Performance Evaluation of FlexRay/CAN Networks Interconnected by a Gateway

E. G. Schmidt and M. Alkan
Department of Electrical
and Electronics Engineering
Middle East Technical University, Ankara, Turkey
Email: {eguran,e124374}@metu.edu.tr

K. Schmidt
Department of Electronic
and Communication Engineering
Cankaya University, Ankara, Turkey
Email: schmidt@cankaya.edu.tr

E. Yürüklü and U. Karakaya
TOFAŞ Türk Otomobil Fabrikası
A.Ş., BURSA
Email: {emrah.yuruklu,
utku.karakaya}@tofas.com.tr

***Abstract*—The coexistence of both CAN and FlexRay networks in contemporary and future automobiles necessitates the use of FlexRay/CAN *gateways* that support the *timely* data exchange among the different networks. In this paper, we report on the implementation of such FlexRay/CAN gateway. Moreover, for the first time, we investigate appropriate network and gateway configurations that are beneficial for the overall network performance in the sense of small *delays* of gateway messages.**

I. Introduction

The currently most widely used communication network for in-vehicle communication is the Controller Area Network (CAN) [1]. CAN's data rates between 50 Kbit/s and 1 Mbit/s and its event-triggered arbitration mechanism is not well-suited for modern applications such as x-by-wire. Accordingly, the FlexRay protocol [2] with a high bandwidth of 10 Mbit/s and support for both time-triggered and event-triggered message traffic is expected to replace CAN as the de-facto standard in in-vehicle communication [3].

The anticipated technology transition from CAN to FlexRay is such that the communication for low-speed applications will still be carried out on CAN while new high-speed functionality will be implemented on FlexRay. Consequently, a *gateway* has to enable the data exchange between the new components on FlexRay and the existing components on CAN. Such gateway has to perform fast protocol conversion between both networks in the sense that the *signals* packed in incoming messages on one network have to be mapped to outgoing messages on the other network with a bounded *processing delay*.

Previous work on gateways for in-vehicle networks includes implementations on FPGA [4], [5] which focus on the hardware performance and timing properties of pure message conversions. Only the work in [5] briefly studies the realization of a signal mapping. In addition, micro-controller-based implementations such as [6], [7], [8], [9] demonstrate that their gateway correctly converts messages. Among these studies, [9] implements a gateway in a Hybrid Electrical Vehicle test bench and reports experimental results for the achieved data rate of 0.9285 Mbit/s on CAN and 4.3478 Mbit/s on FlexRay.

The first contribution of this work-in-progress is the implementation of a FlexRay/CAN gateway that allows arbitrary mapping of signals to messages. We illustrate the gateway functionality by experimental measurements in a test environment including components from an existing vehicle. Our second contribution is the description of the scheduling requirements of a FlexRay/CAN network connected by a gateway node in conjunction with an end-to-end message delay analysis for a realistic message set. To the best of our knowledge, there is no such study on vehicular networks that are composed of more than one network.

The paper is organized as follows. Section II briefly describes the operation of CAN and FlexRay, and Section III discusses requirements and basic findings related to the FlexRay/CAN gateway realizaton. Our experimental results are reported in Section IV.

II. FlexRay and CAN Protocols: Overview

CAN is an asynchronous multi-master serial data bus that uses Carrier Sense Multiple Access/Collision Resolution (CSMA/CR). It was designed as a simple and robust broadcast bus by Robert Bosch GmbH beginning in 1983 and was standardized in 1993 [1]. CAN is capable of operating at speeds of up to 1 Mbit/s and carries messages with a payload of at most 8 B. The signals to be transmitted on CAN are packed into message frames with unique CAN identifiers (CAN IDs). The bus access conflicts are resolved with a non-destructive bit-wise arbitration of the CAN IDs by a wired-and-mechanism which gives the CAN ID with the lowest binary value the highest priority. The arbitration mechanism relies on the fact that all messages have distinct CAN IDs and all nodes on a CAN network use the same type of ID (11 bit for the base frame format and 29 bit for the extended format). Medium access delay bounds (so-called *worst-case response times*) can be computed if signal periods or minimum inter-arrival times of sporadic signals are known [10], [11].

