different implementation methods of mixed CAN messages.

2.1. Method 1: Implementation of a Mixed Message

The CANopen protocol [1] provides an example of the first implementation method of a MIXED message. A mixed message can be queued for transmission at an arrival of an event provided an Inhibit Time has expired. The Inhibit Time is the minimum time that must be allowed to elapse between the queueing of two consecutive messages. A mixed message can also be queued periodically at the expiry of an Event Timer. Hence, the expiry of an Event Timer is considered as an additional event for queueing of a mixed message. The Event Timer is reset every time the message is queued. It should be noted that once a mixed message is queued for transmission, any additional queueing of the same message will not take place during the Inhibit Time [1]. The transmission pattern of a

978-1-4577-0018-7/11/$26.00

mixed message in CANopen is illustrated in Figure 1. The down-pointing arrows (labeled with numbers) symbolize the queueing of messages while the upward lines (labeled with alphabets) represent arrival of the events.

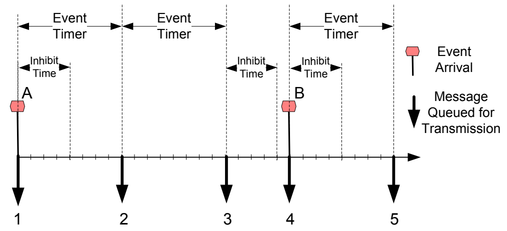

Figure 1. Transmission pattern of a Mixed Message in CANopen

In Figure 1, message 1 is queued for transmission as soon as an event A arrives (assume that the Inhibit Timer was expired). In this case, the Event Timer is reset along with the Inhibit Time. As soon as the Event Timer expires, message 2 is queued for transmission and both the Event Timer and Inhibit Time are reset. Similarly, message 3 is queued for transmission because of the expiry of the Event Timer. When an event B arrives, message 4 is immediately queued for transmission because the Inhibit Time has already expired. Note that the Event Timer is also reset at the same time when the message 4 is queued. The message 5 is transmitted because of the expiry of the Event Timer. Hence, there exist a dependency relationship between the Inhibit Time and the Event Timer.

2.2. Method 2: Implementation of a Mixed Message

The HCAN protocol [10] provides an example of the second implementation method of a MIXED message. A mixed message defined by HCAN protocol contains signals of which some are periodic and some are of event type. A mixed message is queued for transmission not only periodically but also, as soon as, an event occurs that changes the value of one or more event signals provided MUT between the queueing of two successive event messages has elapsed. Hence, the transmission of a mixed message due to arrival of events is constrained by MUT. The transmission pattern of a mixed message is illustrated in Figure 2.

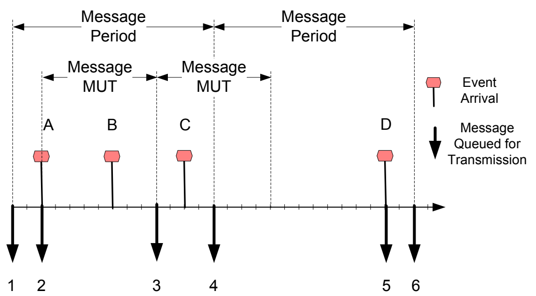

Figure 2. Transmission pattern of a Mixed Message in HCAN

In Figure 2, message 1 is queued for transmission because of the partly periodic nature of a mixed message. As soon as the event A arrives, message 2 is queued. When the event B arrives it is not queued immediately because MUT is not expired yet. As soon as MUT expires, message 3 is queued. Message 3 contains the signal changes that correspond to event B. Similarly, a message is not immediately queued when an event C arrives because the MUT is not expired. Message 4 is queued because of the periodicity. It should be noted that although, MUT was not yet expired, the event signal corresponding to event C was packed in message 4 and queued as part of the periodic message. Hence, there is no need to queue an additional event message when MUT expires. It should be noted that the periodic transmission of a mixed message cannot be blocked by the event transmission. When an event D arrives, an event message 5 is immediately queued because the MUT has already expired. Message 6 is queued due to the periodicity.

2.3. Discussion

In the first method, the Event Timer is reset every time a mixed message is queued for transmission. The most natural interpretation of a mixed message from the specification of CANopen is that there is an implicit requirement that the periodicity of transmission of a mixed message can never be higher than the Inhibit Time [1] [8]. Hence, it can be assumed that in the worst case, a mixed message is queued for transmission every time the Inhibit Timer expires. Therefore, the original CAN analysis [9, 3] can be used for mixed messages in the first method. Moreover, schedulability analysis of CAN messages with FIFO queues [4] is also applicable for mixed CAN messages that are implemented using method 1.

