Modelling, simulation, and performance analysis of a CAN FD system with SAE benchmark based message set

Mahmut Tenruh, Panagiotis Oikonomidis, Periklis Charchalakis, Elias Stipidis
Mugla S. K. University, TR; visiting researcher, VRC, University of Brighton, UK
Vetronics Research Centre (VRC), University of Brighton, UK

Abstract. The increasing amount of electronic control units (ECUs) and message traffic in in-vehicle and industrial communication networks cause a rising demand for higher bandwidth and low message response times. As a prominent real-time distributed control network, CAN responds to this demand with the CAN FD protocol which provides improved bandwidth and payload capacities. CAN FD provides performance improvements mainly in two ways: firstly, by increasing the bit-rate in the data-phase of the frame, which results in faster message transmission; secondly, by increasing the payload size, which provides better payload to overhead ratio. In a control network, it is crucial to meet the critical timing requirements of hard real-time tasks. Simulating a real-time system is an important process to investigate the timing behaviour and analyse the performance of system. This study provides the performance analysis of a CAN FD system with SAE Benchmark based message set, comprised of both time-triggered and event-triggered messages. In order to investigate the performance improvements, the models for both CAN and CAN-FD systems have been developed in Matlab Simulink environment. The message delay and bus utilisation results obtained with the simulation models show that the CAN-FD protocol provides high performance improvements to meet the requirements of real-time control systems.

Introduction

Controller Area Network (CAN) is an asynchronous serial communication bus initially introduced for in-vehicle communication applications. Due to its low cost and robust protocol structure it has also become a widespread fieldbus in industrial applications. CAN applies a priority based medium access method which guaranties immediate bus access to the bus for the highest priority messages. However, the lower priority messages may experience extensive bus access delays especially under heavy bus load and low transmission bit-rate conditions. Increasing amount of Electronic Control Units (ECUs) and message traffic require higher transmission speeds in the network. However, the CAN protocol limits the message transmission speed to 1 Mbps at maximum 40 m bus length due to its priority based arbitration mechanism and acknowledgement method [1].

The protocols like Time Triggered Protocol (TTP) [2] and Flexray [3] use higher transmission speeds in order to meet the real-time communication requirements. In order to meet the demand for higher transmission speed in CAN, the CAN FD protocol has been developed [4][5]. CAN FD improves the protocol capabilities by using higher bit-rates to transmit the payload and, by increasing the payload size. In this way, CAN FD provides increased bandwidth capacity and better data to overhead ratio. CAN FD uses the existing CAN environment without requiring significant changes in hardware and software structure in order to provide a smooth transition from CAN to CAN FD.

It is an important subject to model and simulate a system in order to evaluate its behaviour and performance. In this study, a CAN FD system with the SAE benchmark based message set is modelled and simulated. In order to compare the performance improvements, a CAN model with the same message set has also been developed. The simulation results are investigated to evaluate the performance improvements achieved by CAN FD. The results show that CAN FD provides considerably improved performance results for the system in order to meet real-time requirements.

CAN and CAN FD overview

CAN basics

CAN is a prominent real-time network technology in automotive and industrial applications. The protocol has standard CAN and extended CAN frame formats, which have 11 bit and 29 bit identifiers respectively. In this study, the standard CAN protocol is used in simulation models. Figure 1 shows the standard CAN message frame. The CAN protocol applies a medium access method known as CSMA/CA with non-destructive bit-wise arbitration. In this mechanism, every message has an identifier which also indicates the priority of the message. In case of two or more messages starting transmission at the same time, the arbitration mechanism ensures that the highest priority message has the immediate access, and the lower priority messages are transmitted later according to their priorities. This method provides immediate access for the highest priority messages, however the lower priority messages may experience long delays. The response time for message m_m is

$$R_m = J_m + W_m + C_m$$

where Jm is the queuing jitter, the latest queuing time of message mm on the host CPU. Wm 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. Cm is the time taken to transmit message mm on the bus [6][7][8].

The CAN protocol is subject to bit stuffing in order to keep synchronisation during message transmission. According to bit-stuffing rule, after the transmission of every five bits with the same polarity, a bit with the opposite polarity is inserted. Therefore, the message transmission time (Cm) with worst-case bit stuffing for a standard CAN frame is

$$C_m = \left(47 + 8s_m + \left\lfloor \frac{34 + 8s_m}{4} \right\rfloor\right) \tau_a$$

where sm is the payload size in bytes. 47 is the fixed form size of a CAN frame excluding the payload. $\lfloor(34 + 8s_m)/4\rfloor$ 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, and τα is the bus bit-time [6].

