Comparing the inaccessibility characteristics of CAN and CAN FD protocols

João de Sousa Alves and José Rufino

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal,
fc41937@alunos.fc.ul.pt, jmrufino@ciencias.ulisboa.pt *

Abstract. The Controller Area Network (CAN) protocol, originally designed more than two decades ago, has been widely used in multiple domains, including industrial control, automotive and aerospace.
To overcome two important limitations of the original CAN protocol, low data transmission rates and small data frame payload sizes, a new specification, the CAN with Flexible Data rate (CAN FD), was issued.
This paper addresses timing and safety issues of the new specification, demonstrating that CAN FD, although showing an improvement of timeliness in the absence of errors, continues to exhibit (almost) the same shortcomings of the original protocol with respect to its use in the design of highly reliable real-time embedded systems.

Keywords: Real-time and embedded systems; Real-time communications; Reliability and safety; Non-intrusive system observability.

1 Introduction

The Controller Area Network (CAN) [6, 12] is a simple and robust protocol. Originally designed by Bosch [12] for automotive applications it is widely used in a number of other domains such as industrial control, medical devices, transportation and vehicular applications, as well as aerospace.

Nevertheless, a key issue was that the original CAN protocol design exhibit two important limitations [6, 12], which may preclude the application of CAN in more general domains. These limitations were: low data transmission rates (only up to 1 Mbps) and small data frame payload sizes (8 bytes, maximum).

An effort to overcome those limitations has resulted in the definition of a new protocol specification, known as CAN FD, the CAN with Flexible Data rate protocol [13]. CAN FD allows message encapsulation with payloads up to 64 bytes and switching to a faster bit signalling rate after network access, through the original CAN deterministic node/message arbitration scheme, has been decided. This paper studies and analyses the timing and safety characteristics of the new

* This work was partially supported by FCT, through funding of LaSIGE Research Unit, ref. UID/CEC/00408/2013, and FCT/CAMPUS FRANCE (PHC PESSOA), through the transnational cooperation project NORTH. This work integrates the activities of COST Action IC1402 - Runtime Verification beyond Monitoring (ARVI), supported by COST (European Cooperation in Science and Technology).

specification, under normal operating conditions and in the presence of errors, aiming to provide an answer to one fundamental question: did it worth it?

The paper is structured as follows. Section 2 briefly presents the CAN protocol, its frame formats and the evolution into CAN FD. Section 3 analyses the timing of both CAN and CAN FD protocols under normal operating conditions while Section 4 addressees their inaccessibility characteristics. A comparative evaluation of CAN and CAN FD timing properties is provided in Section 5. Sections 6 and 7 respectively address the related work and conclude the paper.

2 Controller Area Network

CAN is a multi-master network that uses a twisted pair cable as transmission medium [6, 12, 4]. The network maximum length depends on the data rate. The maximum values are: 40m @ 1 Mbps [4]. A single bit can be transmitted in the bus at a time. The signalling of a bit in the bus takes one of two values: *recessive*, also the state of an idle bus; *dominant*, which always overwrites a recessive value. This behaviour, together with the use of unique frame identifiers, is exploited for bus arbitration. A *carrier sense multi-access with deterministic collision resolution* policy is used [6, 12]. When several nodes compete for bus access, the node transmitting the frame with the lowest identifier always gets the bus. Frames that have lost arbitration or have been destroyed by errors are automatically scheduled for retransmission.

CAN FD: CAN with Flexible Data rate

The CAN FD protocol has two differentiating features: it enables the transmission of data frames with a payload up to 64 bytes; it introduces a secondary data rate at which part of the data frame can be transmitted. Limitations of current transceiver[1] technology do not allow secondary data rates higher than 8 Mbps [7]. For transmitting at an higher data rate, CAN FD [13] enables a mechanism of data rate switch at the BRS bit (Bit Rate Switch) of the data frame (Figure 1). A recessive BRS bit establishes a boundary separating:

- *arbitration-phase* - which uses the "normal" bit signalling rate, allowing the CAN deterministic arbitration scheme to operate properly;
- *data-phase* - enabling the use of a "higher" rate for bit signalling.