The FlexRay protocol [3], [2] is a time-triggered protocol. Its operation is based on a repeatedly executed FlexRay cycle (FC) with a fixed duration. Messages are transmitted in FlexRay frames that consist of a payload in multiples of two-byte words and a framing overhead. The maximum specified payload is 127 two-byte words, i.e., 254 B. The FC comprises a *static segment* (SS), a *dynamic segment* (DS), a *symbol window* (SW), and the *network idle time* (NIT). The organization of the SS is based on a time-division multiple access (TDMA) scheme. It consists of a fixed number of equal size *static slots* (STSs) which are exclusively assigned to the nodes through unique frame identifiers (FIDs) of the messages that are sent. Since the STSs in the SS recur periodically, the SS is suitable for periodic messages. The DS employs the flexible time-division multiple access (FTDMA) approach with *minislots* (MS) as the smallest time unit. The operation of the DS comprises consecutive *dynamic slots* (DYS) that are superimposed on MSs. If a message is transmitted in a DYS, then the length of the DYS is equal to the number of MSs needed for message transmission. Otherwise, the length of the DYS is one MS. Since the bus arbitration in the DS dynamically adapts to the transmission requirements, the DS is suitable for sporadic messages. Finally, the SW and the NIT provide time for internal control information and protocol-related computations.

III. FLEXRAY/CAN GATEWAY NETWORK

In this paper, we focus on the case where a CAN network and a FlexRay network exchange signals via a gateway. Such gateway has to operate as both a CAN node and a FlexRay node, while performing the *protocol conversion* between both networks. In this respect, it extracts signals from the payload of each incoming message and assembles new frames for the respective other bus. That is, information about the *signal mapping* from CAN data to FlexRay messages and vice versa has to be available. We now describe the principal operation of the Gateway developed in our work.

A. Gateway Functionality

The transmission path from FlexRay to CAN is performed by the gateway task `FR2CAN`. Since it is usually the case that the payload of a FlexRay message is larger than the maximum payload of CAN messages, we assume that the data from one FlexRay message is assigned to potentially more than one CAN message. In addition, we require that the assignment is unique in the sense that a CAN message that is used to carry payload data of one FlexRay message cannot be used for another Flexray message. Regarding the fact that the payload of each message is normally divided into separate signal data, our approach amounts to uniquely mapping the respective signal bits in the FlexRay message to signal bits in the assigned CAN messages. Figure 1 part (a) shows a mapping example where FlexRay message D_3 with FID 68 is received by the gateway. Signals S_3, S_4 and S_5 in D_3 are mapped to 3 separate CAN messages C_{17}, C_{19} and C_{22} with CAN IDs 19, 13 and 8 respectively.

The transmission path from CAN to FlexRay is performed by the gateway task `CAN2FR`. We suggest an analogous mapping from CAN signal data to the payload of FlexRay messages. Considering periodic messages, we additionally allow that the signal data of multiple periodic CAN messages (with the same period) can be mapped to the same periodic FlexRay message. This reflects the fact that FlexRay messages can have much larger payloads. Respecting the nondeterministic arrival of sporadic messages, we suggest a one-to-one mapping from CAN to FlexRay for such messages. Figure 1 part (b) shows a mapping example with 3 CAN messages C_{23} (ID 6), C_{21} (ID 9) and C_{20} (ID 11) carrying signals S_{12}, S_{11} and S_{10} respectively. Signal S_{12} is mapped to FlexRay message P_{16} with FID 34. Signals S_{11} and S_{10} are mapped together to a single FlexRay message P_{10} with FID 33.

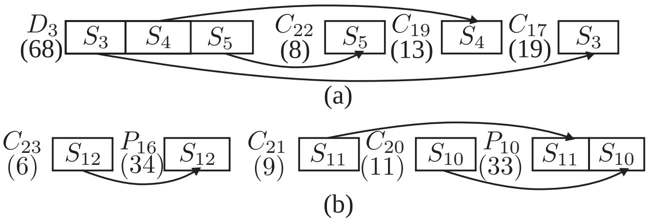

Fig. 1. Signal mapping: (a) `FR2CAN`; (b) `CAN2FR`.