The increasing amount of ECUs and message traffic make it necessary for the CAN protocol to use higher transmission bit-rates in order to meet the real-time application requirements. However, the transmission speed is limited to 1 Mbps for the CAN protocol.

CAN FD basics

In order to provide higher transmission speeds to solve the bandwidth problem in CAN, the CAN FD protocol has been introduced [4][5]. The CAN FD protocol improves the system performance in two ways: firstly, by using higher bit-rates to transmit the payload, and secondly, by increasing the payload size up to 64 bytes. The bus speed is limited for the CAN protocol since the bit-wise arbitration mechanism and acknowledgement method do not allow higher transmission bit-rates on the bus. However, the payload can be transmitted with higher bit-rates in order to achieve higher transmission speeds. Therefore, the CAN FD protocol divides the message frame into two phases as the Arbitration-phase and the Data-phase. Figure 2 shows the structure of a message frame with 11-bit identifier.

The arbitration phase of the message is transmitted with standard CAN bit-rate from the Start Of Frame (SOF) bit to the Bit-Rate Switch bit (BRS). The transmission speed is switched to the higher data-phase bit-rate at the sample point of BRS bit on sampling a recessive level. At the sample point of CRC delimiter, the bus speed is switched back to the arbitration-phase bit-rate. The Extended Data Length (EDL) bit indicates that the frame is in the CAN FD format.

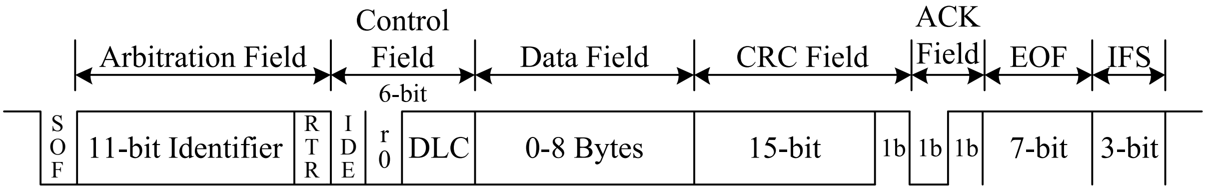

Figure 1: Standard CAN message frame.

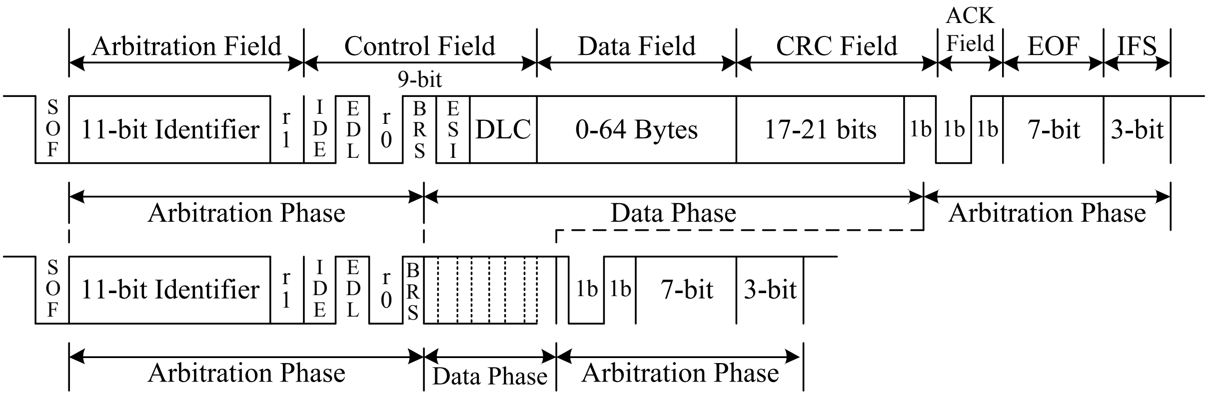

Figure 2: CAN FD message frame.