A third phase, also performed at the normal bit signalling rate, intended for frame acknowledgement, terminates the CAN FD data frame transmission.

Frame Formats

A *data frame* is a piece of encapsulated information disseminated on the network, which contains as payload a *message*, a user-level piece of information. A *remote*

[1] A transceiver (abbreviation of transmitter/receiver) is a physical (PHY) layer device that links a node to the network cabling infrastructure.

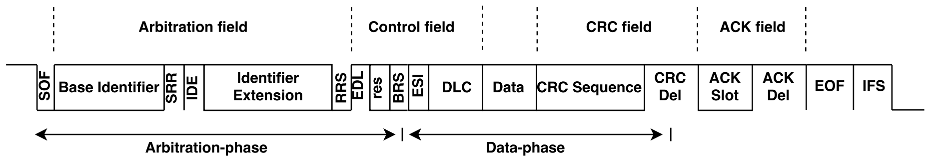

Fig. 1. CAN FD data frame structure and format.

frame has no payload and can be used to request the transmission of the data frame having the same identifier. Thus:

- *data frame* - includes the mandatory 11-bit frame's base identifier and possibly the optional 18-bit frame's identifier extension, as well as a number of control bits, which includes DLC, the 4-bit Data Length Code. DLC directly specifies the frame's payload length up to 8 bytes, the maximum payload size of the original CAN data frame. The remaining values encode the discrete lengths of 12, 16, 20, 24, 32, 48 and 64 byte in CAN FD data frames.
- *remote frame* - a CAN remote frame has a zero length payload and is identified by transmitting a recessive value in the RTR (Remote Transmission Request) bit. Since there is no advantage in encapsulating remote frames in CAN FD transmissions, it is replaced in CAN FD with a dominant value Remote Request Substitution (RRS) bit (Figure 1). This means, the remote frame format is not defined in CAN FD [13].

In the signalling of abnormal network operation incidents, the CAN protocol uses: *error frames*, for (global) error signalling; *overload frames*, to react to violations of the standard interframe spacing [6].

The signalling of an error detected during the data-phase of CAN FD implies switching first to the normal bit signalling rate which is then followed by the issuing of a standard CAN error frame. Thus, the CAN FD protocol does not formally defines nor uses own error/overload frame formats [13].

3 Analysing CAN and CAN FD Timings

This section establishes a set of easy-to-use expressions allowing the computation of the minimum and maximum normalised durations of a frame. The evaluation is independent of the actual bit signalling rate. To obtain the real duration of a frame, the normalised frame duration must be multiplied by the nominal bit time, $t_{bit} = \frac{1}{baud}$, being *baud* the nominal rate of bit signalling, on the bus. The normalised duration of a bit is represented by $\mathcal{T}_{bit}$, being $\mathcal{T}_{bit} = 1$ *bit-time*.

CAN Data and Remote frames

The main parameters that define the normalised duration of a frame are the frame format specification (base or extended) and the size of the payload field. With exception of the end-of-frame sequence, all the fields in a CAN frame are subject to dynamic bit-stuffing coding. To establish a lower bound (^{lb}) for the duration of a data frame, we assume no bits are stuffed in the outgoing stream[2]:

$$\mathcal{T}^{lb}_{data} = (l_{fix} + l_{dlc} + l_{data} + l_{efs}) \,.\, \mathcal{T}_{bit} \tag{1}$$

where the meaning of the different length parameters in equation (1) is:

- l_{fix} is the length (in bits) of the fixed size fields subject to dynamic bit-stuffing. It includes the dominant one-bit *start-of-frame* (SOF) delimiter, the frame identifier and control bits, as well as the CRC sequence. The exact value depends on the CAN frame format specification (base or extended). However, it does not include l_{dlc}, the 4-bit DLC field;
- l_{data} is the length (in bits) of the payload field. It varies between 0 and 64, in 8 bit increments, being also subject to dynamic bit-stuffing;
- l_{efs} is the length (in bits) of the fixed form sequence, not subject to bit-stuffing, that ends every data or remote frame. It includes the CRC delimiter, the 2-bit acknowledgement field and the 7-bit *end-of-frame* delimiter.

