Comparative Evaluation of Various Generations of Controller Area Network Based on Timing Analysis

*Aldin Berisa, †Adis Panjevic, †Imran Kovac, ‡Hans Lyngbäck,
*Mohammad Ashjaei, *Masoud Daneshtalab, *Mikael Sjödin,*Saad Mubeen
*Mälardalen University, Västerås, Sweden; firstname.lastname@mdu.se
†Arcticus Systems, Järfälla, Sweden; firstname.lastname@arcticus-systems.com
‡HIAB, Hudiksvall, Sweden; firstname.lastname@hiab.com

***Abstract*—This paper performs a comparative evaluation of various generations of Controller Area Network (CAN), including the classical CAN, CAN Flexible Data-Rate (FD), and CAN Extra Long (XL). We utilize response-time analysis for the evaluation. In this regard, we identify that the state of the art lacks the response-time analysis for CAN XL. Hence, we discuss the worst-case transmission times calculations for CAN XL frames and incorporate them to the existing analysis for CAN to support response-time analysis of CAN XL frames. Using the extended analysis, we perform a comparative evaluation of the three generations of CAN by analyzing an automotive industrial use case. In crux, we show that using CAN FD is more advantageous than the classical CAN and CAN XL when using frames with payloads of up to 8 bytes, despite the fact that CAN XL supports higher bit rates. For frames with 12-64 bytes payloads, CAN FD performs better than CAN XL when running at the same bit rate, but CAN XL performs better when running at a higher bit rate. Additionally, we discovered that CAN XL performs better than the classical CAN and CAN FD when the frame payload is over 64 bytes, even if it runs at the same or higher bit rates than CAN FD.**

***Index Terms*—Controller Area Network, CAN FD, CAN XL, automotive.**

I. Introduction

In the automotive domain, various in-vehicle networks interconnect many onboard Electronic Control Units (ECUs) [1]. Controller Area Network (CAN) [2], referred to as the classical CAN in this paper, is the most widely used in-vehicle real-time network to date. The classical CAN is limited by the maximum network speed of 1 Mbit/s and maximum size of data payload of 8 bytes in a single frame. Due to these limitations, classical CAN is unable to support the bandwidth requirements imposed by advanced communication-demanding functionalities in modern vehicles. The second generation of the CAN standard, known as the CAN FD (Flexible Data-rate) [3], attempts to address these limitations to some extent. CAN FD offers higher network speeds up to 8 Mbit/s and includes a frame format that can hold a payload of up to 64 bytes. These enhancements allow increased data throughput compared to the classical CAN.

The enhancements to CAN FD are not sufficient in handling the high-bandwidth demands of the upcoming self-driving vehicles. To address this challenge, the third generation of CAN, regarded as the CAN Extra Long (XL) [4], is recently introduced to support network speeds of 10 Mbit/s and higher. Furthermore, the frame format has been redesigned to carry payloads of up to 2048 bytes. In addition, CAN XL also offers improved security and better coexistence with other in-vehicle networks, such as real-time Ethernet and previous generations of CAN. Therefore, the current status is that three CAN generations exist with different characteristics. In order to identify the advantages of each CAN generation and its suitability for industrial use cases, a comparative evaluation of these generations of CAN is essential. Such an evaluation is currently lacking in the state of the art. To the best of our knowledge, this is the first paper that performs a comparative evaluation of the three generations of CAN based on their timing analysis. While performing the evaluation, we identified that the state of the art lacks the response-time analysis that supports all generations of CAN. This is the first paper that fills this gap in the state of the art by extending the existing response-time analysis for CAN to support CAN XL frames. The main contributions in this paper are as follows:

- We extended the existing response-time analysis for CAN [5], [6] to support the analysis of CAN XL frames. The extended analysis is backwards compatible with the previous generations of CAN, including CAN FD and classical CAN.
- Using the extended analysis, we perform a comparative evaluation of the three generations of CAN with respect to various parameters including the payload size, transmission times, response times, and different configurations of CAN FD and CAN XL. The evaluation is performed based on an automotive industrial use case.

II. Related work

There are several works that have performed a comparative evaluation of classical CAN and CAN FD, but not CAN XL. For instance, Kim et al. [7] and Xie et al. [8] compared classical CAN and CAN FD using both real-world and simulated scenarios and found out that using CAN FD instead of classical CAN significantly reduces the frames' response times and improves the network's bandwidth utilization.

When it comes to analyzing CAN frames, most of the existing research has focused on response-time analysis (RTA) of classical CAN frames in different configurations. Tindell et al. [5] developed the RTA for CAN that was later revised by Davis et al. [6]. Yomsi et al. [9] further extended the RTA to support CAN frames with offsets, and Davis et al. [10] extended it to support CAN controllers with FIFO queues. Mubeen et al. [11] extended the RTA for CAN to include mixed messages that can be both periodic and sporadic. They have also extended their RTA for mixed messages to include offsets, as described in [12]. Furthermore, the RTA for mixed messages was extended to support FIFO queues [13] and practical limitations in the CAN controllers [14].

Calculating the response times of frames in CAN FD has not been extensively studied. A challenge in frame packaging arises when dealing with signals that have varying properties. In addressing this issue, Bordoloi et al. [15] proposed an algorithm that prioritizes frames based on signal deadlines. This approach greatly improves the bus's bandwidth utilization while ensuring that all signal constraints are met. Their work also includes equations for determining the best-case and worst-case transmission times of CAN FD frames.

Ikumapay et al. [16] presented a worst-case execution time calculation for CAN XL frames, while their intention was to support authentication for CAN. Moreover, their analysis does not include different transmission bit rates for arbitration and data phases used in CAN XL and CAN FD. Although there exist several works that do comparative evaluation of classical CAN and CAN FD as done in [7], [8], there is no published work that includes CAN XL in the comparative evaluation. One reason could be that CAN XL is recently introduced. In this paper, we aim to address this gap by conducting a comparative evaluation of all generations of CAN based on their timing analysis. To achieve this, we extended the existing RTA for CAN to support response-time calculations of CAN XL and CAN FD frames besides classical CAN.