B. Performance Metrics

We investigate two types of performance metrics.

First, the basic *operation of the gateway* requires the correct *protocol conversion* between FlexRay and CAN including the capability of *mapping signals to messages*. This prerequisite enables the efficient data exchange among both networks. In addition, the *gateway processing delay* t_{GW} that comprises the execution time of the tasks `CAN2FR` and `FR2CAN` must be bounded to support the timely data exchange.

Second, noting that the real-time operation of modern vehicles requires that the worst-case response time for each signal S should be less than its deadline, we quantify the performance of the *overall network* by each signal's *worst-case response time*. In particular, we focus on signals that pass the gateway and hence encounter the gateway processing delay as well as delays on both CAN and FlexRay that depend on the delivery timing of the respective network.

C. Message Scheduling

The values for the worst-case response time of the signals are directly related to the schedules of the messages that carry them on CAN and FlexRay networks. If signal S with deadline d_S is transmitted through the gateway, the correctness of the following relation involving the worst-case response time of S on CAN and FlexRay networks (wc_S^C and wc_S^{FR}, respectively) must be ensured by the appropriate scheduling decisions for both networks as well as bounded gateway processing delay.

$$d_S \geq wc_S^C + wc_S^{FR} + t_{\mathrm{GW}}. \tag{1}$$

Separate scheduling algorithms that allow bounded worst-case response times on CAN and FlexRay exist in the literature [11], [12], [13]. However, the combined scheduling problem on a FlexRay/CAN gateway network has not been addressed, yet. In particular, (1) suggests that there is a cyclic dependency between the worst-case response times on CAN and FlexRay for each signal S since only their sum $wc_S^C + wc_S^{FR}$ has to be adjusted in order to fulfill (1). We resolve this cyclic dependency as follows. We propose to first apply an appropriate CAN scheduling strategy that causes as little delay on the CAN network as possible. If a feasible schedule is found, the computed wc_S^C resulting from this schedule is used to compute wc_S^{FR} which then serves as the deadline of the message that carries S on the FlexRay network.

In this work-in-progress, we focus on a specific version of this combined message scheduling problem that is relevant especially in an industrial context. The new functionalities such as x-by-wire applications are implemented over FlexRay, whereby the information exchange with the existing components on the CAN network is conducted through the gateway which requires new CAN messages. Noting that CAN networks have been used in series-production vehicles for many years, it is not feasible to completely change the existing reliable CAN schedules. It is rather desired to add the new messages to the CAN schedule while preserving the priorities (IDs) of the existing CAN messages. To this end, we first identify the unused CAN IDs which constitute the gaps between the priorities of the existing CAN messages. We place the new CAN messages in these priority gaps with as high priorities as possible. Then, we test if the *entire* CAN message set is schedulable following the approach in [11], i.e., comparing the resulting worst-case response times with the respective signal deadlines. If all deadlines are met, we accept the computed CAN schedule. Otherwise, we decrease the priorities of the new messages until the message set is schedulable or it is determined that there is no schedulable priority assignment which can be decided in polynomial time.

Next, we employ existing methods such as [12], [13] to construct the message schedule for the FlexRay network, whereby the deadline for each FlexRay message passing the gateway is determined based on (1).

IV. Gateway Implementation and Experiments

A. Gateway Implementation Environment

In our implementation, the network nodes are realized by the evaluation boards SK-91465X-100MPC [14] which comprise a 32-bit Flash microcontroller unit (MCU) MB91F465XA that supports both the FlexRay and the CAN protocol operations. The FlexRay communication controller for two independent FlexRay channels A and B is implemented by two Bosch ERay type IP-modules [15] and the physical layer of the FlexRay bus is realized by AMS8221B transceivers. Likewise, there are two independent high-speed CAN channels. On the one hand, the evaluation boards can be used both as FlexRay nodes and CAN nodes. Then, the task of the MCU is the generation of messages to be sent on the respective automotive network. On the other hand, the evaluation boards are suitable for the implementation of the FlexRay/CAN gateway functionality. In that case, the MCU has to realize the tasks `FR2CAN` and `CAN2FR` as described in Section III. As the bus analysis tool, the Flexcard Cyclone II SE [16] that can observe the traffic on two independent FlexRay channels and 2 high-speed CAN channels is used. In our setup, it records payloads and accurate timestamps of all messages sent on CAN and FlexRay.

