Modelling and Performance Analysis of a New Time-Triggered CAN FD Protocol for Real-time Distributed Control Systems

Mahmut Tenruh
Mugla S. K. University
Department of Electrical and Electronics Engineering
48000 Mugla, Turkey

Periklis Charchalakis
Vetronics Research Centre
University of Brighton, Lewes Road
Brighton, BN2 4GJ, UK

Elias Stipidis
Vetronics Research Centre
University of Brighton, Lewes Road
Brighton, BN2 4GJ, UK

ABSTRACT

Controller Area Network (CAN) is a widely accepted and implemented real-time communication bus in automotive and industrial applications. The CAN protocol applies a priority based medium access method which provides fast access to the bus for the highest priority messages. However, lower priority messages may face extensive access delays especially under heavy bus load and low transmission bit-rate conditions. As a solution, the CAN with flexible data-rate (CAN FD) protocol provides higher transmission speeds. However, it still applies the same medium access method as the traditional CAN protocol. The application of time-triggered access feature of TTCAN with the fast transmission feature of CAN FD results in the Time-Triggered CAN FD (TTCAN FD) protocol. As the main contribution of the study, this paper introduces a new method by modifying the periodic CAN FD frames to achieve higher transmission bit-rates with the arbitration-free feature of the time-triggered access. The simulation results show that the new method with time-triggered access provides considerable performance improvements compared to existing approaches.

General Terms

Industrial communication networks, Fieldbus, Network protocols.

Keywords

Controller Area Network, Autobus, CAN FD, Real-time.

1. INTRODUCTION

The rising number of electronic systems in vehicles has led to the development of in-vehicle communication networks. These networks must provide efficient and reliable communication between Electronic Control Units (ECUs), comprised of microcontrollers, sensors, and actuators, to perform different functionalities in the system. As a prominent fieldbus, Controller Area Network (CAN) has been in use since the early 1990s in both automotive and industrial applications [1,2]. The CAN protocol applies a priority-based medium access method which guaranties immediate access to the bus for the highest priority messages in the arbitration process. However, the method can cause extensive message delays for lower priority messages especially under heavy bus load and low transmission speed conditions. One potential solution is to increase the bus transmission speed in order to reduce the message latency. However, the bit-wise arbitration mechanism in CAN limits the bus speed to 1 Mbps at maximum 40 m bus length. In order to solve the bandwidth problem, a new protocol known as CAN with flexible data-rate (CAN FD) provides bit-rates up to envisioned 8 Mbps or even experimental 15 Mbps [3,4,5].

Although the CAN FD protocol provides higher bus bit-rates for data transmission, it inherits some limitations of the standard CAN protocol such as the latency experienced by the lower priority messages since it applies the same arbitration mechanism. Real-time network traffic may consist of periodic, sporadic, or a combination of both message types. As in a CAN system, all the periodic and sporadic messages are also subject to the same priority-based arbitration mechanism in a CAN FD system. Time-triggered protocols such as FlexRay [6], and Time Triggered Protocol (TTP) [7] provide deterministic medium access in communication for real-time systems. In order to provide deterministic access in CAN, Time-Triggered CAN (TTCAN) has been introduced [8,9,10]. TTCAN provides the advantage of time-triggered access for periodic messages, and also accommodates event-triggered access for sporadic messages, but lacks the fast transmission feature of CAN FD. Time-triggered access with CAN FD results in the Time-Triggered CAN FD (TTCAN FD) protocol, which combines the fast transmission feature of CAN FD with the deterministic access feature of TTCAN.

As the main contribution, this paper introduces a method to develop a new version of the TTCAN FD protocol which transmits periodic messages faster using the arbitration-free feature of the pre-allocated exclusive-time message windows. The new protocol applies an approach which makes the transmission speed of the main fields of the periodic message frames, including the identifier field, independent from the limitations imposed by the arbitration process, which is not possible in CAN or CAN FD. This paper provides a detailed comparative performance analysis of the TTCAN FD protocol and its proposed new version. In order to investigate the performance improvements, the simulation models have been developed and performance analysis has been realized. The simulation results show that the new protocol provides considerable performance improvements and increased message transmission capacity compared to existing approaches.

The paper is organized as follows. In Section 2, basic information about CAN, CAN FD, and TTCAN is presented. Section 3 introduces the method to develop the proposed TTCAN FD protocol version. In Section 4, explanations about modeling and simulation are provided. Section 5 provides a comparative analysis of simulation results, including the worst-case and average message delays. Conclusions are given in Section 6.

2. OVERVIEW OF CAN PROTOCOL AND RELATED WORK

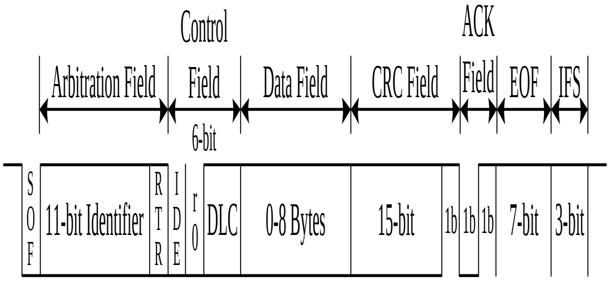

Fig 1: Message frame for standard CAN.

CAN is an event triggered asynchronous real-time network which provides low cost and robust communication among ECUs [1,2]. The CAN protocol has two types of message frames as the standard CAN frame with an 11-bit identifier, and the extended CAN frame with a 29-bit identifier. This study focuses on the CAN frame with the 11-bit identifier and uses this message standard in simulations. Fig. 1 shows the standard CAN message frame. CAN applies a priority-based medium access method, known as carrier-sense multiple-access with non-destructive bit-wise arbitration, in which every message has a unique identifier which also determines the priority of the message. If two or more message transmissions start at the same time, the priority-based bit-wise arbitration mechanism ensures that the highest priority message is transmitted first, and no time is wasted for collisions. However, the bit-rate is limited at certain bus lengths since all nodes have to observe the same bit level within one bit-time. In this mechanism, the bit level "0" has priority over "1". Therefore, the message with the lowest binary identifier value has the highest priority. In bit-wise arbitration, if a node sends the recessive bit level "1" and observes a dominant "0" on the bus, it switches to the receiver mode and re-enters the next arbitration [1].