III. Various Generations of CAN Protocols

A. Controller Area Network (CAN)

The CAN protocol [2] is an ISO standard and is one of the most widely used onboard real-time network protocols in the automotive domain. Basically, CAN is an asynchronous multi-master serial communication bus that uses Carrier Sense Multiple Access/Collision Resolution (CSMA/CR) as its access control method. In the CAN network, "0" is the dominant bit and "1" is the recessive bit. The maximum bus speed supported by the classical CAN protocol is 1 Mbit/s. A CAN frame can support a maximum data payload of 8 bytes. The structure of CAN data frame is depicted in Fig. 1. Various fields in the CAN data frame are described as follows.

- *Start Of Frame (SOF) field*: Initial bit of every CAN data frame which is transmitted as a single dominant bit. It serves the purpose of synchronizing ECUs on the CAN bus.
- *Arbitration field*: This field contains the identifier and Remote Transmission Request (RTR) bit. Each CAN data frame is assigned a unique identifier, which can be either 11 bits standard or 29 bits extended format. The RTR bit is utilized to indicate whether the data frame is a request for remote transmission of data.
- *Control field*: This field consists of three parts: (i) Identifier Extension, denoted by IDE bit (shows whether the frame format is standard or extended), (ii) Reserved bit (r_0) for future extensions of the protocol, (iii) Data Length Code, denoted by DLC (size of data payload in number of bytes).
- *Data field*: The data field holds the frame's payload and can range from 0 to 8 bytes.
- *Cyclic Redundant Check (CRC)*: The CRC field ensures the integrity of the data frame. This field contains a CRC delimiter bit that allows the ECU to perform a CRC calculation and verify the integrity of the frame.
- *Acknowledge (ACK)*: This field comprises ACK slot and delimiter bits. To acknowledge the successful reception of a frame, the receiver transmits a dominant bit in the ACK slot.
- *End of Frame (EOF)*: The EOF field consists of seven recessive bits that indicate the end of a CAN data frame.
- *Intermission field (Int)*: The intermission field (Int) is utilized to separate consecutive CAN data frames.

When a problem is detected in the CAN bus by an ECU, it transmits an error frame that contains a sequence of six consecutive dominant bits. Other ECUs that encounter the same problem will also transmit an error frame with a sequence of six consecutive recessive bits. These sequences indicate bus faults and should be avoided during regular data frame transmissions. To address this issue, dynamic bit stuffing is introduced. This approach requires transmitting a bit of the opposite polarity after every sequence of five identical bits. Dynamic bit stuffing applies to all bits until the 15-bit CRC.

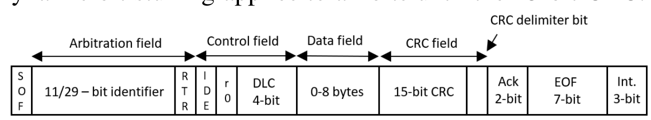

Fig. 1: Data frame format in CAN classic.

B. CAN Flexible Datarate (CAN FD)

CAN FD [3] is an ISO standard that is an enhanced version of the classical CAN protocol. By increasing the payload size up to 64 bytes and providing higher data rates up to 8 Mbit/s, CAN FD allows for higher data throughput compared to classical CAN. The structure of CAN FD data frame, shown in Fig. 2, is built upon the classical CAN data frame by modifying/expanding the functionality of the following fields.

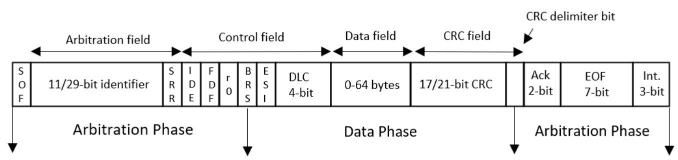

Fig. 2: Data frame format in CAN FD.

- *Arbitration field*: In the extended frame format, the RTR bit is replaced by the Substituted Remote Request (SRR) bit. The SRR bit is sent as a recessive bit and helps distinguish between frames with the same first 11 bits of the identifier but use different frame formats. If two frames have the same identifier, the one with the base format is prioritized since its RTR bit is transmitted as dominant.
- *Control field*: The control field has been enhanced with three new additions: Flexible Data Rate Format (FDF), Error State Indicator (ESI), and Bit Rate Switch (BRS) fields. The FDF field distinguishes between classic CAN frames and CAN FD frames, while the ESI indicates whether the ECU is transmitting in error passive or error active state. During the arbitration phase, the transmission occurs at a nominal bit rate of up to 1 Mbit/s until the BRS field is reached. If the BRS is set to 1, the data phase utilizes a higher bit rate than the nominal bit rate until the CRC delimiter field is not transmitted. However, if the BRS is set to 0, the data phase is transmitted using the nominal bit rate.
- *Data field*: CAN FD data frame supports all the payloads that are supported by the data frame of classic CAN (0-8 bytes). In addition, CAN FD data frame supports larger payloads of 12, 16, 24, 32, 48, and 64 bytes.
- *CRC field*: In CAN FD, the CRC has 17 bits for payloads of up to 16 bytes and 21 bits for payloads larger than 16

bytes. If the higher bit rate is used during the data phase, the arbitration phase begins again when transmitting the CRC delimiter bit. The remaining fields are then transmitted using the nominal bit rate.

In CAN FD, the bit stuffing mechanism has been modified to include fixed bit stuffing in the CRC field. The dynamic bit stuffing is still employed in the fields leading up to the CRC. With fixed bit stuffing, the stuff bits are inserted at predetermined positions by adding a fixed stuff bit of opposite polarity after every tenth bit.

