IPP-HURRAY! Research Group

Polytechnic Institute of Porto
School of Engineering (ISEP-IPP)

Integrating Inaccessibility in Response Time Analysis of CAN Networks

Luís Miguel PINHO
Francisco VASQUES (FEUP)
Eduardo TOVAR

HURRAY-TR-0004

June 2000

relatório técnico

technical report

THIS WORK IS PARTIALLY SUPPORTED BY FCT UNDER PROJECT DEAR-COTS (Praxis/P/EEI/14187/1998) AND BY FLAD UNDER PROJECT SISTER (Proj.471/97)

Integrating Inaccessibility in Response Time Analysis of CAN Networks

Luís Miguel PINHO, Eduardo TOVAR

IPP-HURRAY! Research Group
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340502, Fax: +351.22.8340529
E-mail: {lpinho,emt}@dei.isep.ipp.pt
http://www.hurray.isep.ipp.pt

Francisco VASQUES

University of Porto (FEUP)
Rua Dr. Roberto Frias
4050-123 Porto
Portugal
Tel.: +351.22.5081702, Fax:
E-mail: vasques@fe.up.pt
http://www.fe.up.pt/~vasques

Abstract:
Controller Area Network (CAN) is a fieldbus network suitable for small-scale Distributed Computer Controlled Systems, being appropriate for transferring short real-time messages. Nevertheless, it must be understood that the continuity of service is not fully guaranteed, since it may be disturbed by temporary periods of network inaccessibility [1].
In this paper, such temporary periods of network inaccessibility are integrated in the response time analysis of CAN networks. The achieved results emphasise that, in the presence of temporary periods of network inaccessibility, a CAN network is not able to provide different integrity levels to the supported applications, since errors in low priority messages interfere with the response time of higher priority message streams.

Integrating Inaccessibility in Response Time Analysis of CAN Networks

Luís Miguel Pinho [1], Francisco Vasques [2], Eduardo Tovar [1]

[1] Department of Computer Engineering,
ISEP, Polytechnic Institute of Porto
Rua São Tomé, 4200 Porto, Portugal
E-mail: {lpinho, emt}@dei.isep.ipp.pt

[2] Department of Mechanical Engineering
FEUP, University of Porto
Rua Bragas, 4050-123 Porto, Portugal
E-mail: vasques@fe.up.pt

Abstract

Controller Area Network (CAN) is a fieldbus network suitable for small-scale Distributed Computer Controlled Systems, being appropriate for transferring short real-time messages. Nevertheless, it must be understood that the continuity of service is not fully guaranteed, since it may be disturbed by temporary periods of network inaccessibility [1].

In this paper, such temporary periods of network inaccessibility are integrated in the response time analysis of CAN networks. The achieved results emphasise that, in the presence of temporary periods of network inaccessibility, a CAN network is not able to provide different integrity levels to the supported applications, since errors in low priority messages interfere with the response time of higher priority message streams.

1. Introduction

Controller Area Network (CAN) [2] was originally developed to be used within road vehicles to interconnect microprocessor-based components. More recently, CAN is also being considered for the automated manufacturing and distributed process control environments [3], and is being used as the communication interface in proprietary architectures, such as DeviceNet [4]. Several studies on how to guarantee the real-time requirements of messages in CAN networks are available (e.g. [5]), providing the necessary pre-run-time schedulability equations for the timing analysis of the supported traffic.

Nevertheless, a drawback of communication networks is that continuity of service is not fully guaranteed, since it may be disturbed by temporary periods of network inaccessibility (periods during which stations cannot communicate with each other, due to the existence of on-going error detection and recovery mechanisms). A study of the inaccessibility characteristics of CAN networks has been presented at [1], identifying the duration of its error detection and recovery periods.

In this paper, such temporary periods of network inaccessibility are integrated in the response time analysis of CAN networks, providing a more accurate analysis of its real-time behaviour. Essentially, formulae are provided to evaluate the response time of messages, considering a CAN network disturbed by temporary periods of inaccessibility.

The remainder of this paper is organised as follows. Section 2 describes the most important characteristics of CAN networks. Particular relevance is given to its error detection and recovery mechanisms. Section 4 describes the most relevant previous work on response time analysis and inaccessibility studies of CAN networks.