B. Protocol Conversion Experiment

We first test the correctness of our gateway implementation in terms of protocol conversion. The tasks `FR2CAN` and `CAN2FR` are demonstrated on a test bed with components from an existing vehicle. The CAN nodes comprise an Instrument Panel Cluster (IPC) node on B-CAN (50 Kbit/s-29bit CAN IDs), Steering Angle Sensor (SAS) node on C-CAN (50 Kbit/s-29bit CAN IDs) and 2 nodes on FlexRay (see Fig. 2). The IPC receives and displays vehicle speed in km/h and engine speed in $\mathrm{rpm} \times 1000$. It also sends status messages to the FlexRay nodes. SAS sends the steering angle to the FlexRay nodes that are implemented on SK-91465X-100MPC evaluation boards. They generate and send the values for engine and vehicle speed to the IPC on B-CAN network. In addition, one of the FlexRay nodes sends a status message to C-CAN network. The gateway interconnects these three networks and sends messages between FlexRay and two CAN networks.

We observed that the displays in the IPC correctly show the engine speed and vehicle speed values as sent by the FlexRay nodes. In addition the status message sent to C-CAN network was verified using a CAN Analyzer tool which is not shown in the figure. The messages received from B-CAN and C-CAN networks were monitored and verified by the FlexAlyzer tool. A video of the running experiment can be seen at [17].

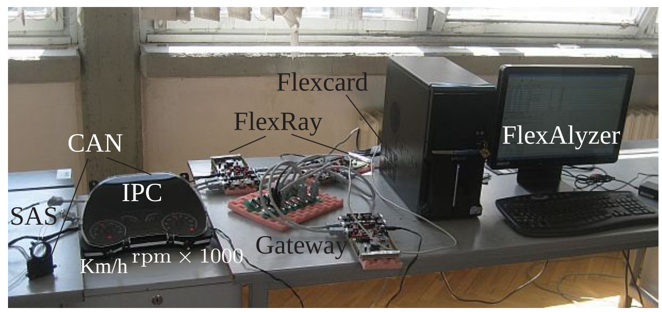

Fig. 2. Experimental setup.

C. Worst-case Response Time Experiments

This experiment set-up contains a FlexRay network with 3 nodes and a 500 Kbit/s C-CAN network with 3 nodes and a gateway node, each implemented on a different SK-91465X-100MPC evaluation board. There are 15 signals (S_1 to S_{15}) which are mapped to CAN and FlexRay messages to be transferred between two networks. The message set used in the experiment is derived from the message set of a real vehicle. It consists of 26 CAN messages (C_1 to C_{26}) and 46 FlexRay messages where P_1 to P_{41} are sent in the SS and D_1 to D_5 are sent in the DS. The message set results in 624 805 bit/s and 117 500 bit/s of traffic on FlexRay SS and DS, respectively and 322 500 bit/s traffic on CAN. The amount of traffic that is to be transmitted through the gateway by `CAN2FR` task is 75 Kbit/s and by `FR2CAN` task is 77 Kbit/s.

The mapping for the gateway signals is indicated in Table I in columns S_M and R_M for sending and receiving messages, respectively. The remaining messages constitute background traffic on each individual network. The scheduling of CAN messages is carried out according to the approach outlined in Section III-C. The FlexRay schedule is chosen such that the messages are delivered before their deadline on FlexRay. The signal-message assignment ensures that all signal deadlines are met based on the worst-case response times computed for the messages on the different networks and the condition in (1).