In this study, worst-case message response time analysis is used for comparative performance analysis purposes. The worst-case message response time (R_m) [11,12], the longest time taken from the start of the task that queues the message m_m to the latest time that the message arrives in the destination, can be written as

$$R_m = J_m + W_m + C_m \qquad (1)$$

where J_m is the queuing jitter, which gives the latest queuing time of message m on the host CPU. W_m includes the queuing delay due to the higher priority messages and the blocking time due to a lower priority message that has already taken the bus since the protocol is non-preemptive. C_m is the longest time taken to transmit message m_m on the bus, which includes the time taken by the frame overheads, payload, and extra stuff bits [11,12,13]. The protocol applies a bit-stuffing rule, where an opposite polarity bit is inserted whenever five consecutive bits of the same polarity are transmitted, to provide enough edges for re-synchronization during message transmission. The message transmission time for a standard CAN frame including the worst-case bit-stuffing is

$$C_m = \left(47 + 8s_m + \lfloor (34 + 8s_m)/4 \rfloor\right)\tau_a \qquad (2)$$

where s_m is the payload size in bytes. 47 is the fixed form size of a standard CAN frame excluding the payload. ⌊(34+8sm)/4⌋ is the floor operator giving the stuff bit number for the payload and the 34-bit part of fixed form field of the frame that is subject to bit stuffing. Since the stuffed bits are also subject to bit stuffing, the following four same level data bits after the stuff bit may constitute another five same level bits sequence requiring another opposite level stuff bit. Therefore, divisor 4 is used to compute the worst-case stuff bit number [14]. τ_α is the bus bit-time (s/bit), which determines the transmission bit-rate. If the bit-rate is 1 Mbps, the bit-time is 1 μs/bit.

The CAN FD protocol improves the capability of the CAN protocol in two ways: firstly, by supporting bit-rates higher than 1 Mbps; and secondly, by supporting payloads larger than 8 bytes, up to 64 bytes. Fig. 2 shows the standard CAN FD message frame. The CAN FD message frame is divided into two phases as the arbitration-phase and the data-phase. The message is transmitted with the arbitration-phase bit-rate from the start of frame (SOF) bit to the bit-rate switch (BRS) bit. If the BRS bit is sampled recessive, the bit-rate is switched to the higher data-phase bit-rate. The bit-rate is switched back to the first configuration at the sample point of the CRC delimiter bit. The extended data length (EDL) bit indicates that the message is in the CAN FD format. The data length code (DLC) bits indicate the data field length [1,3,4].

In CAN FD, the bit-stuffing rule is applied with some changes [15,16]. In the CRC sequence of the CAN FD frame, stuff bits are inserted at fixed positions in order to improve the error detection capability [15,16]. As the CAN FD protocol applies two different bit-rates in the frame, and the bit-stuffing is also different for the CRC sequence, transmission time can be written as

$$C_m = (29 + \lfloor s_{arb}/4 \rfloor)\tau_a + (28 + 8s_m + \lfloor 8s_m/4) \rfloor + \mathrm{CRC})\tau_d$$
(3)

where s_{arb} is the number of bits subject to bit-stuffing in the first arbitration-phase. 29 is the total number of bits in the arbitration-phase, including the BRS bit and excluding the CRC delimiter bit. τ_α is the bit-time of arbitration-phase, and τ_d is the bit-time of data-phase in s/bit. The data-phase (excluding the payload) is 28 bits in length. This comprises 27 bits of a CAN FD frame plus 1 stuff bit that may be inserted due to the BRS, ESI, and DLC bits. s_m is the size of the payload in bytes. The CRC field also includes additional 3 stuff count bits with a parity bit and fixed stuff bits for improved error detection together with 17 CRC bits for the payload sizes up to 16 bytes, and 21 CRC bits for the payload sizes larger than 16 bytes [15,16].

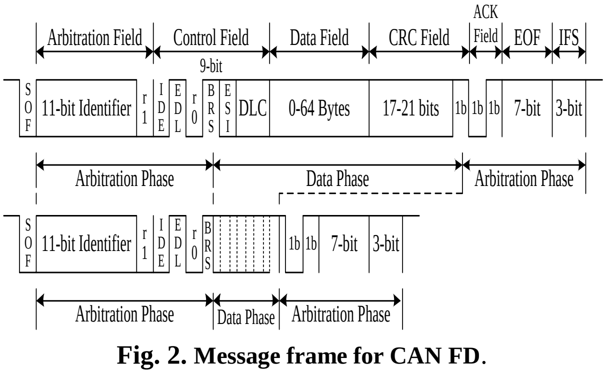

Fig. 2. Message frame for CAN FD.

As the EDL, r0, and BRS bits contain 1, 0, and 1, respectively, 13 bits remain for bit stuffing in the field before the first bit-rate switch [16]. Considering the worst-case bit stuffing in this field, the ⌊sarb/4⌋ operator can produce a maximum of 3 stuff bits, resulting in 32 bits. Therefore, the equation can be simplified as

$$C_m = 32\,\tau_a + (28 + 10\,s_m + \mathrm{CRC})\tau_d \qquad (4)$$

TTCAN provides a deterministic feature for the CAN systems by applying a TDMA structure to the CAN protocol. TTCAN corresponds to the session layer of the Open System Interconnection (OSI) reference model, whereas CAN is implemented in the data link and physical layers of the model [8]. All nodes are synchronized by a periodically transmitted reference message. Time synchronization is applied at two levels. Level 1 provides synchronization for the time-triggered scheduling by Network Time Unit (NTU) based on the network bit-time. Level 2 provides a high precision global time base, which allows TTCAN to synchronize and interface to other networks [8].