C. CAN Extra Long (CAN XL)

A new generation of CAN, called the CAN Extra Long (XL), is recently developed that modifies both the physical layer [17] and data-link layer [4] of the classic CAN protocol. CAN XL supports data rates higher than 10 Mbit/s and data payload of up to 2048 bytes. In addition, CAN XL includes several new features, such as tunneling frames from other protocols like Ethernet, CAN FD, and classic CAN. Receiving ECUs can filter out frames at the hardware level, reducing the load on the ECU. Fragmentation and aggregation of frames are also supported in CAN XL.

Similar to CAN FD data frame, CAN XL data frames can be transmitted in two phases. During the arbitration phase, the frame is sent at a nominal bit rate of up to 1 Mbit/s. However, during the data phase, CAN XL data bit rate is used and can reach up to 10 Mbit/s or more depending on the physical implementation of the CAN XL controllers. The structure of a CAN XL data frame is depicted in Fig. 3. It further expands and adds functionality in the following fields.

- *Arbitration field*: Two new fields are included in the arbitration field of CAN XL: the CAN XL Format Indicator (XFL) and the Arbitration to Data Sequence field (ADS). The data frames of previous CAN generations can be transmitted by ECUs in the same network, which is why the combination of IDE, FDF, and XFL fields is used to indicate whether classic CAN or CAN FD (standard or extended format) is being transmitted. Prioritization in CAN XL is determined by the 11 + 18-bit identifier, with the first 11 bits used for frames with an 11-bit identifier and the full 29 bits used for extended frame formats of classic CAN or CAN FD.
- *Arbitration to data sequence (ADS)*: This field works similar to the BRS in CAN FD. The ADS field consists of the ADH, DH1, DH2 and DL1 bits. If bit rate switching is used, the ADH bit is the last bit sent with the arbitration bit rate, and the data phase starts with the transmission of the DH1 bit.
- *Control field*: The control field has been expanded by including various fields. The Service Data Type (SDT) indicates the next OSI layer protocol used, allowing for tunneling of multiple classic CAN, CAN FD, or Ethernet frames. The DLC field has been extended to 11 bits to support values up to 2048, representing the maximum payload size in bytes. The Stuff Bit Count (SBC) provides information on the number of dynamic stuff bits in the arbitration field. The preface CRC (PCRC) sequence checks the integrity of the Arbitration Field, SDT, DLC, and SBC. The Virtual CAN Network ID (VCID) indicates the virtual CAN network to which the frame belongs. When transmiting CAN XL data frames, addressing is separated from prioritization and done by the 31-bit Addressing Field (AF).
- *Data field*: It contains data between 1 and 2048 bytes, with each byte being numbered from 0 to the value of the DLC.
- *CRC field*: The CAN XL protocol includes a CRC field to check the integrity of the arbitration, control, and data fields using the frame CRC (FCRC) sequence. The Format Check Pattern (FCP) sequence in the CRC field is transmitted prior to switching the bit rate back from the data bit rate to the nominal bit rate. It provides a synchronizing edge before the phase transition.
- *ACK field*: When utilizing the CAN XL Data bit rate in the data phase, the bit rate reverts back to the nominal bit rate during the transmission of Data to Arbitration Sequence (DAS) in the ACK field.

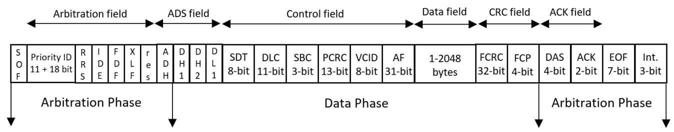

Fig. 3: Data frame format in CAN XL.

The CAN XL protocol has made some modifications to the bit stuffing mechanism. Fixed bit stuffing is used in the control, data, and CRC fields, while dynamic bit stuffing is employed in the SOF and arbitration fields.

IV. System model

The system consists of a set of ECUs that are interconnected using a CAN bus. During the bus arbitration, each ECU transmits its highest priority frame since all ECUs implement priority-based queues in the CAN interfaces. Each CAN frame m is uniquely identified with identifier ID_m. The priority of a CAN frame, denoted by P_m, is assumed to be equal to its identifier. The priority of frame m is considered higher than the priority of frame n if $P_m < P_n$. Let the set $hp(m)$ contain the frames with priorities higher than the priority of frame m. Whereas the set $lp(m)$ contain the frames with priorities lower that the priority of m.

Each CAN frame m has the worst-case transmission time C_m, a transmission period T_m and queuing jitter J_m that is inherited from the task that queues the frame for transmission. The size of the data payload in m is denoted by s_m. The range of values in s_m varies depending upon the generation of the CAN protocol being used. That is, the range of values in s_m can be 0-8 bytes in a classic CAN data frame, 0-64 bytes in a CAN FD data frame, and 1-2048 bytes in a CAN XL data frame.

The time to transmit a single bit on a CAN bus is represented by τ_{bit} and depends on the speed of the network. If the newer generations of the CAN protocol are utilized, including CAN FD and CAN XL, the frame transmission occurs in two stages - arbitration and data. The arbitration phase has a slower transmission rate, denoted by τ_{arb}, while the data phase has a faster transmission rate, τ_{data}. The two-phase transmission is used to maintain backward compatibility with classic CAN.

The worst-case response time of a CAN frame, denoted by R_m, is defined as the longest time between the queuing of the frame for transmission and the delivery of the frame to the destination ECU. The amount of time m has to wait for the transmission of a lower priority frame is defined as the blocking time, denoted by B_m. Each frame has a deadline

D_m. In this model, the deadlines are set to be equal to the frame period, also known as implicit deadlines.

V. Supporting Response Time Analysis of CAN XL

This section presents the response-time analysis for different generations of CAN. However, we found that the current state-of-the-art response-time analysis (RTA) for CAN [6], [11], [18] is not able to support CAN XL and CAN FD frames. This is because it doesn't take into account the varying transmission rates that CAN FD and CAN XL supports during arbitration and data-transmission phases. In this section, we address this issue and make the existing RTA applicable to both CAN XL and CAN FD frames.