The second method of implementing a mixed message is more complex because the periodic transmission is independent of the event transmission. In other words, the Event Timer is not reset with every event transmission. In this case, for the purpose of analysis we treated a mixed message as two separate message streams with same IDs and priorities [7]. However, this analysis does not support the nodes with FIFO queues. This calls for the need for the extension of CAN analysis with FIFO queues to support the analysis of mixed messages.

3. System Scheduling Model

The system scheduling model is based on the communication model that was developed by Tindell et al. [9]. It combines the communication model for the response-time analysis of CAN messages with FIFO queues [4] with the communication model of CAN analysis for mixed messages [7]. The system consists of a number of nodes (processors) connected to a single CAN network. The message queue in each node may have priority-based or FIFO-based implementation. If a node implements a priority-based queue, it is designated as PQ-node. On the other hand, if a node implements a FIFO queue then it is identified as FQ-node. For a PQ-node, the highest priority message enters into the bus arbitration whereas, the oldest message enters into the bus arbitration in a FQ-node [4].

Each CAN message m has an ID_m which is a unique identifier. Associated to each message is a $FRAME\_TYPE$ that specifies whether the frame is a Standard or an Extended CAN frame. The difference between the two frame types is that a standard CAN frame uses an 11-bit identifier whereas an extended CAN frame uses a 29-bit identifier. There is a $TRANSMISSION\_TYPE$ of each message that specifies whether the message is PERIODIC or EVENT or MIXED (both PERIODIC and EVENT). Each message has a unique priority (P_m), transmission time (C_m) and queueing jitter (J_m) which is inherited from the response time of the task queueing the message.

Each message can carry a data payload (ranges from 0 to 8 bytes) denoted by s_m. In case of PERIODIC transmission, each frame has a period, denoted by T_m. In case of EVENT transmission, each frame has a MUT_m that refers to the minimum time that should elapse between the transmission of any two EVENT frames. Each message has a blocking time B_m which refers to the largest amount of time this message can be blocked by any lower priority message. Each message has a worst-case response time, denoted by R_m, and defined as the longest time between the queueing of the message (on the sending node) and the delivery of the message to the destination buffer (on the destination node).

When a message has a MIXED transmission type, we duplicate the message in the analysis model. Hence, each MIXED message has two copies which are treated as separate messages. One copy is the PERIODIC message and the other is an EVENT message. All the attributes of duplicates, including ID, priority, release jitter, transmission time and blocking time, are the same except the PERIODIC copy inherits T_m while the EVENT copy inherits MUT_m.

If an FQ-node transmits a message m then the set of all the messages transmitted by this node is defined by $M(m)$. The Lowest priority message in $M(m)$ is defined by L_m. The sum of the transmission times of all the messages in $M(m)$ is defined by C_m^{SUM}. Moreover, the transmission time of the shortest and longest messages in $M(m)$ is designated by C_m^{MIN} and C_m^{MAX} respectively. f_m denotes the maximum buffering time between the instant a message m enters the priority-based or FIFO queue and the instant it takes part in priority-based arbitration or it becomes the oldest message in the queue respectively [4].

4. Extended Analysis with FIFO Queues for Mixed Messages

We extend the existing analysis of CAN with FIFO queues [4] by adapting the schedulability analysis of CAN for mixed messages [7]. Let the message under analysis be denoted by m. We treat a message differently based on its transmission type. In order to keep the notations simple and consistent, we define a function $\xi(m)$ that represents the transmission type of a message m. It can be either periodic or event or mixed. Formally, the domain of this function can be defined as:

$$\xi(m) \in [\text{PERIODIC}, \quad \text{EVENT}, \quad \text{MIXED}]$$

4.1. Case: When m is a Periodic or an Event Message

According to [4], the worst-case response time of a message under analysis is computed differently for PQ-and FQ-nodes.

4.1.1 Priority-Queued Messages

The worst-case response time of each instance q of a periodic or event-type message that is queued at a PQ-node is computed by the following equation.

$$R_m(q) = \begin{cases} J_m + \omega_m(q) - qT_m + C_m, \\ \qquad \text{if } \xi(\text{k}) = \text{PERIODIC} \\ J_m + \omega_m(q) - q(MUT_m) + C_m, \\ \qquad \text{if } \xi(\text{k}) = \text{EVENT} \end{cases} \quad (1)$$

where, q is computed according to [7]. ω_m represents the worst-case queueing delay and is equal to the longest time that elapses between the instant a message m is queued by the sending task in the send queue and the instant when the message starts its transmission. For a message queued at a PQ-node, ω_m is computed by the following fixed-point iteration.