In TTCAN, all messages are transmitted in a fixed sequence of time windows comprising the system matrix (SM), also known as the matrix cycle (MC), repeated cyclically. Each node has the predefined schedule of the messages, and each message is transmitted in pre-allocated time slots [8]. Fig. 3 shows an example of the TTCAN system matrix. The SM is composed of Basic Cycles (BCs) which start with a reference message sent periodically by the Time Master node. Each Transmission Column (TC) containing time windows starts with a Time Mark (TM). The TTCAN system matrix is comprised of exclusive, arbitrating, and free time windows. An exclusive-time window is reserved only for a specific periodic message. An arbitrating-time window is designed to transmit sporadic messages with priority-based arbitration mechanism. The free-time windows are reserved for future extensions.

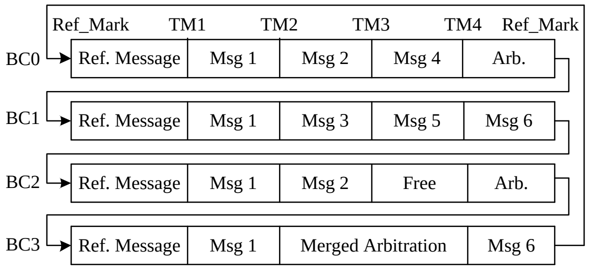

Fig 3: An example of TTCAN matrix cycle.

3. PROPOSED TTCAN FD MESSAGE FRAME

TTCAN applies a TDMA structure over the existing CAN protocol. Similarly, the same method can be applied to the CAN FD protocol leading to the TTCAN FD protocol. In this way, the time-triggered access with a fast transmission feature can be achieved. In this study, the TTCAN FD protocol implementation is realized in two stages. In the first stage, CAN FD frames are transmitted in a TTCAN system matrix. The standard time-triggered and event-triggered CAN FD frames are allocated in the exclusive and arbitrating-time windows, respectively. This stage presents the current state of the protocol which can readily be applied to the existing standard CAN FD frames. The current protocol structure is called TTCAN FD1 in this study. There are already some emerging hardware products promoting the application of this protocol stage [17,18].

In CAN FD, the identifier field has to be transmitted with a limited bit-rate due to the priority-based bit-wise arbitration used for medium access. However, in TTCAN FD, the exclusive-time windows provide an opportunity for periodic messages. Since periodic messages are pre-allocated in the exclusive-time windows, there is no need for the arbitration process to access the bus. This allows a higher transmission bit-rate to be used for the identifier field. Therefore, the main part of the CAN FD frame, from the SOF bit to the CRC-delimiter bit, can be transmitted with the fast data-phase transmission speed, termed as the high bit-rate phase. As the main contribution of the study, this process constitutes the second stage development of the protocol and called TTCAN FD2. This feature is not possible in CAN or CAN FD, and it has been implemented first time in this study. Although some recent commercial products include the TTCAN architecture with CAN FD [17,18], they do not provide the proposed feature. As an important advantage of this method, the transmission speed is no longer limited by the arbitration process for periodic messages, and the main parts of the messages can be transmitted with higher speeds that the physical layer properties allow. Consummately, the fields related to the actual data can be transmitted with higher bit-rates. The only remaining part of the frame transmitted with the low arbitration-phase bit-rate, termed as the standard bit-rate phase, is the last 13-bit part of the frame. In this study, this field is left to be transmitted with low arbitration-phase bit-rate to keep the usual acknowledgement feature of the frame, providing data consistency within the whole system, and also to allow processing time for the received frame before the next message is transmitted on the bus. Fig. 4 shows the message frame format for TTCAN FD2.

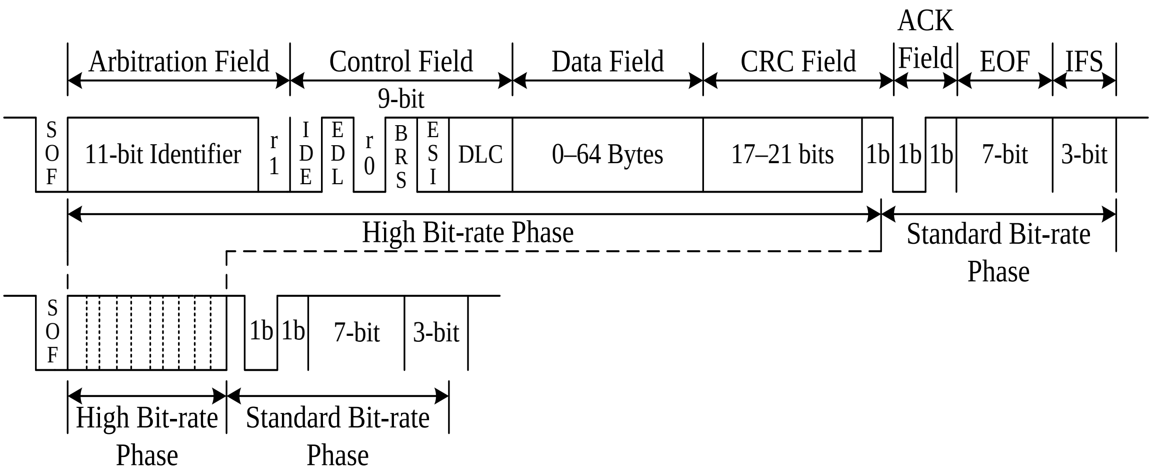

Fig. 4. Message frame for the second development stage of TTCAN FD.

Equation (3) applies to the message transmission time for TTCAN FD1 since the same standard CAN FD frame is used in this stage. Equation (3) is also valid for all sporadic messages. However, for periodic messages in TTCAN FD2, the transmission time can be written as

$$C_m = 13\tau_a + (16 + \lfloor s_{arb}/4 \rfloor + 28 + 8s_m + \lfloor 8s_m/4) \rfloor + \mathrm{CRC})\tau_d \qquad (5)$$

Equation (5) has the same properties as (3), whereas the initial 16-bit part of the frame, which is also the first part of arbitration-phase, is moved to the following data-phase section of the equation. Equation (5) can be simplified as

$$C_m = 13\tau_a + (47 + 10\, s_m + \mathrm{CRC})\tau_d \qquad (6)$$

4. MATRIX CYCLE CONSTRUCTION AND SIMULATION MODEL

The message scheduling method determines the system performance. Since the dense allocation method creates long queuing delays for sporadic messages due to the consecutively allocated exclusive-time windows, the sparse allocation method has been used in this study [19]. It is a common method to set the BC time (T_{BC}) equal to the least time-triggered message period time (pm_1). The Least Common Multiple (LCM) of all periodic messages can be used to assign the matrix cycle time (T_{MC}). However, since the number of BCs in a system matrix must be 2^j with the condition $0 \leq j \leq 6$, it may not always be possible to use the LCM method for the matrix cycle time as the required BC number may exceed the limit. The reduced matrix cycle method, introduced in [20] and also applied in [21], provides a solution to this problem. In a matrix cycle, ideally, all time-triggered message periods are expected to be the 2^j multiple of pm_1, where $0 \leq j \leq 6$, for a smooth message scheduling. In the method introduced in [21], the message periods any multiples of pm_1 can also be scheduled smoothly.