To establish an upper bound (^{ub}) for the duration of a data frame, we assume that all the fields subject to dynamic bit-stuffing exhibit a pattern that leads to the maximum insertion of stuffed bits. Therefore:

$$\mathcal{T}^{ub}_{data} = \left(l_{fix} + l_{dlc} + l_{data} + \left(1 + \left\lfloor \frac{l_{fix} - l_{stuff} + l_{dlc} + l_{data}}{l_{stuff} - 1} \right\rfloor \right) + l_{efs} \right) . \, \mathcal{T}_{bit} \tag{2}$$

where $\lfloor\ \rfloor$ represents the *floor* function[3]; l_{stuff} represents the bit-stuffing width, i.e. the maximum number of consecutive bits of identical value that can be found in the outgoing stream, stuffed bits included. In the worst case, the first (recessive) stuffed bit is inserted in the outgoing stream immediately after the transmission of l_{stuff} initial dominant bits, starting with the SOF delimiter.

The minimum and maximum durations of a data frame can be derived by setting l_{data} to zero in equation (1) and by setting l_{data} to the maximum value (64 bits) in equation (2), respectively. For obtaining the minimum and maximum durations of a remote frame, l_{data} must be set to zero in equations (1) and (2).

CAN FD Data frames

The changes introduced by the CAN FD protocol [13] in the format of data frames, lead to the following updates to equations (1) and (2):

$$\mathcal{T}^{FD-lb}_{fddata} = (l_{fdix} - 1) \,.\, \mathcal{T}_{bit} + (1 + l_{dlc} + l_{data} + l_{fdcrc}) \,.\, \frac{\mathcal{T}_{bit}}{\sigma} + l_{efs} \,.\, \mathcal{T}_{bit} \tag{3}$$

[2] Equations (1) up to (8) do not account for the nominal three bit bus idle period, the interframe space, t_{IFS}, that usually precedes any data or remote frame and ends every CAN or CAN FD frame transmission (*intermission*).

[3] The *floor* function $\lfloor x \rfloor$ is defined as the greatest integer not greater than x.

$$\mathcal{T}_{fddata}^{FD-ub} = \left(l_{fdix} + \left\lfloor \frac{(l_{fdix}-1)-l_{stuff}}{l_{stuff}-1} \right\rfloor \right) . \mathcal{T}_{bit} + \left(1 + l_{dlc} + l_{data} + l_{fdcrc} + \left\lfloor \frac{(1+l_{dlc}+l_{data})}{l_{stuff}-1} \right\rfloor \right) . \frac{\mathcal{T}_{bit}}{\sigma} + l_{efs} . \mathcal{T}_{bit} \tag{4}$$

where $\sigma = \frac{baud\_high}{baud\_normal}$, is defined as the ratio between the higher and the normal data rates. The meaning of the new length parameters is as follows:

- l_{fdix} is the length (in bits) of fixed size fields subject to dynamic bit-stuffing transmitted at the lower data rate, which in CAN FD excludes the 4-bit DLC field and the full CRC sequence, as shown in Figure 1;
- l_{fdcrc} is the length (in bits) of the CRC sequence, which varies according to the size of the payload: 17 bit for payloads up to 16 byte; 21 bit for payloads longer than 16 byte. This parameter also includes the statically inserted fixed stuffed bits (FSB) in the CRC sequence, as illustrated in Figure 2.

The mandatory values taken by a relevant set of control bits, such as SRR, IDE, RRS, and EDL, in the CAN FD frame format (Figure 1), are not considered in the definition of Equation (4), which thus slightly overestimates the maximum number of bits that may be dynamically stuffed in the outgoing stream.

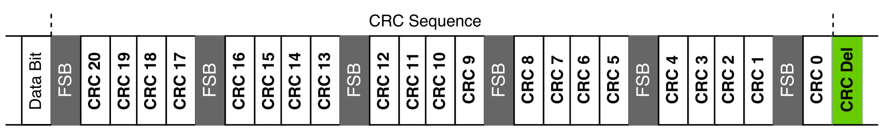

Fig. 2. CAN FD CRC 21 sequence with statically stuffed bits, as specified in [13].