A. Quickly Revisiting the Existing RTA for CAN

According to the existing RTA, the response time (R_m) of a CAN frame m is equal to the sum of its worst-case transmission time denoted by C_m, queuing delay denoted by ω_m and queuing jitter denoted by J_m, as shown by Eq. (1).

$$R_m = J_m + \omega_m + C_m \tag{1}$$

1) Calculations for Queuing Jitter: The queuing jitter, J_m, is inherited from the task that queues the frame for transmission. J_m represents the maximum variation in time between the release of the sending task and queuing of the frame in the output queue of the CAN interface within an ECU. J_m is calculated by considering the difference between the worst- and best-case response times of the sending task [11].

2) Calculations for Queuing Delay: The queuing delay in Eq. (1) represents the maximum amount of time a frame remains in the output queue of the CAN interface within an ECU before it is transmitted to the CAN bus. The queuing delay consists of two components: (i) blocking time due to the set of lower priority frames, denoted by B_m, and (ii) interference from higher priority frames shown by the second term on the right-hand side of Eq. (2). To calculate the queueing delay, we need to use fix-point iteration as both sides of the equation in Eq. (2) include ω_m.

$$\omega_m = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{\omega_m + J_m + \tau_{bit}}{T_k} \right\rceil C_k \tag{2}$$

Note that the signal propagation on the CAN bus gives uncertainty in observation of its arrival at remote nodes compared to the nodes that are closest to the signal's source. This results in jitter in the signal propagation that is bounded by τ_{bit} (i.e., time to transmit a single bit on CAN) in Eq. (2). The blocking time refers to the largest transmission time of a message among all lower-priority messages, shown in Eq. (3).

$$B_m = \max_{\forall k \in lp(m)} (C_k) \tag{3}$$

3) Calculations for Worst-case Transmission Time (WCTT): To calculate the WCTT of a CAN frame, we need to derive the number of bits transmitted in the frame. This includes the payload denoted by the number of bytes s_m, the size of other fields in bits, and the worst-case number of stuff bits inserted in the frame. The total number of bits in the frame are multiplied by the data transmission rate (τ_{bit}) to get the WCTT of the frame. The WCTT of a classical CAN frame with standard and extended frame identifier formats can be calculated using Eq. (4) and Eq. (5) respectively.

$$C_m = (55 + 10s_m)\tau_{bit} \tag{4}$$

$$C_m = (80 + 10s_m)\tau_{bit} \tag{5}$$

Similarly, the WCTT of a CAN FD frame with standard and extended frame identifier formats can be calculated using Eq. (6) and Eq. (7), respectively. Note that the WCTT calculations for CAN FD frames were developed in [15]. These calculations involve determining the number of bits transmitted during the arbitration phase at nominal bit rate of CAN FD τ_{arb} and data phase at data bit rate of CAN FD τ_{data}. If bit rate switching is not used, then τ_{data} is equal to τ_{arb}, and the entire frame is transmitted at the nominal bit rate. The number of bits during the arbitration phase depends on the frame format, while the number of bits in the data phase depends on the payload size s_m. The CRC also varies depending on the payload size; if the payload is less than 16 bytes, a 17-bit CRC is used, otherwise a 21-bit CRC is used.

$$C_m = 32\tau_{arb} + \left(28 + 5\left\lceil \frac{s_m - 16}{64} \right\rceil + 10s_m\right)\tau_{data} \tag{6}$$

$$C_m = 54\tau_{arb} + \left(28 + 5\left\lceil \frac{s_m - 16}{64} \right\rceil + 10s_m\right)\tau_{data} \tag{7}$$

B. WCTT Calculations for CAN XL Frames

In this section, we will discuss the WCTT calculations for CAN XL frames that were developed during the same timeline as the one presented by Ikumapay et al. [16]. To determine the WCTT of a CAN XL frame, we must calculate the number of bits transmitted during both the arbitration and data phases of CAN XL, similar to how it is done in CAN FD. In Fig. 3, we notice that the arbitration phase includes a 15-bit arbitration field, a single-bit SOF field, two control bits from the ADS, 6 bits from the ACK field, 7 bits in the EOF field and 3 bit of interframe gap. The dynamic bit stuffing is used in the SOF and arbitration fields, which can add up to 3 dynamic stuff bits in the worst-case scenario. This is because an extra bit of opposite polarity is added after every tenth bit in the case of dynamic bit stuffing. As the total number of bits in the above mentioned fields is 34 bits, the total number of extra bits required for dynamic bit-stuffing is 3.

Additionally, transitioning from the arbitration phase to the data phase can cause a phase error due to differences in clock speeds between the transmitter and the receiver. To account for this, the receiver should tolerate up to six consecutive recessive bits starting from the DH1 recessive bit and a missing DH2 bit in the ADS [4]. Nonetheless, a total of four extra recessive bits can be transmitted during the data phase, aside from the DH1 and DH2 bits.

If we take into account the possibility of 3 stuffed bits, the total number of bits transmitted during the arbitration phase at nominal bit rate τ_{arb} can be up to 37 bits (34 + 3 bits).