$$\omega_m^{n+1}(q) = max(B_m, C_m) + qC_m + \sum_{\forall k \in hp(m)} I_k C_k \quad (2)$$

4.1.2 FIFO-Queued Messages

The worst-case response time of each instance q of a periodic or event-type message that is queued at an FQ-node is computed by the following equation.

$$R_m(q) = \begin{cases} J_m + \omega_m(q) - qT_m + C_m^{MIN}, \\ \qquad \text{if } \xi(\text{k}) = \text{PERIODIC} \\ J_m + \omega_m(q) - q(MUT_m) + C_m^{MIN}, \\ \qquad \text{if } \xi(\text{k}) = \text{EVENT} \end{cases} \quad (3)$$

ω_m is computed by the following fixed-point iteration.

$$\omega_m^{n+1} = max(B_{L_m}, C_m^{MAX}) + (C_m^{SUM} - C_m^{MIN}) + \sum_{\forall k \in hp(L_m) \wedge k \notin M(m)} I_k C_k \quad (4)$$

In (2) and (4), I_k is computed differently for different values of $\xi(k)$ (k is the index of any higher priority message) as shown below. Note that the interference by a higher priority MIXED message contains the contribution from both the duplicates.

$$I_k = \begin{cases} \left\lceil \frac{\omega_m^n(q) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil, & \text{if } \xi(\text{k}) = \text{PERIODIC} \\ \left\lceil \frac{\omega_m^n(q) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(\text{k}) = \text{EVENT} \\ \left\lceil \frac{\omega_m^n(q) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil + \\ \quad \left\lceil \frac{\omega_m^n(q) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(\text{k}) = \text{MIXED} \end{cases} \quad (5)$$

4.2. Case: When m is a Mixed Message

When a message under analysis is mixed, we treat the message as two separate message streams i.e., periodic and event. The response time of each is computed separately and maximum between the two is selected as the worst-case response time of the mixed message as shown below.

$$R_m = max(R_{m_P}, \quad R_{m_E}) \quad (6)$$

$$R_{m_P} = max(R_{m_P}(q_{m_P})) \quad (7)$$

$$R_{m_E} = max(R_{m_E}(q_{m_E})) \quad (8)$$

where, R_{m_P} and R_{m_E} are computed separately for the messages queued at PQ-nodes and FQ-nodes.

4.2.1 Priority-Queued Messages

The worst-case response time of each instance q of the periodic and event copies of a mixed message under analysis that is queued at a PQ-node is computed by the following equations:

$$R_{m_P}(q_{m_P}) = J_m + \omega_{m_P}(q_{m_P}) - q_{m_P} T_m + C_m \quad (9)$$

$$R_{m_E}(q_{m_E}) = J_m + \omega_{m_E}(q_{m_E}) - q_{m_E} MUT_m + C_m \quad (10)$$

where ω_m for periodic and event copies are computed by the following fixed-point iterations.

$$\omega_{m_P}^{n+1}(q_{m_P}) = max(B_m, C_m) + q_{m_P} C_m + \sum_{\forall k \in hp(m)} I_{k_P} C_k + \left\lceil \frac{q_{m_P} T_m + J_m}{MUT_m} \right\rceil C_m \quad (11)$$

$$\omega_{m_E}^{n+1}(q_{m_E}) = max(B_m, C_m) + q_{m_E} C_m + \sum_{\forall k \in hp(m)} I_{k_E} C_k + \left\lceil \frac{q_{m_E} MUT_m + J_m}{T_m} \right\rceil C_m \quad (12)$$

4.2.2 FIFO-Queued Messages

The worst-case response time of each instance q of a periodic and event copy of a mixed message under analysis that is queued at an FQ-node is computed by the following equations:

$$R_{m_P}(q_{m_P}) = J_m + \omega_{m_P}(q_{m_P}) - q_{m_P} T_m + C_m^{MIN} \quad (13)$$

$$R_{m_E}(q_{m_E}) = J_m + \omega_{m_E}(q_{m_E}) - q_{m_E} MUT_m + C_m^{MIN} \quad (14)$$

where ω_m for periodic and event copies are computed by the following fixed-point iterations.