Error and overload frames

Error and overload frames are not subject to bit-stuffing coding. The evaluation of their best (^{bc}) and worst case (^{wc}) durations is extremely simple. Equations (5) and (6) specify those durations for error frames:

$$\mathcal{T}_{error}^{bc} = (l_{flag} + l_{del}) . \mathcal{T}_{bit} \tag{5}$$

$$\mathcal{T}_{error}^{wc} = (2 . l_{flag} + l_{del}) . \mathcal{T}_{bit} \tag{6}$$

where l_{flag} and l_{del} are, respectively, the length in bits of the (dominant) six-bit *error flag* and of the recessive eight-bit *error delimiter*. Similar expressions can be established with regard to the evaluation of overload frame durations:

$$\mathcal{T}_{oload}^{bc} = (l_{flag} + l_{del}) . \mathcal{T}_{bit} \tag{7}$$

$$\mathcal{T}_{oload}^{wc} = (2 . l_{flag} + l_{del}) . \mathcal{T}_{bit} \tag{8}$$

4 Analysing CAN and CAN FD Accessibility Constraints

Data frame transfers are susceptible to errors due to bit corruption, e.g. by electromagnetic interference. Such errors may lead to periods of inaccessibility, a state where the network temporarily refrains from providing service, without that having to be necessarily considered a failure [18]. Networks have means of recovering from those situations. However, the recovery process takes time. So, in the meantime, the network is inaccessible.

This section addresses the inaccessibility characteristics of CAN FD, taking as basis previous studies on CAN [15, 19, 14]. In contrast with those studies, aiming comprehensive analyses, one address only errors affecting data frame transfers, the single protocol unity defined in CAN FD. Given CAN FD ability of transferring data at higher rates, susceptibility to errors increases.

Bounded and known message delivery latency, in the presence of faults, is fundamental to secure a real-time operation of CAN and CAN FD. Thus, this section is focused on a worst case analysis, being the exception situations where error signalling is issued in a pre-determined bit, making analytical computation of inaccessibility periods dependent only of data frame's payload size.

Bit Errors

In both protocols, transmitter-based error detection is achieved by listening to the bus while transmitting and comparing both streams on a bit-by-bit basis. A recessive level issued on the bus, by a given transmitter, can only be received back as dominant, without that being considered an error, in the following exceptional circumstances:

- inside the *arbitration field*, meaning the loss of the arbitration process;
- during the *acknowledgment slot*, meaning that at least one node has not detected, so far, any error in the current transmission;
- when a *passive* recipient[4] is transmitting an error flag.

All other situations are errors, signalled on the bus by starting the transmission of an error frame at the next bit slot. In CAN FD, this may imply switching back from the higher to the normal bit signalling rate. The corruption of the transmitted bit-stream cannot occur later than the transmission of the last bit of the *end-of-frame* delimiter. The corresponding inaccessibility period is obtained considering the transmission of data and error frames with the maximum size, for both CAN (Equation 9) and CAN FD (Equation 10) protocols:

$$\mathcal{T}^{wc}_{ina \leftarrow berr} = \mathcal{T}^{wc}_{data} + \mathcal{T}^{wc}_{error} + \mathcal{T}_{ifs} \tag{9}$$

<u>CAN FD</u>:

$$\mathcal{T}^{FD-wc}_{fdina \leftarrow berr} = \mathcal{T}^{FD-wc}_{fddata} + \mathcal{T}^{wc}_{error} + \mathcal{T}_{ifs} \tag{10}$$

[4] A node enters the passive state after being persistently affected by errors [6, 12, 13].

Bit-Stuffing Errors

The CAN and CAN FD protocols also use receiver-based error detection mechanisms. The first receiver-based error detection scheme to be analysed performs the monitoring of *bit-stuffing* violations. Whenever a receiving node monitors l_{stuff} consecutive bits of identical level, it dynamically and automatically deletes the next received (stuffed) bit. Under error free operation, the deleted bit presents a polarity opposite to the preceding ones; should this condition be violated, an error will be signalled on the bus, by starting the transmission of an error frame, at the next bit slot.