When transmitting using the data transmission bit rate, denoted as τ_{data}, 119 bits (a control field of 79 bits + a CRC field of 36 bits + additional 4 recessive bits during phase error), and s_m number of bytes ($8s_m$ bits) in the data field are transmitted. Fixed bit stuffing is used from the DL1 bit up to the end of the FCRC field, in which after every tenth bit, a bit of the opposite polarity is added. In summary, a total of 109 bits and the payload undergo fixed bit stuffing. The total number of stuff-bits can be represented by Eq. (8).

$$\text{Nr. of Stuff Bits in CAN XL Frame} = \left\lfloor \frac{109 + 8s_m}{10} \right\rfloor \quad (8)$$

The WCTT of a CAN XL frame can be calculated by Eq. (9).

$$C_m = 37\tau_{arb} + \left(119 + 8s_m + \left\lfloor \frac{109 + 8s_m}{10} \right\rfloor\right)\tau_{data} \quad (9)$$

C. Supporting RTA for the Next Generations of CAN

In order to support the worst-case response-time calculations of CAN XL frames using the existing RTA for CAN, the calculations for the queuing delay in Eq. (2) need to be adapted to account for the jitter in the signal propagation (τ_{bit}). This is because CAN XL uses different transmission bit rates for arbitration (τ_{arb}) and data phases (τ_{data}). In the case of CAN XL, we use the maximum among the two transmission bit rates to account for the jitter in the signal propagation. As τ_{arb} can be either equal to or greater than τ_{data} when the bit-rate switching is enabled in CAN XL, we replace τ_{bit} with τ_{arb} in the calculations for the queuing delay in Eq. (2). Note that τ_{data} is equal to τ_{arb} when the bit-rate switching is not enabled in CAN XL. The similar reasoning applies to CAN FD as it also uses different transmission bit rates for arbitration (τ_{arb}) and data phases (τ_{data}). The calculations for the queuing delay in the three generations of CAN are depicted in Eq. (10). Note that the remaining equations for RTA including Eq. (1) and Eq. (3) remain the same for all generations of CAN.

$$\omega_m = \begin{cases} B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{\omega_m + J_m + \tau_{bit}}{T_k} \right\rceil C_k, & \text{If CAN} \\ B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{\omega_m + J_m + \tau_{arb}}{T_k} \right\rceil C_k, & \text{If CAN FD} \\ B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{\omega_m + J_m + \tau_{arb}}{T_k} \right\rceil C_k, & \text{If CAN XL} \end{cases} \quad (10)$$

VI. Evaluation: Automotive Industrial Use Case

In this section, we presents an evaluation of the extended RTA on various generations of the CAN protocol in an automotive industry use case, which can also be considered a common scenario. Further, we will perform a comparative evaluation of the transmission times and response times of the frames in the use case across all three generations of CAN.

A. Use-case Description and Scenarios

The industrial use case represents the lever system that is used to control forestry and recycling vehicles. The architecture of the lever system is illustrated in Fig. 4. There are 7 ECUs that communicate through 47 periodic frames. The properties of the frames are shown in Tables I & II.

The evaluation on the industrial use case was done for all generations of CAN in two different scenarios. The first scenario represents a realistic case where classical CAN and the arbitration phase of CAN FD and CAN XL run at 500 Kbit/s. Classical CAN is typically implemented at 500/250 Kbit/s in many industrial applications. The existing physical

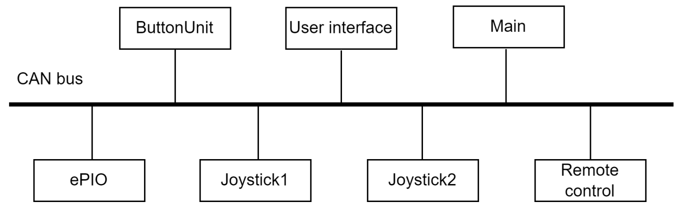

Fig. 4: The architecture of the automotive industrial use case.

implementations of CAN FD and CAN XL support up to 8 Mbit/s and 20 Mbit/s, respectively, during the data phase. In addition, CAN XL is also evaluated at 8 Mbit/s to make it comparable to CAN FD. The second scenario involves raising the bit rate of classical CAN and arbitration phase to the maximum theoretical limit of 1 Mbit/s. The data phase for CAN FD and XL remains unchanged from the first scenario.

B. Evaluation Results: Response Times with 8 Bytes Payload

The evaluation results from the first and second scenarios are depicted in Fig. 5 and Fig. 6, respectively. After analyzing the graphs of both scenarios, it is apparent that the response times of frames in the second scenario are lower than those in the first scenario. This was expected since the second scenario utilized the theoretical upper limit of bandwidth that CAN protocols can support. Notably, CAN FD had the lowest response times for all frames in both scenarios, while classical CAN had the highest response times. This is due to classical CAN having a lower transmission bit rate compared to CAN FD and CAN XL. Despite CAN XL running at 20 Mbit/s, CAN FD still outperformed it because of the higher overhead in a CAN XL frame compared to a CAN FD frame. Let's consider an example scenario with an 8 bytes payload. With a classical CAN frame, 135 bits are transmitted, while a CAN FD frame transmits 140 bits, and a CAN XL frame transmits 234 bits for the same payload. This means that almost 100 more overhead bits are needed to transmit a CAN XL frame with the same payload. These results indicate that if the maximum payload size is 8 bytes then CAN FD can be preferred over classical CAN or even CAN XL.

C. Evaluation Results: Payload Sizes vs Transmission Times

During our evaluation of the industrial use case, we discovered that the payload size could affect the performance of different CAN generations. This section evaluates the impact of various payload sizes on the transmission times and response times of the frames across all CAN generations, with and without bit rate switching. Multiple frames will be transmitted to achieve the required payload size if a particular generation doesn't support a certain payload size. For example, two frames will be transmitted to support 12 bytes of payload in the case of classical CAN as the maximum payload in a classical CAN frame is 8 bytes.