Based on the characteristics of the CAN protocol, in Section 5 the response time analysis of CAN networks is extended to integrate temporary periods of network inaccessibility. In Section 6, a benchmark is used to compare the results obtained using this extended analysis with those obtained using the classical analysis. Finally, in Section 7, the pessimism inherent to the proposed analysis is studied.

2. A Brief Description of the CAN Protocol

2.1. Main Characteristics

The CAN protocol implements a priority-based bus, with a carrier sense multiple access with collision avoidance (CSMA/CA) MAC. In this protocol, any station can access the bus when it becomes idle. However, contrarily to Ethernet-like networks, the collision resolution is non-destructive, in the sense that one of the messages being transmitted will succeed.

There are 4 types of frames that can be transferred in a CAN network. Two are used during the normal operation of the CAN network: the Data Frame, which is used to send local data and the Remote Frame, which is used to request remote data. The other two are used to signal an abnormal state of the CAN network: the Error Frame signals the detection of an error and the Overload Frame signals that a station is not ready to transmit data.

Fig. 1 shows the structure of a Data Frame (specific fields: SOF, Identifier, Control, DLC, CRC and EOF are described in [2]). A Remote Frame has the same structure (without data field) and identifier of the remotely requested Data Frame. The structure of both

the Error and Overload Frames will be presented in Section 2.2.

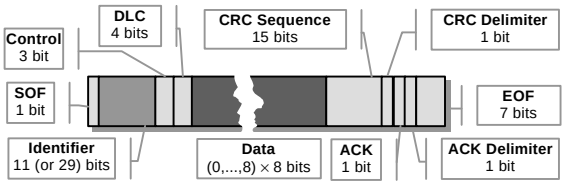

Fig. 1. Structure of a Data Frame

Bus signals can take two different states: *recessive bits* (idle bus), and *dominant bits* (which always overwrite recessive bits). The collision resolution mechanism works as follows: when the bus becomes idle, every station with pending messages will start to transmit. During the transmission of the identifier, if a station transmitting a recessive bit reads a dominant one, it means that there was a collision with a higher-priority message, and consequently the transmission is aborted. The highest-priority message (the one with most leading dominant bits on the identifier) being transmitted will proceed without perceiving any collision, and thus will be successfully transmitted. Stations that loose the arbitration phase will automatically retry the transmission of requested messages.

At the physical layer, frames are transmitted using the NRZ (Non Returning to Zero) coding technique, with the insertion of stuff bits. That is, whenever there are more than five equal consecutive bits (up to the end of the CRC Field), there is the insertion of an opposite bit in the frame. This opposite bit will be detected and removed by the physical layer at the receiving side. This bit stuffing technique ensures that, in the normal behaviour, there will never be more than 5 consecutive equal bits on the bus.

2.2. Error Detection and Recovery Mechanisms

In the CAN protocol, all stations continuously monitor every frame being transmitted on the bus, to detect any transmission error. The station which firstly detects an error, starts immediately the transmission of an Error Frame (violating the bit stuffing rule). As a consequence, all receiving stations know that the frame being transmitted has an error. An Error Frame has the following structure:

- 6-12 consecutive dominant bits (Error Flag). The station that firstly detects the error starts transmitting the Error Flag. If any other station only recognises the bit stuffing error induced by the Error Flag, it will start the transmission of a new Error Frame, thus the Error Flag will be up to 12 bits long;
- 8 consecutive recessive bits (Error Delimiter) which signal the end of the Error Frame.

Concerning the available error detection and signalling mechanisms, the CAN protocol has the following capabilities:

- Bit error: a transmitting station is continuously sensing its transmission; if the observed bit does not correspond to the transmitted one, the station signals a transmission error (except if the difference is observed in the Identifier or the ACK Slot Fields);
- CRC error: if the receiving station detects an error in the CRC code, it sends an Error Frame. This CRC code can detect up to 5 randomly modified bits or up to 15 consecutively modified bits on the CAN frame.
- Stuff bit error: whenever a receiving station detects a sixth equal consecutive bit, it signals a stuff bit error. Neither the Error Frame, nor the Overload Frame, are coded by the bit stuffing mechanism.
- Form error: if a station verifies that the structure of the received frame is not correct, it signals an error.
- ACK error: stations receiving a correct frame write a dominant bit on the ACK Field. A recessive bit on this field result either from the absence of receiving stations or from a transmission error recognised by every receiver. In such case, the transmitting station signals an error.
- Overload error: stations not ready to receive another frame may transmit one or two consecutive Overload Frames (with the same structure of the Error Frame).