In this study, the reduced matrix cycle with sparse allocation method has been applied to TTCAN FD messages based on the SAE benchmark message set [21]. As in some previous studies [11,20,21], the message set has also been used in this study for performance analysis. Table 1 shows the piggybacked SAE benchmark message set [11], where messages are given in priority order from the highest (M_1) to the lowest (M_{17}). Messages from M_2 to M_6, and from M_{12} to M_{17} are time-triggered and produced periodically. Messages M_1 and M_{11} are event-triggered, and messages from M_7 to M_{10} are piggybacked sporadic messages, also modelled as event-triggered with the exponential distribution function. T is the message period, and D is the deadline in milliseconds.

Table 1. SAE benchmark based message set.

Message	Size (bytes)	T (ms)	D (ms)
M_1	1	50.0	5.0
M_2	2	5.0	5.0
M_3	1	5.0	5.0
M_4	2	5.0	5.0
M_5	1	5.0	5.0
M_6	2	5.0	5.0
M_7	6	10.0	10.0
M_8	1	10.0	10.0
M_9	2	10.0	10.0
M_{10}	3	10.0	10.0
M_{11}	1	50.0	20.0
M_{12}	4	100.0	100.0
M_{13}	1	100.0	100.0
M_{14}	1	100.0	100.0
M_{15}	3	1000.0	1000.0
M_{16}	1	1000.0	1000.0
M_{17}	1	1000.0	1000.0

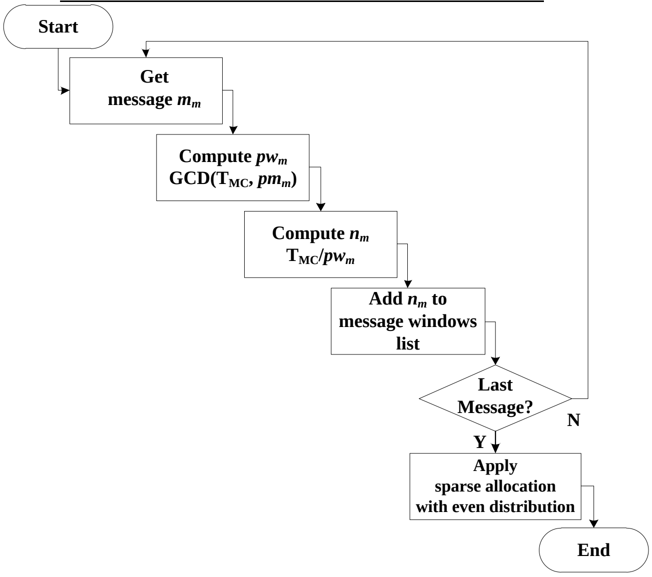
Start
Get message m_m
Compute pw_m GCD(T_{MC}, pm_m)
Compute n_m T_{MC}/pw_m
Add n_m to message windows list
Last Message?
N
Y
Apply sparse allocation with even distribution
End

Fig. 5. Message scheduling algorithm with reduced matrix cycle.

In the matrix cycle design, the BC time is set to T_{BC} = 5 ms, equal to the least period time of the time-triggered messages pm1. The number of the BCs is set to L=4, and therefore, the matrix cycle time is T_{MC} = 20 ms. The number of exclusive-time windows for the periodic messages are computed for scheduling according to the reduced matrix cycle algorithm [21]. However, a more advanced and simplified form of the algorithm is proposed and used in this study to determine the number of the exclusive-time windows to be allocated in the system matrix. Therefore, this algorithm provides scheduling results with less computation, reducing the amount of processing required. The sparse allocation part of the algorithm is applied as explained in [21].

Fig. 5 shows the algorithm with the modified reduced matrix computation part. In the modified algorithm, the Greatest Common Divisor (GCD) function plays a key role to produce the matrix cycle. The repetition period of an exclusive window in the system matrix for a time-triggered message m_m is computed as pw_m = GCD(T_{MC}, pm_m). The number of exclusive windows n_m, for a periodic message m_m, can be calculated as $n_m = T_{MC}$ / pw_m. This process is repeated for every periodic message in the message set until the complete exclusive-time message windows list is prepared. Fig. 6 provides the matrix cycle design for the TTCAN FD simulation model.

Ref + M2	Arb	M3	Arb	M4	Arb	M5	Arb	M6	Arb	M12	Arb	M13	Arb
Ref + M2	Arb	M3	Arb	M4	Arb	M5	Arb	M6	Arb	M15	Arb	M14	Arb
Ref + M2	Arb	M3	Arb	M4	Arb	M5	Arb	M6	Arb	Free	Arb	M16	Arb
Ref + M2	Arb	M3	Arb	M4	Arb	M5	Arb	M6	Arb	Free	Arb	M17	Arb

Fig. 6. Matrix cycle for the TTCAN FD simulation model.

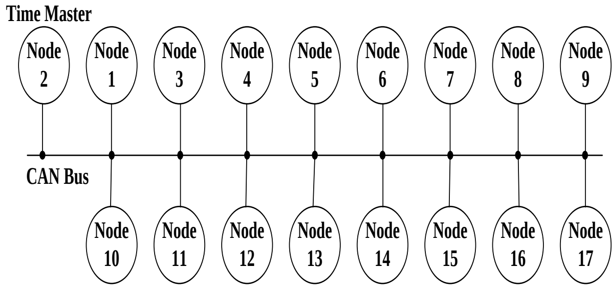

Fig. 7. Simplified representation of the simulation model.