The transmission time for different payload sizes with bit rate switching turned off is shown in the right column in Fig. 7. Exact transmission times for frames can be found in Table IV. In this scenario, classical CAN, CAN FD and XL transmit frames with a bit rate of 1 Mbit/s during both arbitration and data phases. By examining the graphs, we can see that classical CAN performs the best until the payload of 8 bytes, which is expected due to the lower overhead resulting

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
T (ms)	50	100	500	50	20	20	20	20	50	100	500	500	100	100	500	500	500	500	500	100	20	500	500	500
DLC	7	8	3	1	2	2	8	8	8	4	5	6	8	8	8	8	8	8	8	8	4	8	8	8

TABLE I: CAN frames ID 0-23 used in the industrial use case. Periods (T) and their data length code (DLC)

ID	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
T (ms)	100	100	500	100	500	500	500	100	500	500	100	100	500	20	100	50	50	40	50	100	200	500	500
DLC	8	8	8	6	3	8	8	8	8	8	8	8	8	8	1	3	8	6	2	8	5	7	8

TABLE II: CAN frames ID 24-46 used in the industrial use case. Periods (T) and their data length code (DLC)

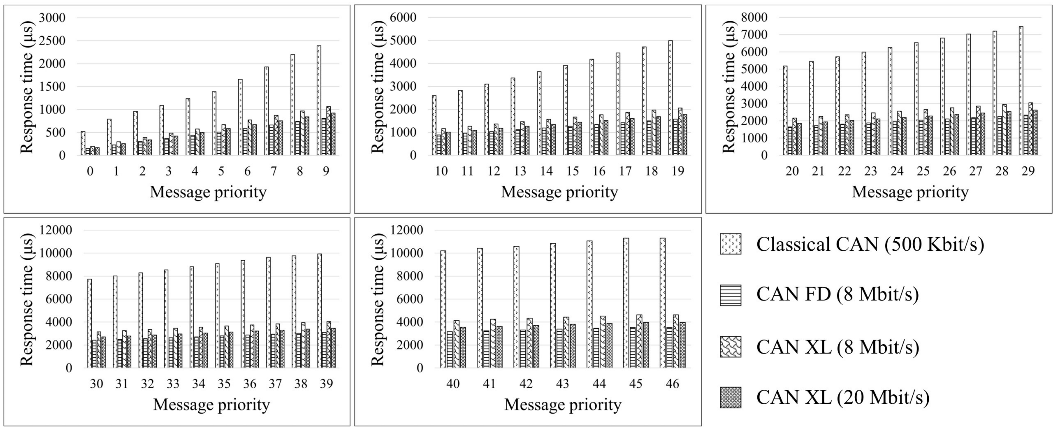

Fig. 5: Evaluation results for the industrial use case with bandwidth set to existing physical implementation limit.

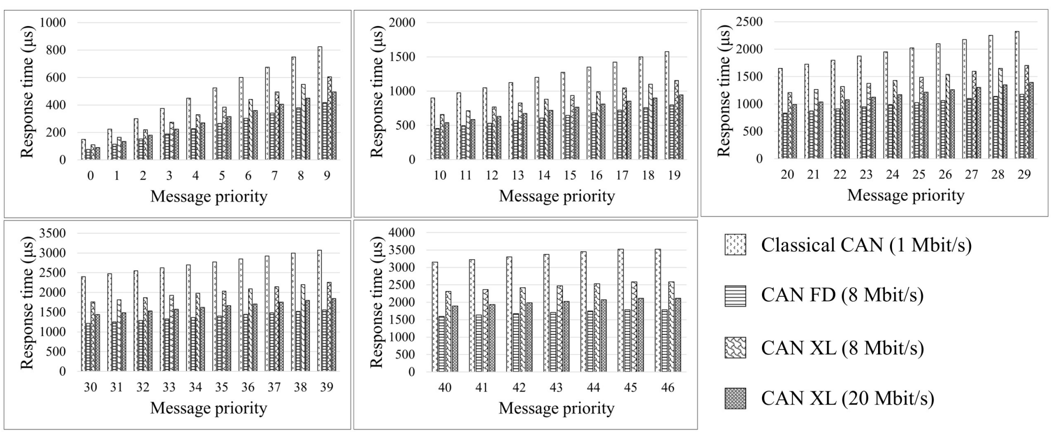

Fig. 6: Evaluation results for the industrial use case with bandwidth set to the theoretical limit.

Payload	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64	128	256	512	1048	2048
Classical CAN (1 Mbit/s)	63	75	85	95	105	115	125	135	230	270	365	405	540	810	1080	2160	4320	8640	17280	34560
CAN FD (8 Mbit/s)	36	38	39	40	41	43	44	45	50	55	61	66	76	96	116	232	464	928	1856	3712
CAN XL (8 Mbit/s)	54	55	56	57	58	59	60	62	66	70	75	79	88	106	123	194	334	616	1179	2306
CAN XL (20 Mbit/s)	43	44	44	45	45	46	46	47	48	50	52	54	57	64	71	99	156	268	494	944

TABLE III: Exact transmission times (μs) for various payloads of CAN frames with bit rate switching.

Payload	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64	128	256	512	1048	2048
Classical CAN (1 Mbit/s)	63	75	85	95	105	115	125	135	230	270	365	405	540	810	1080	2160	4320	8640	17280	34560
CAN FD (1 Mbit/s)	70	80	90	100	110	120	130	140	180	220	265	305	385	545	705	1410	2820	5640	11280	22560
CAN XL (1 Mbit/s)	175	184	193	202	210	219	228	237	272	307	342	378	448	589	730	1293	2419	4672	9178	18189

TABLE IV: Exact transmission times (μs) for various payloads of CAN frames without bit rate switching.

in a lower transmission time. However, as the payload size increases, classical CAN performs the worst, and CAN FD has the lowest transmission time until the payload of 64 bytes. Once the payload reaches 128 bytes, CAN XL takes over since only one CAN XL frame is needed for transmission.