In CAN, the dynamic *bit-stuffing* coding scheme is used in the transmission of data and remote frames, up to the end of the 15-bit *CRC field*. A bit-stuffing error cannot occur after the reception of the last bit of the CRC field. Again, the worst case duration for an inaccessibility period occurs upon the transmission of a maximum size data frame, as specified by equation:

$$\mathcal{T}_{ina \leftarrow stuff}^{wc} = \mathcal{T}_{data}^{wc} - \mathcal{T}_{EFS} + \mathcal{T}_{error}^{wc} + \mathcal{T}_{ifs} \qquad (11)$$

where $\mathcal{T}_{EFS} = l_{efs} \,.\, \mathcal{T}_{bit}$, accounts for the normalised duration of the fixed form sequence – not subject to bit-stuffing coding – that ends every data and, in the case of the CAN protocol, every remote frame.

In CAN FD, the *CRC sequence* is not subject to dynamic *bit-stuffing* coding and, as such, a bit-stuffing error can only be detected up to reception of the last bit of the payload and, consequently, signalled on the bus up to the reception of the first bit of the CRC sequence (Figure 1). Thus:

CAN FD: $$\mathcal{T}_{fdina \leftarrow stuff}^{FD-wc} = \mathcal{T}_{fddata}^{FD-wc} - \mathcal{T}_{FD\_CRC} - \mathcal{T}_{EFS} + \mathcal{T}_{error}^{wc} + \mathcal{T}_{ifs} \qquad (12)$$

where $\mathcal{T}_{FD\_CRC} = l_{fdcrc} \,.\, \mathcal{T}_{bit}/\sigma$, accounts for the normalised duration of the CRC sequence (Figure 2), fully transmitted, in the CAN FD protocol, at the secondary bit signalling rate.

Bit errors may occur in a way that they do not produce a violation of the bit-stuffing policy. In this case, the error will be detected slightly afterwards, either by *CRC checking* or through the detection of a frame format violation.

CRC Errors

After the reception of the *CRC sequence*, a node should do one of two things, depending on whether the CRC was good:

- if the CRC is correct, the node is obliged to change the bus level from recessive to dominant, during the *acknowledgement slot*;
- otherwise, the node does not modify the value of the *acknowledgement slot*, and signals the CRC error through the transmission of an error frame, that starts immediately after the *acknowledgement delimiter*. This allows to distinguish a situation where the CRC error is detected in all the nodes from the situation where such an error is detected only by a subset of the recipients.

The corresponding inaccessibility time is generically given by expression:

$$\mathcal{T}_{ina \leftarrow crc} = \mathcal{T}_{data} - \mathcal{T}_{EOF} + \mathcal{T}_{error} + \mathcal{T}_{ifs} \quad (13)$$

where $\mathcal{T}_{EOF}$ represents the normalised duration of the *end-of-frame* delimiter. The best and worst case bounds of equation (13) can be obtained considering the shortest and the longest data and error frame durations. An identical description is also valid for the CAN FD protocol. Thus:

CAN FD: $$\mathcal{T}_{fdina \leftarrow crc} = \mathcal{T}_{fddata} - \mathcal{T}_{EOF} + \mathcal{T}_{error} + \mathcal{T}_{ifs} \quad (14)$$

Acknowledgment Errors

Whenever a transmitter does not monitor a dominant level on the bus during the *acknowledgment slot*, it interprets that as an error, and the transmission of an error frame is started at the next bit slot. So, the duration of the corresponding network inaccessibility period is generically given by equation:

$$\mathcal{T}_{ina \leftarrow ack} = \mathcal{T}_{data} - \mathcal{T}_{EFS} + 2 \,.\, \mathcal{T}_{bit} + \mathcal{T}_{error} + \mathcal{T}_{ifs} \quad (15)$$

The best and worst case inaccessibility bounds are obtained as usual, i.e. considering the minimum and the maximum durations for data and error frames. Specifically, in the CAN FD protocol:

CAN FD: $$\mathcal{T}_{fdina \leftarrow ack} = \mathcal{T}_{fddata} - \mathcal{T}_{EFS} + 2 \,.\, \mathcal{T}_{bit} + \mathcal{T}_{error} + \mathcal{T}_{ifs} \quad (16)$$

In the presence of multiple errors, it may happen the transmitter monitors a faulty dominant level, at the *acknowledgement slot*. The transmitter cannot detect such an error. However, the lack of success in the transfer of the frame is signalled by the recipients that having detected a CRC error issue a negative acknowledgement, in the form of an error frame, at the bit slot immediately after the *acknowledgement delimiter*.