In order to investigate the comparative performance of the new protocol, the simulation models for the CAN FD and TTCAN FD protocols have been developed with the SimEvents toolbox of Matlab Simulink software package [22,23]. In the TTCAN FD model, in order to provide time synchronization for the network nodes, a Time Master (Node 2) has been assigned and the level-2 synchronization is applied. Fig. 7 illustrates the simplified representation of the models used for CAN FD and TTCAN FD, where the Time Master is only used in the TTCAN FD models. Each node in the model produces one message of the set, and each message is produced with the node which corresponds to its priority number shown in Table 1. The designed models are simulated with 1 Mbps, 500 kpbs, 250 kbps, and 125 kbps standard CAN protocol transmission bit-rates for the arbitration-phase, and the high bit-rate of 5 Mbps is applied for the data-phase since most emerging CAN FD products also support this bit-rate [5,24].

5. PERFORMANCE ANALYSIS

Various studies have investigated the performance of the CAN FD protocol [25,26,27]. This study focuses on the performance analysis and improvements of the TTCAN FD1 protocol and proposed TTCAN FD2 protocol. The worst-case message delay is analyzed as an important parameter in the performance analysis of real-time systems. The worst-case delay indicates the ability of the system to meet message deadlines. The average message delay values are also analyzed in order to investigate typical latency and efficiency.

5.1 Worst-case Message Delay Analysis

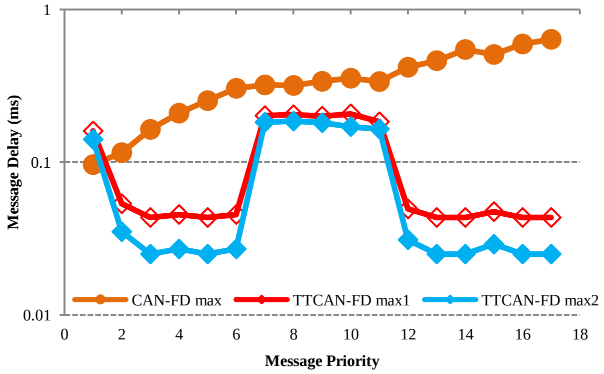

Fig. 8. Worst-case message delays at 1 Mbps arbitration-phase bit-rate.

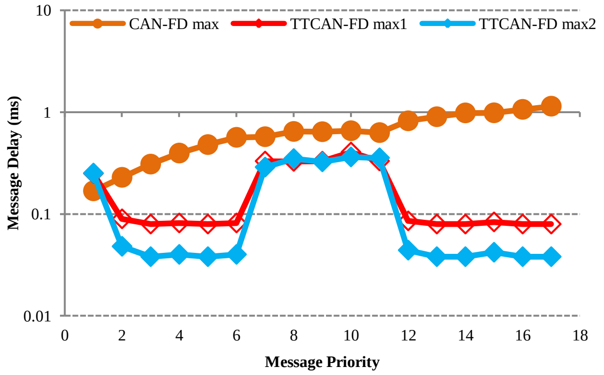

Fig. 9. Worst-case message delays at 500 kbps arbitration-phase bit-rate.

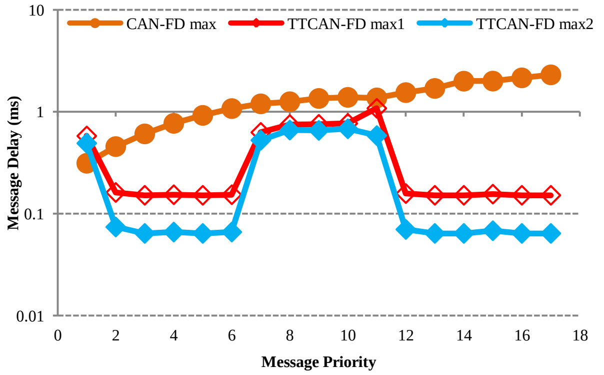

Fig. 10. Worst-case message delays at 250 kbps arbitration-phase bit-rate.

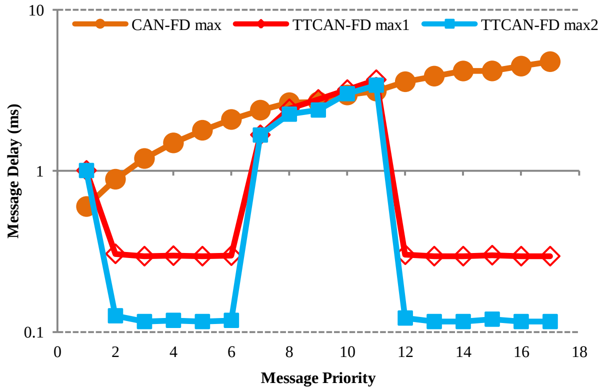

Fig. 11. Worst-case message delays at 125 kbps arbitration-phase bit-rate.

The worst-case message delay, also known as the worst-case response time R_m, is considered as the longest time between the start of queuing message m_m and the latest time that the message arrives at the destination node, which includes W_m and C_m. The worst-case delay graphs provide results for the whole message set. Each message's worst-case delay is mostly affected by its priority, which determines the queuing delay (W_m) caused by the arbitration process. In this study, the worst case message delay for message m_m is obtained with the highest delay encountered by that message throughout the simulation.

Figures 8, 9, 10, and 11 show the worst-case message delays for 1 Mbps, 500 kbps, 250 kbps, and 125 kbps arbitration-phase bit-rates, respectively. Since there is a big difference between the lowest and highest values in the figures, the graphs are formatted with logarithmic scales. The graphs illustrate the message delays with descending message priorities from 1 to 17. The graph "CAN-FD max" illustrates the worst-case message delays where the messages are transmitted according to their priority order with the arbitration mechanism. The graph "TTCAN-FD max1" shows the worst-case message delays for TTCAN FD1, where the standard CAN FD frames are assigned in the message windows. The graph "TTCAN-FD max2" illustrates the worst-case message delays for TTCAN FD2, which provides faster transmission speed for the periodic messages.