S	S_M	R_M	w_S/ms CAN	w_S/ms Flexray	w_S/ms total	d_S/ms
S_1	P_2	C_9	0.96 (0.96)	0.92 (5)	1.82 (6.46)	7.5
S_2	P_2	C_{10}	0.64 (0.64)	0.91 (5)	1.52 (6.14)	7.5
S_3	D_3	C_{17}	2.29 (5.04)	10 (10)	12.38 (14.04)	20
S_4	D_3	C_{19}	1.13 (4.16)	10 (10)	11.22 (13.52)	20
S_5	D_3	C_{22}	0.64 (2.68)	10 (10)	10.78 (12.68)	20
S_6	C_{14}	P_5	7.8 (7.82)	8.23 (10)	15.73 (17.22)	20
S_7	C_{15}	P_5	2.4 (7.24)	8.23 (10)	10.73 (17.24)	20
S_8	C_{16}	P_7	4.22 (6.28)	7.68 (10)	10.76 (16.28)	20
S_9	C_{18}	P_7	4.78 (5.12)	7.65 (10)	10.76 (15.12)	20
S_{10}	C_{20}	P_{10}	2.86 (3.52)	9.33 (10)	10.73 (13.52)	15
S_{11}	C_{21}	P_{10}	2.34 (3.0)	9.33 (10)	10.73 (13.0)	15
S_{12}	C_{23}	P_{10}	1.02 (2.04)	10 (10)	10.76 (12.04)	15
S_{13}	C_{24}	P_{16}	0.88 (1.84)	5 (10)	5.76 (11.84)	15
S_{14}	C_{25}	D_4	0.7 (1.68)	6.64 (10)	7.04 (11.68)	10
S_{15}	C_{26}	D_4	0.7 (1.16)	6.64 (10)	7.04 (11.16)	10

TABLE I
SIMULATION RESULTS FOR THE SIGNALS PASSING THE GATEWAY.

The values in parentheses for the worst-case response times in Table I represent the theoretical maximum for the respective performance metric that is computed analytically.

The results in Table I are obtained from timestamps that are recorded in the payloads of the messages as well as the measurements of the FlexAlyzer which is connected to both networks. For each signal, a timestamp is taken for each message transmission on the path to the destination. The gateway processing delay t_{GW} is measured to be between 44μs and 50μs. The results show that the worst-case response times of the signals that pass the gateway satisfy the analytically computed worst-case response times on the individual networks ("w_S/ms on CAN" and "w_S/ms") as well as their end-to-end deadline ("w_S/ms total").

We finally demonstrate the capability of the gateway to map signals using the subset of signals S_3-S_5 and $S_{10}-S_{12}$ in Table I that is also depicted in Fig. 1. Fig. 3 shows screenshots of FlexAlyzer that demonstrate the correct signal mapping. Note that the FlexAlyzer displays a different byte order for the payloads transmitted on FlexRay and CAN.

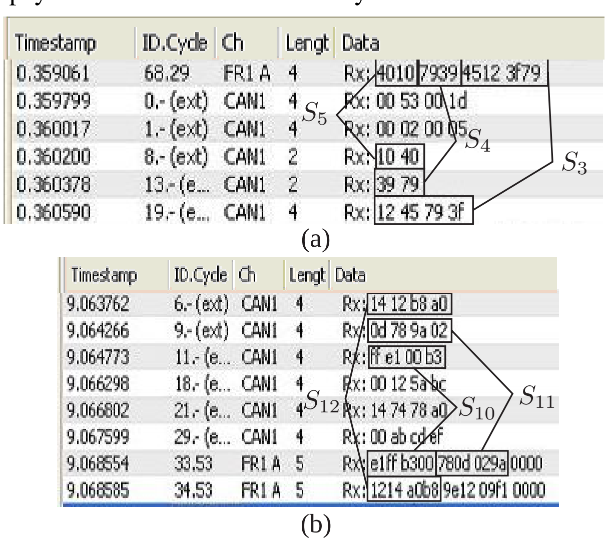

Fig. 3. Gateway operation: (a) FR2CAN; (b) CAN2FR.