Form Errors

All the frames used by the CAN protocol obey to a few pre-defined formats. Data (and remote) frames have a fixed form end-sequence, where:

- the *CRC delimiter* and the *acknowledgement delimiter*, should always exhibit recessive values;
- the *end-of-frame delimiter*, consists of seven consecutive recessive bits.

An error in the frame ending sequence implies the transmission of an error frame. The longest period of inaccessibility caused by a form error is when it occurs while receiving the last but one bit of the *end-of-frame* delimiter, because a receiver monitoring a dominant level at the last bit of this delimiter does not take that as a form error[5]. Thus, one will have for the CAN and CAN FD protocols, respectively:

$$\mathcal{T}^{wc}_{ina\leftarrow form} = \mathcal{T}^{wc}_{data} - \mathcal{T}_{bit} + \mathcal{T}^{wc}_{error} + \mathcal{T}_{ifs} \quad (17)$$

CAN FD:
$$\mathcal{T}^{FD-wc}_{fdina\leftarrow form} = \mathcal{T}^{FD-wc}_{fddata} - \mathcal{T}_{bit} + \mathcal{T}^{wc}_{error} + \mathcal{T}_{ifs} \quad (18)$$

5 Evaluation and Discussion

This section aims to provide a comparative evaluation of CAN and CAN FD timing characteristics, both under normal operation and in the presence of errors.

Parameters	Frame field length (*bit*)							
CAN Parameters	l_{bid}	l_{eid}	l_{ctl}	l_{crc}	l_{fix}	l_{data}		
CAN Base format (2.0A)	11	-	4	15	30	≤ 64		
CAN Extended format (2.0B)	11	18	6	15	50	≤ 64		
CAN FD Parameters	l_{bid}	l_{eid}	l_{ctl}	l_{fdix}	l_{data}	l_{crc}	l_{fsb}	l_{fdcrc}
CAN FD Base format	11	-	7	18	≤ 128	17	5	22
	11	-	7	18	> 128	21	6	27
CAN FD Extended format	11	18	8	37	≤ 128	17	5	22
	11	18	8	37	> 128	21	6	27

l_{bid} and l_{eid}, are the base and extended identifier field length, respectively;
l_{crc} - CRC field length;
l_{ctl} - number of control bits, including the 1-bit SOF but excluding the 4-bit DLC field;
l_{fsb} - number of statically stuffed bits, in CAN FD.

Table 1. Field length parameters for different frame formats.

Frame	Normalised worst case frame durations			
	$\mathcal{T}$ (*bit-times*)		$\mathcal{T}$ (*bit-times*)	
	CAN Base (2.0A)	CAN Extended (2.0B)	CAN FD Base $\sigma = 8$	CAN FD Extended $\sigma = 8$
Data frame	132,0	157,0	115,1	138,1
Remote frame	52,0	77,0	-	-
Error frame	20,0	20,0	-	-
Overload frame	20,0	20,0	-	-

Table 2. Normalised CAN and CAN FD frame durations.

[5] This will be considered an *early reactive overload error*, whose analysis is out of the scope of this paper, but that was thoroughly studied in [15, 19, 14].

Fundamental parameters required for the assessment of both frame durations and periods of inaccessibility are established in Table1. The normalised worst case duration of the different frames, for the CAN protocol, and of data frames, for the CAN FD protocol, are inscribed in Table 2. The CAN FD protocol uses a speedup factor $\sigma = 8$ [7].