The bit stuffing process is also applied in CAN FD protocol with minor changes. In the CRC field, fixed stuff bits are used in order to improve the error detection capability of the protocol [4][9]. In CAN FD, CRC field is 17 bits plus 4 fixed stuff bits for payload sizes up to 16 bytes, and 21 bits plus 5 fixed stuff bits for payload sizes larger than 16 bytes. Therefore, the message transmission time with worst-case bit-stuffing [10] can be written as

$$C_m = \left(29 + \left\lfloor \frac{s_{arb}}{4} \right\rfloor\right) \tau_a + \left(28 + 8s_m + \frac{8s_m}{4} + CRC\right) \tau_d$$

where s_{arb} is the number of bits subject to bit-stuffing in the first arbitration-phase, where the second arbitration phase is not subject to bit stuffing. The total number of bits in the arbitration-phase is 29, including the BRS bit and excluding the CRC delimiter bit. τ_α is the arbitration-phase bit-time. The data-phase, excluding the payload, is 28 bits in length, which comprises 27 bits of a CAN FD frame plus 1 stuff bit that may be inserted due to the BRS, ESI, and DLC bits. sm is the payload size in bytes. τ_α is the data-phase bit-time. The EDL, r0, and BRS bits contain 1, 0, and 1, respectively, remaining 13 bits for bit stuffing in the field before the first bit-rate switch [10]. In the worst-case bit stuffing, the $\lfloor s_{arb}/4 \rfloor$ operator can produce 3 stuff bits, resulting in 32 bits.

Therefore, the equation can be simplified as

$$C_m = 32\ \tau_a + (28 + 10s_m + CRC)\ \tau_d$$

Although the bit stuffing process increases message size and consequently message transmission time, it is necessary in order to keep synchronisation during transmission. There are some studies suggesting solutions to decrease the effect of bit stuffing such us in [11].

Modelling and Simulation

In order to investigate the system performance improvements with the CAN FD protocol, the CAN and CAN FD models with the SAE benchmark based message set have been developed. The SAE benchmark message set originally includes 53 signals. In [6], the signals have been piggybacked and the set includes 17 messages. Table 1 shows the SAE benchmark based message set. The set includes piggybacked messages of the SAE benchmark, comprised of both periodic and sporadic messages [6]. Message M1 and M11 are event-triggered, and messages from M7 to M10 includes event-triggered signals piggybacked and transmitted periodically.

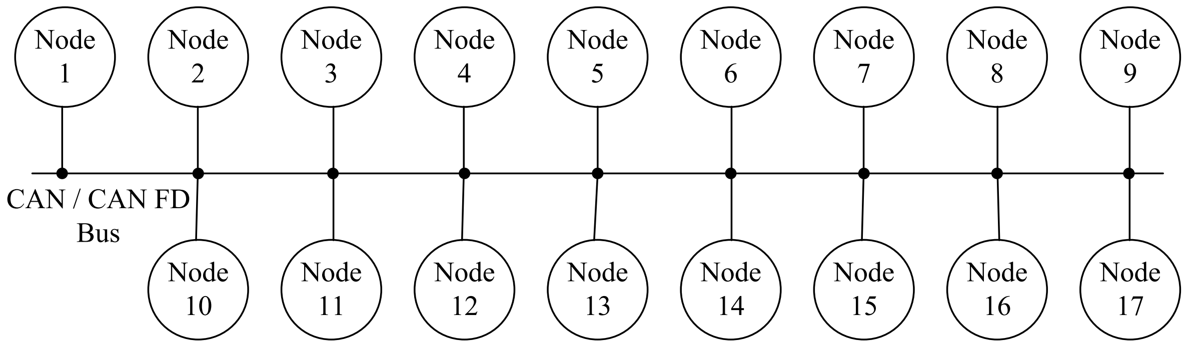

Figure 3: Simplified representation of the simulation model.

Messages from M2 to M6, and from M12 to M17 are time-triggered messages. In the table, the messages are placed in priority order from the highest (M1) to the lowest (M17). T indicates the message period, while D indicates the deadline in milliseconds [6].

Table 1: SAE benchmark based message set.