The effect of arbitration can be clearly seen on the CAN FD worst-case message delay graphs for all bus transmission speeds. In the CAN FD delay graphs, message 1 has the lowest message delay, since it is transmitted first in the arbitration process as it has the highest priority. Although the TTCAN FD models have the optimum arbitration-time windows allocation with evenly distributed time windows for sporadic messages, even the highest priority sporadic message may still face a queuing delay if it becomes ready during the transmission of an exclusive time-window. The effect of this queuing delay can be seen from the graphs of the TTCAN FD models, where the first priority message has slightly higher worst-case message delays than those of the CAN FD models. However, the TTCAN FD models show better worst-case message delay performances for the event-triggered messages from priority 7 to 11 at almost all bus bit-rates. The only case where the CAN FD and TTCAN FD models have close worst-case message delay values is at the 125 kbps arbitration-phase bit-rate, where the arbitrating-time window sizes become less than those at higher bit-rates since the exclusive-time window sizes are increased due to the lower transmission bit-rate, leaving less space for the arbitrating-time windows.

Better performance improvements are achieved with the TTCAN FD models. Since the periodic messages are transmitted synchronously in pre-allocated exclusive-time windows without arbitration or bus queuing delays, all the time-triggered messages have immediate access to the bus. This can be observed in the worst-case delay graphs for the periodic messages with priorities from 2 to 6 and from 12 to 17. These delay values show that TTCAN FD provides better performance for time-triggered messages, where even the lowest priority message has immediate bus access. With the second development stage, the TTCAN FD2 protocol has further performance improvements as can be seen in the "TTCAN-FD max2" graphs. The improvement is obtained by transmitting the first arbitration-phase field of the CAN FD frame with high transmission speed. With this process, the message transmission times of periodic messages are further reduced. The difference between the "TTCAN-FD max1" and "TTCAN-FD max2" graph values become more distinct as the ratio of the data-phase bit-rate to the arbitration-phase bit-rate increases. This can be seen from Fig. 8 through Fig. 11, where the ratio and consequently the difference between two graph values increase.

The main contributor in worst-case message delay values is the queuing delay (W_m) caused by the priority-based arbitration process. In addition, the transmission delay (C_m) has also a slight effect. As can be seen from the graphs, the higher priority messages experience smaller arbitration delays. An increase can be observed in the worst-case graphs through the lower priority messages in CAN FD models due to the arbitration process. However, the main contributor of the worst-case message delays in periodic messages in TTCAN FD models is the transmission delay (C_m), and there is no queuing delay due to arbitration since all the periodic messages are transmitted in their exclusive-time windows. This effect can be clearly seen on all periodic messages in the TTCAN FD models.

In order to investigate the performance improvements in more detail, the delay performance ratio (Pr_m) of each periodic and sporadic message m_m in CAN FD to TTCAN FD is computed. The message delay performance ratio (Pr_m) for a message can be described as the ratio of delay for the message m_m in one system model to the delay of the same message in another system. The worst-case delay performance ratio for each message m_m is calculated as R_m(CAN FD) / R_m(TTCAN FD1), and R_m(CAN FD) / R_m(TTCAN FD2). Then, the mean value of the delay performance ratios (Pr_m) of all messages in the set is computed. The average values of Pr_m ratios are compared at four arbitration bit-rates in order to analyze the improvements of the TTCAN FD protocols against the CAN FD protocol. For periodic messages, from 1 Mbps to 125 kbps arbitration bit-rates, the average performance ratio of TTCAN FD1 against CAN FD ranges from 8.54 to 9.93, whereas the ratio ranges from 14.51 to 25.07 for TTCAN FD2. The average Pr_m ratio of TTCAN FD2 against TTCAN FD1, (R_m(TTCAN FD1) / R_m(TTCAN FD2)), ranges from 1.68 to 2.52, that is, TTCAN FD2 worst-case message delay is up to 2.52 times smaller on average.

For the worst-case message delays of sporadic messages, the average Pr_m ratio of TTCAN FD1 against CAN FD ranges from 0.98 to 1.64, whereas the ratio ranges from 1.04 to 1.87 for TTCAN FD2. The average performance ratio of TTCAN FD2 against TTCAN FD1 ranges from 1.02 to 1.27. The results show that time-triggered architecture provides significant worst-case delay performance improvements for periodic messages and smaller performance improvements for sporadic messages. As can be seen from the results, TTCAN FD2 yields greater performance improvements than TTCAN FD1.

5.2 Average Message Delay Analysis

The average delay for message m_m is obtained by averaging all the delays encountered by that message throughout the simulation. Therefore, each message has a specific average message delay value. As in the worst-case delay values, the time between the start of queuing message m_m and the time that the message arrives at the destination node is used in average message delay calculations. The average message delay performances of the protocol models are presented in Fig. 12 to Fig. 15. In order to analyze the performance improvements obtained with the proposed protocol, the average message delay values of the TTCAN FD models are compared with the CAN FD model results. In the figures, the

"CAN-FD avrg" graph shows the average message delays for the CAN FD models. The "TTCAN-FD avrg1" and "TTCAN-FD avrg2" graphs illustrate the average message delay results for the TTCAN FD1 and TTCAN FD2 models respectively.

The figures show that due to the arbitration mechanism, there is no advantage for the time-triggered messages over the event-triggered messages in CAN FD. However, the TTCAN FD protocol provides better message delay performances for time-triggered messages. The graphs for the TTCAN FD models show that the periodic messages, from 2 to 6, and 12 to 17, have improved performances. Only M2 in TTCAN FD1 has slightly higher message delays than those of CAN FD at 1 Mbps and 500 kbps bit-rates. This is because M2 carries additional 4 bytes reference message data. However, M2 delays in TTCAN FD1 are smaller at 250 kbps and 125 kbps bit-rates than those of CAN FD as the arbitration delays become higher.