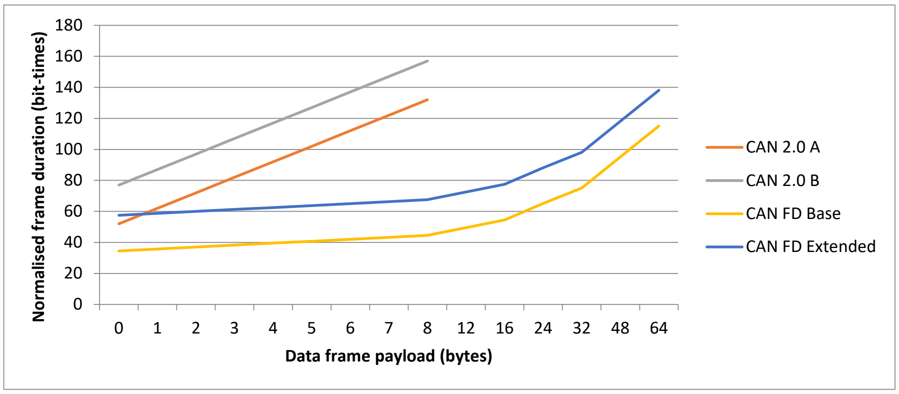

Fig. 3. Normalised CAN and CAN FD frame durations as a function of payload size.

The normalised data frame duration, for both protocols and for different payloads, is graphically represented in Figure 3. Again, the CAN FD protocol uses a speedup factor $\sigma = 8$ [7]. The timeliness of network operation, in the absence of errors, is thus summarily analysed in Figure 3.

The impact of CAN FD in the decrease of data frame transmission times and, therefore, in the system end to end message delivery latency, is evident in the graphic representation of Figure 3: for the same data frame format (e.g., 2.0B/extended) the data frame transmission times are always lower; for a zero payload the simple transmission of the CRC sequence (see, Figure 1) at a higher bit signalling rate is sufficient to lead to a lower data frame transmission time; as the data frame payload size increases, the difference between data frame transmission times, in the two protocols, becomes more significant.

Conversely, the improvements in network timing behaviour in the presence of faults are not particularly impressive, as shown by the numerical results of Table 3. Comparing the analytical expressions, already established for the CAN protocol [15, 19, 14], with those explicitly derived in this paper for the assessment of CAN FD inaccessibility periods, one conclude that, for most cases, the main difference concern the worst case duration of a data frame transmission.

Since the CAN FD protocol also specifies an increase in the maximum data frame payload size (from 8 byte in CAN to 64 byte in CAN FD), the positive effect of transferring the data frame payload at a higher bit signalling rate is counteracted by the need of transmitting a larger data frame payload. A speedup factor $\sigma \gg 8$ in the operation of the CAN FD protocol could mitigate such an effect, but that is not enabled by existing transceivers [7].

Error scenarios	Normalised worst case inaccessibility			
	$\mathcal{T}_{ina}$ (*bit-times*)		$\mathcal{T}_{fdina}$ (*bit-times*)	
	CAN Base (2.0A)	CAN Extended (2.0B)	CAN FD Base $\sigma = 8$	CAN FD Extended $\sigma = 8$
Bit errors	155,0	180,0	138,1	161,1
Bit-stuffing errors	145,0	170,0	124,8	147,8
CRC errors	148,0	173,0	131,1	154,1
Acknowledgement errors	147,0	172,0	130,1	153,1
Form errors	154,0	179,0	137,1	160,1

Table 3. Normalised CAN and CAN FD worst case inaccessibility bounds.

6 Related Work

A large body of research has been produced around the CAN protocol in the last two decades, addressing several timing related issues, namely: message schedulability analysis [17, 5]; performance of coding mechanisms [11] and behaviour in the presence of errors [3, 15], including their probability of occurrence [2]; integration of error behaviours in message schedulability analysis [9, 10].

Many of these works need now to be revisited and reformulated in the context of the CAN FD protocol. This paper is a step towards that direction, being the first contribution to a comprehensive analysis of CAN FD both in the absence and in the presence of errors.

Previous works on CAN FD are not numerous, being almost restricted to: the performance analysis under normal operating conditions [1]; a modelling and simulation study revisiting the automotive industry SAE benchmark in the context of CAN FD [16]; a detailed study of the improvements introduced in CAN FD with respect to error detection mechanisms [8].

7 Conclusion

This paper presented a comparative analysis of CAN and CAN FD protocols timing, both in the absence and in the presence of errors. Overall there is an improvement of timeliness characteristics but currently available CAN FD transceiver technology place a limit on such improvements. Nevertheless, in the absence of faults there is a significant improvement (in the order of 50%) in terms of the data transfers times for the same data payload length. In addition, the maximum data payload size has increased eight times, up to a maximum of 64 bytes, while maintaining a similar worst case message delivery latency bound.