Sending Error Frames is a very interesting mechanism to ensure that every station sees the same global state of the network (state coherence). However, a failure in one station may induce the transmission of consecutive Error Frames, blocking all the ongoing communications. To solve this problem, CAN controllers have two error counters (for transmitting and receiving errors, respectively) to isolate erratic stations. The values of these counters, which determine the operating state of the station, are increased or decreased (at different rates) as a function of the detected errors. These error counters acts as self-surveillance mechanisms, disconnecting the faulty station (fault-confinement techniques). There are three different operating modes:

- Error-Active, which is the normal operating mode.
- Error-Passive, where the station is still able to transfer / receive frames, but it must wait some time before transmitting (automatically decreasing its priority) and the error signalling is performed with passive Error Flags (6 recessive bits, thus not interfering with frames transmitted by other stations).
- Bus-Off, where the station is not able to transfer / receive frames.

3. System Model

3.1. Network and Message Models

This analysis assumes a network with n message streams defined as:

$$S_i = (C_i, T_i, D_i) \quad (1)$$

where S_i defines a message stream i characterised by a unique identifier. A message stream is a temporal

sequence of messages concerning, for instance, the remote reading of a specific process variable. C_i is the longest message duration of stream S_i and T_i is the periodicity of its requests. In order to have a timing analysis independent from the model of the tasks, it is assumed that this periodicity is the minimum time interval between two consecutive requests arrival to the outgoing queue. Finally, D_i is the relative deadline of a message; that is, the maximum time interval between the instant when the message request is placed in the outgoing queue and the instant when the message is completely transmitted.

3.2. Failure Assumptions

In the assumed network model, temporary failures are a consequence of either bus errors or network interface (transceiver) errors. Such network failures have the following semantics:

- Bus error bursts never affect more than n_{bus} transmissions during an interval of analysis T_{bus}. This means that, even for the case of multiple sources of errors, the time interval during which the network is inaccessible is upper-bounded.
- Transceivers either behave correctly or crash after a given number of failures (n_{transc}), during the interval of analysis T_{transc}. This behaviour is guaranteed by the CAN protocol, since in the case of multiple errors, the station goes first into the *Error-Passive* state and then into the *Bus-Off* state.

It is also assumed that there is no permanent network failure, such as network partitions.

4. Analysis of Previous Relevant Work

The use of CAN networks to support dependable real-time applications requires not only time-bounded transmission services, but also a minimum level of confidence on the continuity of service. This Section presents some of the most relevant results concerning both the study of the inaccessibility characteristics and the analysis of message's response time in CAN networks.

4.1. Inaccessibility Analysis of CAN networks

4.1.1. Inaccessibility Due to Bus Errors. Considering the available error detection and signalling mechanisms, it follows that bit corruption errors can be detected by several of the CAN error detection mechanisms. From all these errors, the longest network inaccessibility [1] results from a Form Error detected at the end of the EOF delimiter. Such network inaccessibility is:

$$t_{ina} = C_{MAX} + C_{error} + C_{IFS} \quad (2)$$

where C_{error} and C_{IFS} are the duration of an Error Frame and the Inter-Frame Spacing (two consecutive frames must be separated by at least 3 recessive bits), respectively, and C_{MAX} is the longest duration of a CAN message.

In the presence of multiple bus errors, two different scenarios can be considered:

- A burst of successive bit errors, where only the first one corresponds to a bit corruption in a Data Frame. The others will just disturb Error Frames being transmitted in response to the first error.
- A longer network inaccessibility results from considering that bus errors are sufficiently apart to interfere with *n* Data Frames. This results in *n* failed attempts to transmit a Data Frame.

The network inaccessibility resulting from this second scenario is:

$$t_{n\_ina} = n \times (C_{MAX} + C_{error} + C_{IFS}) \quad (3)$$

4.1.2. Inaccessibility due to erratic transceivers. Apart from the frame error detection mechanisms, CAN controllers have two error counters to isolate erratic transceivers, preventing them from interfering with the normal bus operation (see Section 2.2). The values of these counters are increased or decreased (at different rates) as a function of the detected error.