The main contributor in average message delay values is the message transmission time (C_m). Although the periodic messages are solely affected by the transmission time, the slight effect of queuing delay (W_m) caused by arbitration can also be observed in sporadic messages in TTCAN FD, and in both periodic and sporadic messages in CAN FD. Since the arbitration delays have little effect on sporadic messages at 1 Mbps, and the message delays are mostly affected by transmission delays, close sporadic message delay values are observed in graphs for the models, which is more obvious with the message M7. As can be seen from Fig. 12, the average delay for the sporadic message M7 has the highest value since it has the largest payload size in the message set, causing the longest transmission time. This effect is very clear at 1 Mbps, and also at 500 kbps with slight arbitration delay differences. However, at 250 kbps and 125 kbps, the effect of arbitration, and consecutively the queuing delay, increases in addition to the transmission delay.

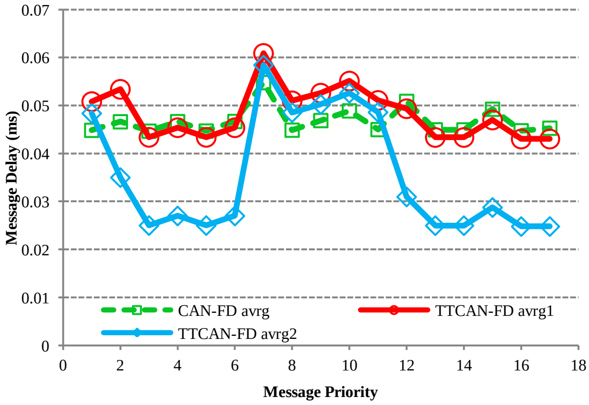

Fig. 12. Average message delays at 1 Mbps arbitration-phase bit-rate.

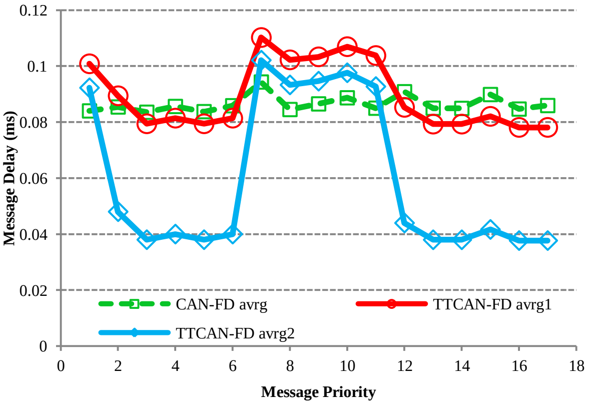

Fig. 13. Average message delays at 500 kbps arbitration-phase bit-rate.

Therefore, the message delays in sporadic messages increase especially through the lower priorities at lower bit-rates in the TTCAN FD models. A similar effect can also be observed with both periodic and sporadic messages in the CAN FD model. In general, while CAN FD provides slightly better performance for sporadic messages, TTCAN FD has better performances with periodic messages. From the graphs, it can be clearly seen that TTCAN FD2 has the advantage of lower transmission delay with periodic messages, which have the smallest message delays in figures.

The performance ratio (Pr_m) is also computed for average message delay values. For periodic messages, the average performance ratio against CAN FD ranges from 1.02 to 1.35 for TTCAN FD1, whereas, it ranges from 1.72 to 3.41 for TTCAN FD2. The sporadic message delays in TTCAN FD are greater than those observed for CAN FD. For sporadic messages, the average performance ratio against CAN FD ranges from 0.69 to 0.89 for TTCAN FD1, while it ranges from 0.80 to 0.93 for TTCAN FD2. This is the effect of queuing delay caused by the exclusive-time windows which cause the event-triggered messages experience more delays in TTCAN FD. The arbitration also causes more a noticeable increase in the TTCAN FD average sporadic message delay values at 125 kbps. This is because the time-space for arbitrating-time windows is decreased at this bit-rate. Therefore, the lowest priority message, M11, in the arbitration process experiences the highest message delay. Although the TTCAN FD model delays have similar characteristics in the graphs, the TTCAN FD2 model has lower average message delays.

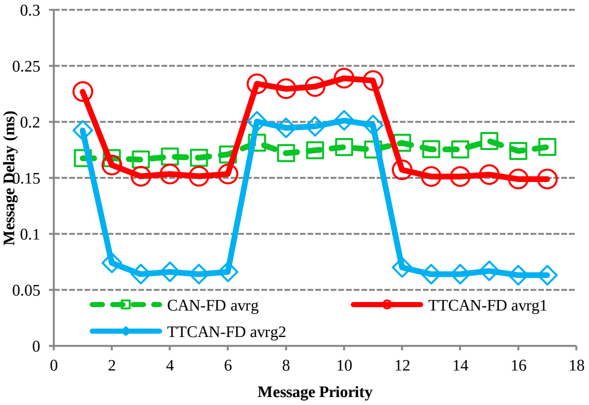

Fig. 14. Average message delays at 250 kbps arbitration-phase bit-rate.

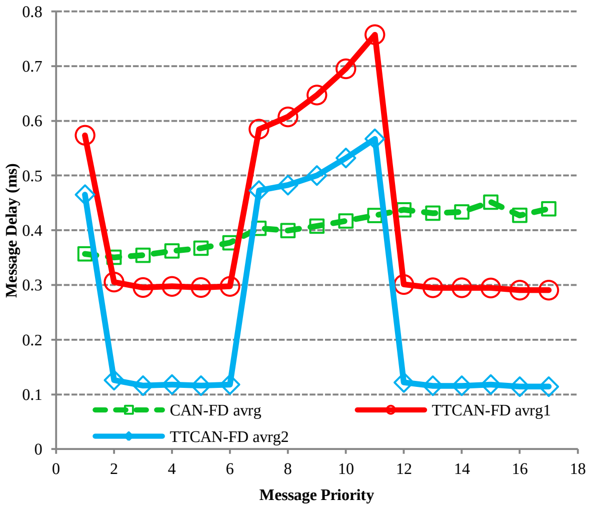

Fig. 15. Average message delays at 125 kbps arbitration-phase bit-rate.

6. CONCLUSIONS

In this study, a new version of the TTCAN FD protocol has been introduced to provide better performance for CAN FD frames. The TTCAN FD protocol development has been described in two stages. The current version of the TTCAN FD protocol constitutes the first stage, which applies time-triggered scheduling to the CAN FD messages. In the proposed second stage, the first arbitration-phase of the frame is also transmitted at high data-phase bit-rate, making the protocol arbitration independent. The second development stage applies a modification for periodic messages by transmitting the first arbitration-phase field also with the high bit-rate of data-phase.