V. CONCLUSION

This work-in-progress studies the performance of FlexRay/CAN networks interconnected by a gateway unit for in-vehicle communication. In particular, we focus on the flexibility of the gateway implementation regarding the mapping of signals to messages and the worst-case response times encountered by signals that pass the gateway including a gateway processing delay. Furthermore, we outline an integrated scheduling problem for priority assignment on both networks such that the end-to-end worst-case signal response times do not exceed their deadlines. We then evaluate these metrics by an experimental study. We first show that the gateway correctly performs the protocol conversion in a FlexRay/CAN network test bed with CAN nodes that are components of an existing vehicle. Second, we demonstrate the signal mapping capability of the gateway and the end-to-end worst-case response time of the signals in the overall network with a large message set that is derived from the signals in a real vehicle. The maximum processing delay of our gateway implementation is measured to be 50 μs.

ACKNOWLEDGMENT

The authors would like to thank TOFAŞ Türk Otomobil Fabrikaları A.Ş. for the support of the project.

REFERENCES

[1] I. Standard-11898, "Road vehicles-interchange of digital information – Controller Area Network (CAN) for high-speed communication," *International Standards Organisation (ISO)*, 1993.

[2] (2004, June) FlexRay communication system, protocol specification, version 2.0. [Online]. Available: http://www.flexray.com

[3] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, June 2005.

[4] S. Shaheen, D. Heffernan, and G. Leen, "A gateway for time-triggered control networks," *Microprocessors and Microsystems*, vol. 31, no. 1, pp. 38–50, 2007.

[5] T. Lorenz, *Advanced Gateways in Automotive Applications*. Ph.D. Thesis, Elektrotechnik und Informatik der TU Berlin, 2008.

[6] H. Li, H. Zhang, D. Peng, and W. Huang, "Design and application of communication gateway based on FlexRay and CAN," in *International Conference on Electronic Computer Technology*, 2009, pp. 664–668.

[7] T. Y. Moon, S. H. Seo, J. H. Kim, S. H. Hwang, and J. W. Jeon, "Gateway system with diagnostic function for LIN, CAN and FlexRay," in *Conference on Control, Automation and Systems*, 2007, pp. 2844–2849.

[8] S. H. Seo, T. Y. Moon, J. H. Kim, K. H. Kwon, J. W. Jeon, and S. H. Hwang, "A fault-tolerant gateway for in-vehicle networks," in *IEEE Conference on Industrial Informatics*, 2008, pp. 1144–1148.

[9] S.-H. Seo, J.-H. Kim, S. Hwang, K. H. Kwon, and J. W. Jeon, "An evaluation of the FlexRay-CAN gateway-embedded system in the HEV test bench," in *Industrial Electronics, IEEE International Symposium on*, 2009, pp. 664–669.

[10] K. Tindell, A. Burns, and A. Wellings, "Calculating controller area network (CAN) message response times," *Control Engineering Practice*, vol. 3, pp. 1163–1169, 1995.

[11] R. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, no. 3, pp. 239–272, April.

[12] E. G. Schmidt and K. Schmidt, "Message scheduling for the FlexRay protocol: The dynamic segment," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 5, pp. 2170–2179, 2009.

[13] K. Schmidt and E. G. Schmidt, "Message scheduling for the FlexRay protocol: The static segment," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 5, pp. 2160–2169, 2009.

[14] (2009) Fujitsu FlexRay Evaluation Board: SK-91465X-100PMC. [Online]. Available: http://mcu.emea.fujitsu.com/mcu_tool/detail/SK-91465X-100PMC.htm

[15] (2009) Bosch E-Ray FlexRay IP-Module user's manual. [Online]. Available: http://www.semiconductors.bosch.de/pdf/ERay_Users_Manual_1_2_6.pdf

[16] (2009) Flexcard Cyclone II SE. [Online]. Available: http://www.eberspaecher.com/servlet/PB/menu/1053178_l2/index.html

[17] (2010) FlexRay/CAN demo. [Online]. Available: http://www.eee.metu.edu.tr/~eguran/FlexRayCANGateway.AVI