Next, we evaluate the transmission times with bit rate

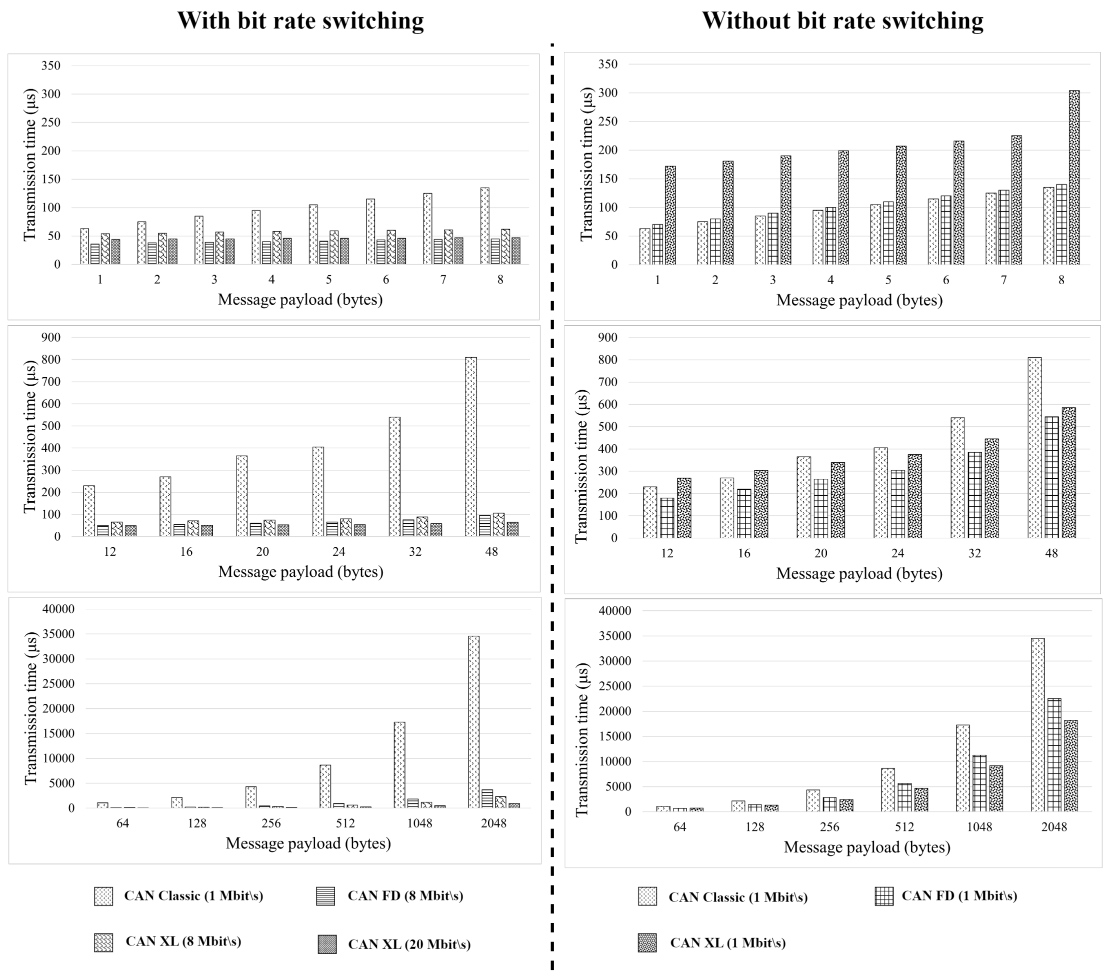

Fig. 7: Transmission time for various payloads of CAN frames with and without bit rate switching.

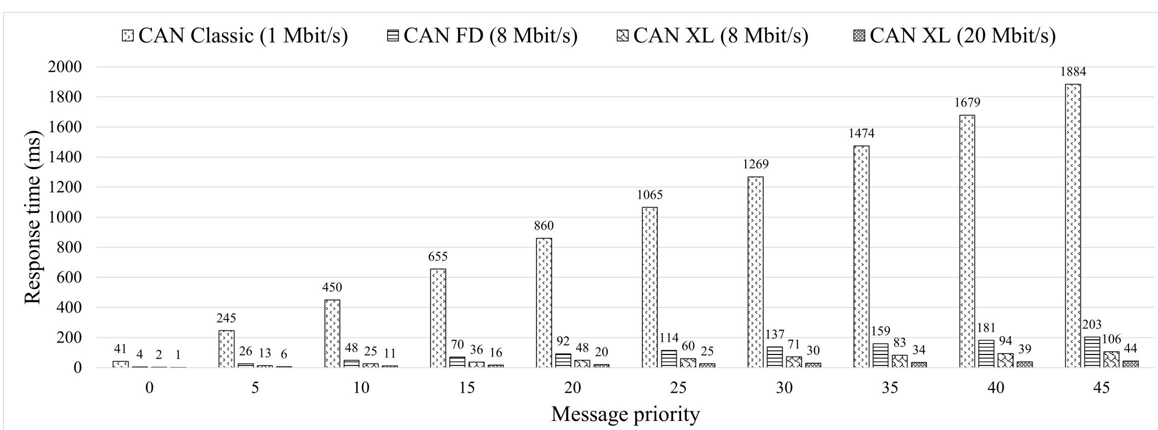

Fig. 8: Evaluation results with response time calculations frames with 2048 bytes payload.

switching turned on in the left column in Fig. 7 and the recorded transmission times of the frames are found in Table III. During the data phase, the transmission bit rate for CAN FD increased to 8 Mbit/s and CAN XL to 20 Mbit/s. To make a fair comparison with CAN FD, we also included CAN XL running at 8 Mbit/s. From the graphs, it is evident that CAN FD has the lowest transmission time until 12 bytes payload. After that, CAN XL with 20 Mbit/s bit rate performs better, while CAN XL running at 8 Mbit/s outperforms CAN FD when the payload exceeds 64 bytes since we start transmitting multiple CAN FD frames at that point.