Message	Size (bytes)	T (ms)	D (ms)
M1	1	50.0	5.0
M2	2	5.0	5.0
M3	1	5.0	5.0
M4	2	5.0	5.0
M5	1	5.0	5.0
M6	2	5.0	5.0
M7	6	10.0	10.0
M8	1	10.0	10.0
M9	2	10.0	10.0
M10	3	10.0	10.0
M11	1	50.0	20.0
M12	4	100.0	100.0
M13	1	100.0	100.0
M14	1	100.0	100.0
M15	3	1000.0	1000.0
M16	1	1000.0	1000.0
M17	1	1000.0	1000.0

In the model development process, the SimEvents toolbox of Matlab Simulink software package has been used. First, a CAN model has been developed in order to investigate the performance of the standard CAN protocol with the SAE based benchmark messages. Then, a CAN FD model has been developed in order to compare the results and evaluate the performance improvements achieved with the CAN FD protocol. In the modelling process, 11 bit identifier size of standard CAN protocol has been used. Figure 3 shows the simplified representation of the simulation model developed for CAN and CAN FD systems. In the simulation model, one node is assigned for each message. In order to investigate the system performances, the models have been simulated at 1 Mbps, 500 kbps, 250 kbps, and 125 kbps transmission bit-rates. In CAN FD, 5 Mbps is used as the data-phase transmission bit-rate.

Performance analysis

In order to evaluate the performance results and compare the performance improvements achieved with the CAN FD protocol, both CAN and CAN FD models have been simulated at four different transmission speeds. The results for worst-case and average message delays have been analysed and visualised in graphs. As an important performance parameter, the bus utilisation results have also been analysed.

Worst-case message delay analysis

The worst-case message delay or response time analysis is important in real-time systems since it indicates the performance of the system to meet the deadline criteria. Figures from 4 to 7 show the worst-case message delay graphs for 1 Mbps, 500 kbps, 250 kbps, and 125 kbps bus transmission bit-rates. The graph “CAN max reference” represents the message delay values given in [6]. These values are used as a reference to evaluate the results obtained in simulations. The graph “CAN max” represents the worst-case delay values for the CAN model.

The graph "CAN FD max" represents the worst-case delay values for the CAN FD model. The delays are shown in the graphs in message priority order from the highest (M1) to the lowest (M17).

As can be seen from the figures, CAN worst-case delay values obtained with simulation show discrepancies from the CAN reference delay values. This is because the worst-case delay is very rare in simulations, whereas the reference values are obtained by analysis. The study in [12] presents the evaluation and comparison of results obtained by simulation and analysis. However, at lower transmission bit-rates, the simulation and reference results present close delay values since at lower transmission speeds it is more likely that the messages experience the worst-case delays. From the graphs, it can be seen that the CAN FD model provides considerable worst-case message delay performance improvement for the system. This improvement is achieved due to the higher transmission speed used in data-phase of the message frame.

In order to investigate the performance improvement in more detail, the worst-case delay ratio of each message in CAN to CAN FD is computed, and the message delay ratios are presented in Table 2 as average, minimum, and maximum. The delay ratios are investigated at four different transmission bit-rates. At 1 Mbps bit-rate, the average ratio is 1.78, that is, the worst-case message delay with CAN FD is 1.78 times smaller on average than it is with CAN. The average ratio at 500 kbps is 2, which means CAN FD has half worst-case message delay of CAN on average. The ratio is 2.16 and 3.28 at 250 kbps and 125 kbps bit rates, respectively. These ratios show the average worst-case message delay improvement achieved with the CAN FD protocol.

The table also gives the minimum and maximum worst-case message delay ratios. The minimum ratio ranges from 1.66 to 2.5, while the maximum ratio ranges from 1.98 to 6.00, which means CAN FD transmits messages up to 6 times smaller worst-case message delays with the SAE benchmark based message set.