In the presence of errors, given its correlation with the data frame duration upper bound, the improvements are less significant, exhibit lower numerical values and only a very modest enhancement is achieved.

On the other hand, the complexity of the protocol specification has increased. In a first analysis, some design decisions (e.g., the use of two alternative CRC polynomials in the CAN FD protocol) may unnecessarily lead to an increase of the resources required for the implementation of CAN FD machinery.

References

1. Bordoloi, U.D., Samii, S.: The frame packing problem for CAN-FD. In: Proceedings of the Real-Time Systems Symposium (RTSS). IEEE, Rome, Italy (Dec 2014)
2. Broster, I., Burns, A., Rodríguez-Navas, G.: Probabilistic analysis of CAN with faults. In: In Proc. 23rd Real-time Systems Symp. IEEE, Austin, Texas (Dec 2002)
3. Charzinski, J.: Performance of the error detection mechanisms in CAN. In: Proc. 1st International CAN Conference. pp. 1.20–1.29. CiA, Mainz, Germany (Sep 1994)
4. CiA - CAN in Automation: CAN Physical Layer for Industrial Applications - CiA Draft Standard 102 Version 2.0 (Apr 1994)
5. Davis, R.I., Burns, A., Bril, R.J., Lukkien, J.J.: Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. Real-Time Systems 35(3), 239–272 (2007)
6. ISO: International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network for high-speed communication (Nov 1993)
7. Microchip Technology Inc.: MCP2561/2FD - High-Speed CAN Flexible Data Rate Transceiver (2014)
8. Mutter, A., Hartwich, F.: Advantages of CAN FD error detection mechanisms compared to classical CAN. In: Proceedings of the 15th international CAN Conference (iCC). CiA, Vienna, Austria (Oct 2015)
9. Pinho, L., Vasques, F., Tovar, E.: Integrating inaccessibility in response time analysis of CAN networks. In: Proceedings of the 3rd International Workshop on Factory Communication Systems. pp. 77–84. IEEE, Porto, Portugal (Sep 2000)
10. Punnekkat, S., Hansson, H., Norstrom, C.: Response time analysis under errors for CAN. In: Proceedings of the Real-Time Technology and Applications Symposium. pp. 258–265. IEEE, Washington, USA (May 2000)
11. Rauchaupt, L.: Performance analysis of CAN based systems. In: Proc. of the 1st International CAN Conference. pp. 1.12–1.19. CiA, Mainz, Germany (Sep 1994)
12. Robert Bosch GmbH: CAN Specification Version 2.0 (Sep 1991)
13. Robert Bosch GmbH: CAN with Flexible Data-Rate - Version 1.0 (Apr 2012)
14. Rufino, J.: Computational System for Real-Time Distributed Control. Ph.D. thesis, Technical University of Lisbon, Inst. Superior Técnico, Lisboa, Portugal (Jul 2002)
15. Rufino, J., Verissimo, P.: A study on the inaccessibility characteristics of the Controller Area Network. In: Proceedings of the 2nd International CAN Conference. pp. 7.12–7.21. CiA, London, England (Oct 1995)
16. Tenruh, M., Oikonomidis, P., Charchalakis, P., Elias Stipidis, E.: Modelling, simulation, and performance analysis of a CAN FD system with SAE benchmark based message set. In: Proc. 15th Int. CAN Conference (iCC). CiA, Austria (Oct 2015)
17. Tindell, K., Burns, A.: Guaranteeing message latencies on Controler Area Network. In: Proc. 1st Int. CAN Conference. pp. 1.2–1.11. CiA, Mainz, Germany (Sep 1994)
18. Verissimo, P., Marques, J.A.: Reliable broadcast for fault-tolerance on local computer networks. In: Proceedings of the 9th Symposium on Reliable Distributed Systems. IEEE, Huntsville, Alabama-USA (Oct 1990)
19. Verissimo, P., Rufino, J., Ming, L.: How hard is hard real-time communication on field-buses? In: Digest of Papers, The 27th International Symposium on Fault-Tolerant Computing Systems. IEEE, Washington - USA (Jun 1997)