$$\omega_{m_P}^{n+1}(q_{m_P}) = max(B_m, C_m) + q_{m_P} C_m + \sum_{\forall k \in hp(L_m) \wedge k \notin M(m)} I_{k_P} C_k + \left\lceil \frac{q_{m_P} T_m + J_m}{MUT_m} \right\rceil C_m \quad (15)$$

$$\omega_{m_E}^{n+1}(q_{m_E}) = max(B_m, C_m) + q_{m_E} C_m + \sum_{\forall k \in hp(L_m) \wedge k \notin M(m)} I_{k_E} C_k + \left\lceil \frac{q_{m_E} MUT_m + J_m}{T_m} \right\rceil C_m \quad (16)$$

where, I_{k_P} and I_{k_E} are computed according to the following fixed-point iterations.

$$I_{k_P} = \begin{cases} \left\lceil \frac{\omega_{m_P}^n(q_{m_P}) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil, & \text{if } \xi(k) = \text{PERIODIC} \\ \left\lceil \frac{\omega_{m_P}^n(q_{m_P}) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(k) = \text{EVENT} \\ \left\lceil \frac{\omega_{m_P}^n(q_{m_P}) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil + \\ \left\lceil \frac{\omega_{m_P}^n(q_{m_P}) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(k) = \text{MIXED} \end{cases} \quad (17)$$

$$I_{k_E} = \begin{cases} \left\lceil \frac{\omega_{m_E}^n(q_{m_E}) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil, & \text{if } \xi(k) = \text{PERIODIC} \\ \left\lceil \frac{\omega_{m_E}^n(q_{m_E}) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(k) = \text{EVENT} \\ \left\lceil \frac{\omega_{m_E}^n(q_{m_E}) + J_k + f_k + \tau_{bit}}{T_k} \right\rceil + \\ \left\lceil \frac{\omega_{m_E}^n(q_{m_E}) + J_k + f_k + \tau_{bit}}{MUT_k} \right\rceil, & \text{if } \xi(k) = \text{MIXED} \end{cases} \quad (18)$$

5. Conclusion

Existing response-time analysis of CAN with FIFO queues does not support the analysis of mixed messages. A mixed message is simultaneously time (periodic) and/or event triggered. Mixed messages are supported by some high-level protocols for CAN, e.g., CANopen and HCAN. We identified two methods of implementation of mixed messages in the high-level protocols for CAN. We extended the existing response-time analysis of CAN messages, in the networks that comprise of PQ- and FQ-nodes, to support the response-time computation of mixed-type CAN messages. A PQ-node implements a priority-based queue while an FQ-node implements a FIFO queue. While extending the response-time analysis, we considered the implementation of mixed message in HCAN protocol.

In future, we plan to implement the extended analysis in an existing industrial tool suite, the Rubus-ICE [6], that provides a component-based development environment for resource-constrained distributed real-time systems.

Acknowledgement

This work is supported by Swedish Knowledge Foundation (KKS) within the project EEMDEF, the Swedish Research Council (VR) within project TiPCES, and the Strategic Research Foundation (SSF) with the centre PROGRESS. The authors would like to thank the industrial partners Arcticus Systems and BAE Systems Hägglunds for the cooperation.

References

[1] CANopen Application Layer and Communication Profile. CiA Draft Standard 301. Version 4.02. February 13, 2002. www.ece.unh.edu/biolab/hof/public/CiA.

[2] L. Casparsson, A. Rajnak, K. Tindell, and P. Malmberg. Volcano - a revolution in on-board communications. In *Volvo Technology Report*, 1998.

[3] R. Davis, A. Burns, R. Bril, and J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35:239–272, 2007.

[4] R. I. 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 (ECRTS11)*, July 2011.

[5] R. B. GmbH. CAN Specification Version 2.0. Postfach 30 02 40, D-70442 Stuttgart, 1991.

[6] K. Hänninen, J. Mäki-Turja, S. Sandberg, J. Lundbäck, M. Lindberg, M. Nolin, and K.-L. Lundbäck. Framework for real-time analysis in Rubus-ICE. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 782 –788, 2008.

[7] S. Mubeen, J. Mäki-Turja, and M. Sjödin. Response-Time Analysis of Mixed-Type Controller Area Network (CAN) Messages. Technical Report, Mälardalen University (to appear in ETFA-2011), May 2011. http://www.mrtc.mdh.se/index.php?choice=publications&id=2533.

[8] O. Pfeiffer, A. Ayre, and C. Keydel. *Embedded Networking with CAN and CANopen*. Annabooks, 2003.

[9] K. Tindell, H. Hansson, and A. Wellings. Analysing real-time communications: controller area network (CAN). In *Real-Time Systems Symposium (RTSS) 1994*, pages 259 – 263.

[10] J. Westerlund. Hägglunds Controller Area Network (HCAN), Network Implementation Specification. *BAE Systems Hägglunds, Sweden (internal document)*, April 2009.