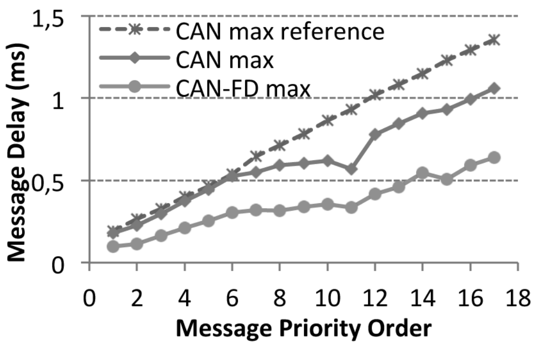

Figure 4: Worst-case delays at 1 Mbps

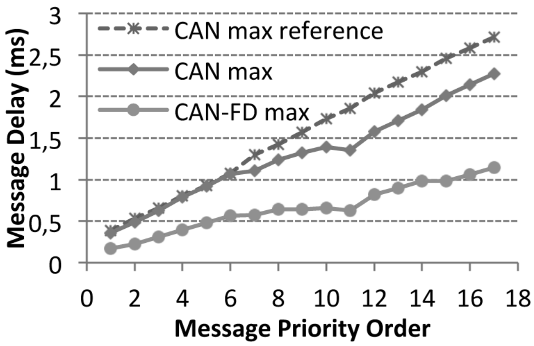

Figure 5: Worst-case delays at 500 kbps

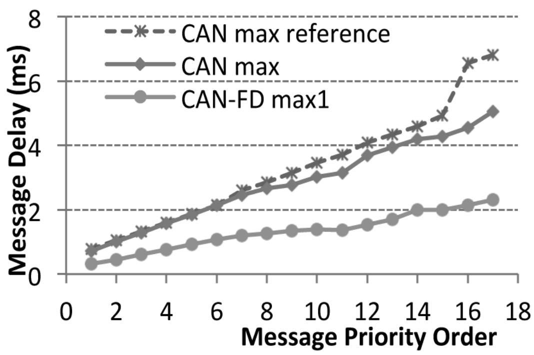

Figure 6: Worst-case delays at 250 kbps

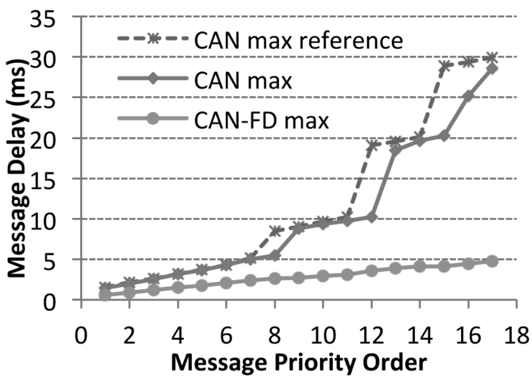

Figure 7: Worst-case delays at 125 kbps

Table 2: Worst-case message delay ratios

CAN /CAN FD	1 Mbps	500 Mbps	250 Mbps	125 Mbps
Average	1.78	2.00	2.16	3.28
Minimum	1.66	1.88	1.99	2.05
Maximum	1.98	2.15	2.39	6.00

Average message delay analysis

Figures from 8 to 11 show the average message delay values for CAN and CAN FD models at four different bus bit-rates. As can be seen from the figures, the dominant factor affecting the CAN average delay is the message transmission time at 1 Mbps ant 500 kbps bus bit-rates. At 250 kbps bus bit-rate, the effect of arbitration delay becomes noticeable in CAN as the lower priority messages experience rising delays due to the increased bus utilisation. At 125 kbps bus bit-rate, the effect of the arbitration delay becomes very obvious in CAN as the lower priority messages experience very high message delays. However, the average message delays in CAN FD are mostly affected by the transmission time and only little effect of the arbitration delay can be observed at almost all bus bit-rates. The highest message transmission time in CAN is observed with message 7 as it has the largest payload size in the message set, and the effect of payload difference is noticeable. However, the payload size difference is not very noticeable in CAN FD. This shows the performance improvement achieved with CAN FD due to the application of higher transmission bit-rate in the data-phase of the message frame. This application provides enough bandwidth for message transmissions with lower delays even at low arbitration-phase bit rates. As can be seen from the figures, even at 125 kbps, CAN FD average message delays are comparatively much smaller than CAN delays, which become extensively high especially with lower priority messages.

In order to analyse performance improvements in more detail, the ratios also for the average message delays have been computed and listed in Table 3 as average, minimum and maximum.