D. Evaluation Results: Payload Sizes vs Response Times

In order to accurately determine the time required for data to reach its destination, it is not enough to add up the transmission time of multiple CAN frames. This is because other frames in the network can cause interference. To demonstrate this, we adapted the industrial use case with 47 frames. Each frame has a payload size of 2048 bytes and a period of 2 seconds due to bandwidth limitations. This allows for a more realistic comparison between the three generations of CAN. If a CAN generation does not support the frame payload, the 2048 bytes frame is split into multiple CAN frames for transmission, each with the same period as that of the original 2048 bytes frame. The frames are prioritized in ascending order, and we use the extended identifier format since there are more than 2048 CAN frames in this scenario. This means that the frame carrying the first segment of the original frame has the highest priority, while the frame with the last segment has the lowest. Each frame's priority is based on the original 2048 bytes frame's priority, ensuring that it is neither higher nor lower than the original frame's priority. When a 2048 bytes frame is converted into multiple frames, the response time is equal to the lowest priority frame's response time among all segments. The bit rate for the CAN bus remains unchanged, with classical CAN and arbitration phase in CAN FD and CAN XL set at 1 Mbit/s for the industrial use case.

Fig. 8 shows the evaluation results. We chose to display only every fifth frame out of the total 47 frames because the response times trend is consistent across all frames. The graph clearly demonstrates that classical CAN has a higher response time compared to CAN FD or CAN XL. This is because classical CAN requires 256 frames to transmit a 2048-byte frame, which results in a lot of interference that these frames can experience from higher priority frames. In contrast, CAN FD only needs 32 frames to transmit the same 2048-byte frame and supports a higher transmission rate, leading to a significantly improved response time. CAN XL at 20 Mbit/s performed the best, followed by CAN XL at 8 Mbit/s. CAN XL's superior performance is due to its capability of transmitting a single 2048-byte frame, avoiding the need for multiple transmissions.

VII. Conclusion

In this paper, we conducted a comparison of Classic CAN, CAN FD, and CAN XL, which are three different generations of Controller Area Network. We used response-time analysis to evaluate their performance. To make our analysis compatible with all generations of CAN, we included worst case transmission times calculation for all CAN generations. Additionally, we considered the different transmission bit rates for arbitration and data phases used in CAN FD and CAN XL.

The comparative evaluation, conducted on an automotive industrial use case, revealed that each generation of CAN may be preferred in particular use-case scenarios with respect to the requirements on the network speed and data payload in the frames. In the use cases that require up to 8 bytes of payload in the frames and the maximum network speed of 1 Mbit/s, then classical CAN is the most suitable option to use. However, if higher network speeds are required with up to 8 bytes of payload in the frames then CAN FD outperforms both classical CAN and CAN XL. Although CAN XL has higher network speeds than CAN FD, it does not improve response times of the frames when the frame payload is up to 8 bytes due to higher transmission overheads of CAN XL frames. Similarly, in the use cases that require 12-64 bytes of payload in the frames and the maximum network speed of 8 Mbit/s, then CAN FD, once again, outperforms the classical CAN and CAN XL. However, if the transmission speeds required are higher than 8 Mbit/s and/or if the payload requirement is higher than 64 bytes then CAN XL outperforms the classical CAN and CAN FD. These results can provide guidelines in the decision-making process for which CAN generation to use in CAN-based vehicular embedded systems.

Acknowledgment

This work is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the INTER-CONNECT, PROVIDENT and DESTINE projects and by the Swedish Knowledge Foundation via the projects DPAC & HERO. We would like to thank all industrial partners, in particular HIAB, Arcticus Systems and Volvo Construction Equipment.

References

[1] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[2] ISO 11898-1, "Road Vehicles - interchange of digital information - controller area network (CAN) for high-speed communication, ISO Standard-11898, Nov. 1993."

[3] R. Bosch, "CAN with flexible data-rate specification," *Robert Bosch GmbH, Stuttgart*, 2012.

[4] C. in Automation (CiA) e.V., "CiA draft specification 610-1 CAN XL specifications and test plans — part 1: Data link layer and physical coding sub-layer requirements — version 1.0.0," March 2022.

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

[6] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, pp. 239–272, 2007.

[7] H. Kim, W. Yoo, S. Ha, and J. Chung, "In-vehicle network average response time analysis for CAN-FD and automotive ethernet," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[8] Y. Xie, P. Huang, W. Liang, and Y. He, "Comparison between can and can fd: A quantified approach," in *IEEE International Symposium on Parallel and Distributed Processing with Applications and IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. IEEE, 2017, pp. 1399–1403.

[9] P. M. Yomsi, D. Bertrand, N. Navet, and R. I. Davis, "Controller area network (CAN): Response time analysis with offsets," in *WFCS*, 2012.

[10] 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*, 2011.

[11] S. Mubeen, J. Mäm-Turja, and M. Sjödin, "Extending schedulability analysis of controller area network (CAN) for mixed (periodic/sporadic) messages," in *IEE ETFA*, 2011.

[12] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Worst-case response-time analysis for mixed messages with offsets in controller area network," in *IEEE ETFA*, 2012.

[13] ——, "Extending worst case response-time analysis for mixed messages in controller area network with priority and FIFO queues," *IEEE ACCESS*, vol. 2, pp. 365–380, 2014.

[14] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Integrating Mixed Transmission and Practical Limitations with the Worst-Case Response-Time Analysis for Controller Area Network," *Journal of Systems and Software*, vol. 99, no. 0, pp. 66 – 84, 2015.

[15] U. D. Bordoloi and S. Samii, "The frame packing problem for CAN-FD," in *IEEE RTSS*, 2014.

[16] O. Ikumapayi, H. Olufowobi, J. Daily, T. Hu, I. C. Bertolotti, and G. Bloom, "Canasta: Controller area network authentication schedulability timing analysis," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2023.

[17] C. in Automation (CiA) e.V., "CiA draft specification 610-3 — CAN XL specifications and test plans — part 3: Physical medium attachment sub-layer requirements — version 1.0.0," March 2022.

[18] K. Tindell, A. Burns, and A. J. Wellings, "Calculating controller area network (CAN) message response times," *Control engineering practice*, vol. 3, no. 8, pp. 1163–1169, 1995.