In order to investigate the performance improvements obtained with the introduced TTCAN FD protocol version, the simulation models with the SAE benchmark based message set have been developed, and the results have been analyzed at different bus speed conditions. Compared to the CAN FD worst-case periodic message delay values, TTCAN FD1 can transmit messages 8 to 10 times faster, while TTCAN FD2 can transmit messages 14 to 25 times faster. These results are for message frames with 11-bit identifiers. In the case of 29-bit identifiers, even higher performance results can be expected since the ratio will be higher with the increased number of identifier bits to be sent with the high transmission bit-rate.

The simulation results have shown that the TTCAN FD protocols provide better performances than the CAN FD protocol, and with TTCAN FD2, it is also possible to create a protocol independent from arbitration limitations to achieve comparatively higher message transmission speeds in order to meet the real-time control system requirements.

7. ACKNOWLEDGMENTS

This research was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) 2219-International Postdoctoral Research Fellowship Program.

8. REFERENCES

[1] ISO 11898-1:2015 - Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling, Second edition, International Organization for Standardization, 2015.

[2] N. Navet, CAN in automotive applications: a look forward, in: Proc. 13th Int. CAN Conf., 2012.

[3] F. Hartwich, CAN with flexible data-rate, in: Proc. 13th Int. CAN Conf., 2012: p. 14_1 – 14_9.

[4] Bosch, CAN with flexible data-rate, Specification version 1.0, Robert Bosch GmbH, 2012.

[5] Microchip Technology Inc., MCP25612FD Dual CAN Flexible Data-Rate Transceiver, (2015). ww1.microchip.com/downloads/en/DeviceDoc/20005409A.pdf.

[6] R. Makowitz, C. Temple, Flexray - A communication network for automotive control systems, in: Proc. IEEE Int. Work. Fact. Commun. Syst., IEEE, 2006: pp. 207–212. doi:10.1109/WFCS.2006.1704153.

[7] H. Kopetz, G. Bauer, The time-triggered architecture, Proc. IEEE. 91 (2003) 112–126.

[8] G. Leen, D. Heffernan, TTCAN: A new time-triggered controller area network, Microprocess. Microsyst. 26 (2002) 77–94.

[9] ISO 11898-4 - Road vehicles –Controller area network (CAN) Part 4: Time-triggered communication, International Organization for Standardization, (reviewed in 2013), 2004.

[10] Robert Bosch GmbH, M_TTCAN Time-triggered Controller Area Network User’s Manual, Revision 3.2.1.1, 2016.

[11] K. Tindell, A. Burns, Guaranteeing message latencies on control area network (CAN), in: Proc. 1st Int. CAN Conf., 1994: pp. 1–11.

[12] K.W. Tindell, A. Burns, A.J. Wellings, Calculating controller area network (CAN) message response times, Control Eng. Pract. 3 (1995) 1163–1169.

[13] R.I. Davis, A. Burns, R.J. Bril, J.J. Lukkien, Controller area network (CAN) schedulability analysis: refuted, revisited and revised, Real-Time Syst. 35 (2007) 239–272.

[14] N. Navet, Q.Y. Song, F. Simonot, Worst-case deadline failure probability in real-time applications distributed over controller area network, J. Syst. Archit. 46 (2000) 607–617.

[15] A. Mutter, CAN FD and the CRC issue, CAN Newsl. (2015) 4–10.

[16] U.D. Bordoloi, S. Samii, The Frame Packing Problem for CAN-FD, in: Proc. IEEE Real-Time Syst. Symp., 2014: pp. 284–289.

[17] CAST Inc., CAN FD Bus Controller IP Core Gains Time-Triggered TTCAN Capability, (2016). www.cast-inc.com/news/2016-02-23-can-fd-bus-controller-ip-core-gains-time-triggered-ttcan-capability.

[18] NXP Semiconductors, MPC5777M: Ultra-Reliable MPC5777M MCU for Automotive & Industrial Engine Management, (2016). www.nxp.com/MPC5777M.

[19] A. Albert, R. Hugel, Heuristic scheduling concepts for TTCAN networks, in: Proc. 10th Int. CAN Conf., 2005: p. 01_9 – 01_17.

[20] K. Schmidt, E.G. Schmidt, Systematic message schedule construction for time-triggered CAN, IEEE Trans. Veh. Technol. 56 (2007) 3431–3441.

[21] M. Tenruh, Message scheduling with reduced matrix cycle and evenly distributed sparse allocation for time-triggered CAN, J. Netw. Comput. Appl. 34 (2011) 1240–1251.

[22] MathWorks, Simulink, (2017). www.mathworks.com.

[23] F. Li, L. Wang, C. Liao, CAN (Controller Area Network) bus communication system based on Matlab/Simulink, in: Proc. 4th Int. Conf. Wirel. Commun., Netw. Mob. Comput., IEEE, 2008: pp. 1–4. doi:10.1109/WiCom.2008.1004.

[24] Texas Instruments, TCAN33x 3.3-V CAN Transceivers with CAN FD (Flexible Data Rate), (2016). www.ti.com/lit/ds/symlink/tcan337.pdf.

[25] T. Nguyen, B.M. Cheon, J.W. Jeon, CAN FD performance analysis for ECU re-programming using the CANoe, in: Proc. 18th IEEE Int. Symp. Consum. Electron., IEEE, 2014: pp. 1–5. doi:10.1109/ISCE.2014.6884472.

[26] S.J. Jang, J.W. Jeon, Software reprogramming performance analysis of CAN FD and FlexRay protocols, in: Proc. IEEE Int. Conf. Informat. Autom., IEEE, 2015: pp. 2535–2540. doi:10.1109/ICInfA.2015.7279712.

[27] M. Tenruh, P. Oikonomidis, P. Charchalakis, E. Stipidis, Modelling, simulation, and performance analysis of a CAN FD system with SAE benchmark based message set, in: Proc. 15th Int. CAN Conf., 2015: p. 7_12 – 7_19.