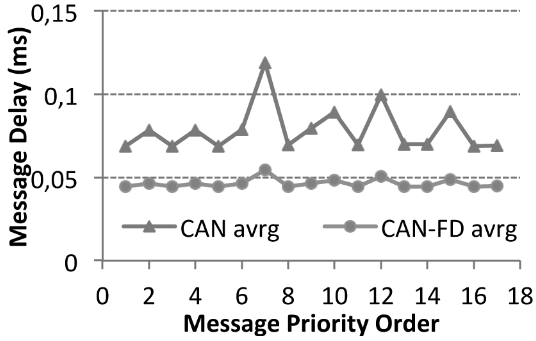

Figure 8: Average delays at 1Mbps

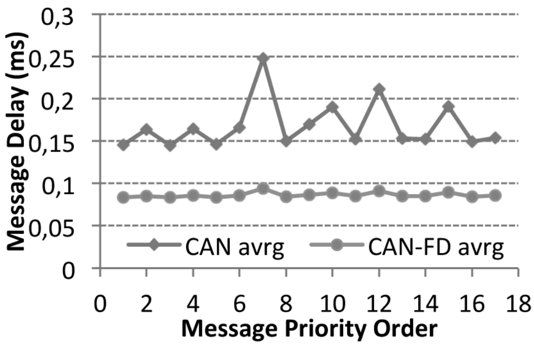

Figure 9: Average delays at 500 kbps

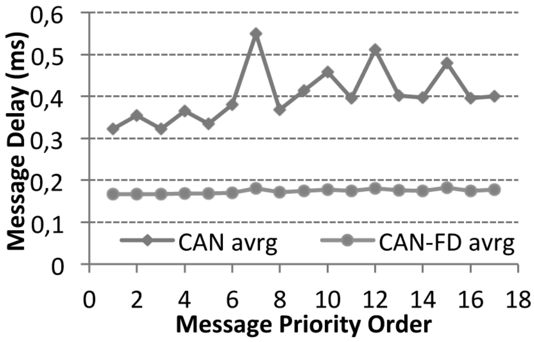

Figure 10: Average delays at 250 kbps

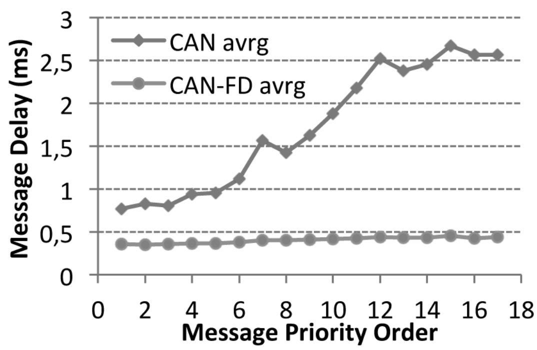

Figure 11: Average delays at 125 kbps

Table 3: Average message delay ratios

CAN /CAN FD	1 Mbps	500 Mbps	250 Mbps	125 Mbps
Average	1.67	1.94	2.31	4.16
Minimum	1.53	1.73	1.92	2.16
Maximum	2.17	2.62	3.03	6.00

At 1 Mbps transmission bit-rate, mean message delay is 1.67 times smaller on average in CAN FD than it is in CAN. The average ratio for bit-rates from 1 Mbps to 125 kbps ranges from 1.67 to 4.16, which means up to 4.16 times smaller average message delays are achieved on average with CAN FD. The minimum ratio ranges from 1.53 to 2.16, and the maximum ratio ranges from 2.17 to 6.00, which means, as in worst-case delay ratios, up to 6 times smaller average message delays are achieved with CAN FD.

Bus utilisation analysis

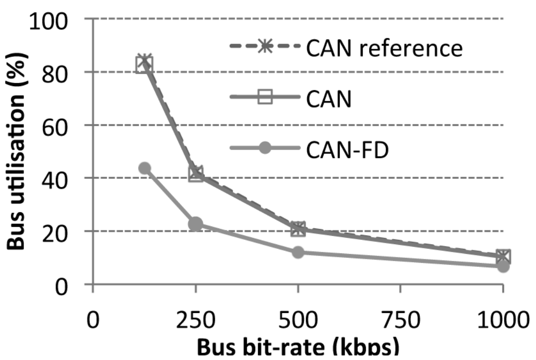

Figure 12: Bus utilisation.

Figure 12 shows the bus utilisation values for CAN reference, CAN, and CAN FD. From the figure, it can be seen that the "CAN reference" and "CAN" graphs show almost the same characteristics. The "CAN FD" graph shows that the same message set is transmitted with less bus utilisation with CAN FD.

Bus utilisation affects the system performance. As the utilisation becomes higher, the delay caused by the arbitration mechanism also gets higher. At high bus utilisation values, especially lower priority messages experience extensive bus access delays as they have to wait for the higher priority messages to be transmitted first. This effect can be seen clearly at 125 kbps bus bit-rate, where the bus utilisation is over 80%. However, the bus utilisation with CAN FD at the same bit-rate is just over 40%, which is almost half that of CAN. The simulated bus utilisation values at bus bit-rates from 1 Mbps to 125 kbps ranges from 10.3 to 82.5 for CAN, and from 6.7 to 43.8 for CAN FD, respectively. The results show that compared to CAN, CAN FD provides faster message transmission with less bus utilisation.

Conclusion

As the amount of ECUs and message traffic in automotive and industrial applications increase, the more need arises for higher bandwidth and faster message transmission. The CAN FD protocol provides a solution for bit-rate limitation in CAN using higher transmission bit-rates in data-phase of the message frame.

In this study, modelling and simulation of the CAN and CAN FD systems with the SAE benchmark based message set have been realised in order to investigate the performance improvements achieved by the CAN FD protocol. The performance analysis have been realised for worst-case message delays, average message delays, and bus utilisations. The worst-case message delay analysis has shown that with the CAN FD model based on the SAE benchmark message set, from 1.78 to 3.28 times smaller worst-case message delays on average can be achieved compared to the CAN model. Similarly, from 1.67 to 4.16 times smaller mean message delays on average has been observed with CAN FD. In bus utilisation results, almost half the bus utilisation values have been observed with CAN FD.

The simulation results have revealed that the CAN FD protocol provides considerable performance improvements in message transmission speed and bus utilisation compared to CAN.

Acknowledgement: This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK).

Mahmut Tenruh
Mugla S. K. University
Department of Electrical & Electronics Engineering
TR-48000 Mugla
tmahmut@mu.edu.tr

Panagiotis Oikonomidis
University of Brighton
Vetronics Research Centre (VRC)
Lewes Road
GB-BN2 4GJ Brighton
p.oikonomidis@vetronics.org
www.vetronics.org

Periklis Charchalakis
University of Brighton
Vetronics Research Centre (VRC)
Lewes Road
GB-BN2 4GJ Brighton
p.charchalakis@vetronics.org
www.vetronics.org

Elias Stipidis
University of Brighton
Vetronics Research Centre (VRC)
Lewes Road
GB-BN2 4GJ Brighton
e.stipidis@vetronics.org
www.vetronics.org

References

[1] CAN Specification, Version 2.0, Robert Bosch GmbH, 1991.

[2] H. Bauer and G. Kopetz, “The time-triggered architecture,” Proc. IEEE, vol. 91, no. 1, pp. 112–126, 2003.

[3] FlexRay Communications System Protocol Specification, Version 3.0.1, the FlexRay Consortium, 2010.

[4] F. Hartwich, “CAN with flexible data-rate,” in 13th Int. CAN Conf., 2012, pp. 14–1–14–9.

[5] CAN with Flexible Data-Rate, specification version 1.0, Robert Bosch GmbH, 2012.

[6] K. Tindell and A. Burns, “Guaranteeing Message Latencies on Control Area Network (CAN),” in 1st Int. CAN Conf., 1994, pp. 1–11.

[7] K. W. Tindell, A. Burns, and A. J. Wellings, “Calculating Controller Area Network (CAN) message response times,” Control Eng. Pract., vol. 3, no. 8, pp. 1163–1169, 1995.

[8] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller area network (CAN) schedulability analysis: Refuted, revisited and revised,” Real-Time Syst., vol. 35, no. 3, pp. 239–272, 2007.

[9] A. Mutter, “CAN FD and the CRC issue,” CAN Newsletter, no. 1, pp. 4–10, 2015.

[10] U. D. Bordoloi and S. Samii, “The Frame Packing Problem for CAN-FD,” in 2014 IEEE Real-Time Systems Symposium, 2014, pp. 284–289.

[11] G. Cena, I. C. Bertolotti, T. Hu, and A. Valenzano, “Fixed-length payload encoding for low-jitter controller area network communication,” IEEE Trans. Ind. Inf., vol. 9, no. 4, pp. 2155–2164, 2013.

[12] N. Navet, “CAN in Automotive Applications: a Look Forward,” in 13th Int. CAN Conf., 2012.