In the case of an erratic transmitter, the maximum number of transmission errors (leading to the transmission of Active Error Frames) is given by:

$$n_{tx} = \left\lceil \frac{ect}{\Delta_{tx}} \right\rceil \quad (4)$$

where *ect* is the error count threshold, and D_{tx} is the increase of the counter at each detected transmission error. As $ect=127$ and $D_{tx}=8$, then 16 consecutive active Error Frames will be transmitted before a failed transmitter enters into the Error-Passive state.

For the case of a receiver, the maximum number of receiving errors (leading to the transmission of active Error Frames) is given by:

$$n_{rx} = \left\lceil \frac{ect}{\Delta_{rx1} + \Delta_{rx2}} \right\rceil \quad (5)$$

where D_{rx1} and D_{rx2} are used according to the detected error [2]. As $D_{rx1}=8$ and $D_{rx2}=1$, then 15 Active Error Frames will be transmitted before a failed receiver enters into the Error-Passive state.

Therefore, the time interval during which an erratic transceiver can interfere with the normal behaviour of the network is upper-bounded. For instance, an erratic transceiver will only stop transmitting Active Error Frames when its error count reaches the Error-Passive threshold. Hence, it can cause up to 16 failed transmissions in the network.

4.2. Response Time Analysis of CAN Networks

In [5] the authors address in detail the response time analysis of CAN networks. They assume fixed priorities

for message streams (since the network access is based on the identifier's priority) and a non-preemptive scheduling model (since lower priority messages being transmitted cannot be preempted by pending higher priority messages). Considering such scheduling model, the existing schedulability analysis [6] is adapted to the case of scheduling messages on a CAN network.

The worst-case response time of a queued message, measured from the arrival of the message request to its complete transmission, is:

$$R_m = I_m + C_m \tag{6}$$

The message stream set schedulability is guaranteed if every message has a response time smaller than its deadline. The term I_m represents the worst-case queuing delay - longest time interval between the arrival of the message request and the start of its transmission.

The deadline monotonic (DM) priority assignment [6] can be directly implemented in a CAN network, by setting the identifier field of each message stream according to the DM rule. Therefore, the worst-case queuing delay of message *m* is:

$$I_m = B_m + \sum_{\forall j \in hp(m)} \left(\left\lceil \frac{I_m + t_{bit}}{T_j} \right\rceil \times C_j \right) \tag{7}$$

where B_m is the worst-case blocking factor, which is equal to the longest duration of a lower priority message:

$$B_m = \max_{\forall k \in lp(m)} \{0, C_k\} \tag{8}$$

$lp(m)$ is the set of message streams with lower-priority than message stream S_m. t_{bit} is the duration of a bit transmission and $hp(m)$ is the set of message streams with higher-priority than S_m. Equation (7) embodies a mutual dependency, since I_m appears in both sides of the equation. The easiest way to solve such equation is to form a recurrent relationship [6].

4.3. Network load analysis in CAN

The computation of the network load is a single measurement based on the characteristics of the message streams. Such network load can be evaluated as follows:

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \tag{9}$$

5. Integration of Inaccessibility Issues with Response Time Analysis

In this Section the response time analysis of CAN networks is extended to integrate temporary periods of network inaccessibility. Essentially, formulae are provided to evaluate both the response time of messages and the resulting network load, considering a realistic assumption of a communication network disturbed by temporary periods of inaccessibility.

5.1. Evaluation of a CAN Message Duration

The duration of a CAN message can be evaluated considering that for each Data Frame there is a Data Field added to 44 bits of overhead (64 bits of overhead in extended frames). Additionally, it must be considered the overhead concerning bit stuffing and Inter-Frame Spacing (refer to Section 2).

Bit stuffing mechanisms are applied to the first 98 bits of the frame (it excludes the CRC delimiter, ACK and EOF fields), considering an 8 byte Data Field. In the worst case, bit stuffing increase the frame by $\lfloor 98/5 \rfloor = 19$ bits (23 bits in extended frames), which means an overhead of 63 bits (87 bits in extended frames), that is approximately 50% of the frame (58% in extended frames).

A Remote Frame is similar to a Data Frame, without the Data Field. Therefore, its maximum size is 44 bits (64 bits in extended frames). As it is also coded by the method of bit stuffing, its size can be increased to 50 bits (74 bits in extended frames).

Additionally, the minimum Inter-Frame Spacing (IFS), which is 3 bits long, must be considered as a time interval during which the bus is not available for further transmissions. Also, if there is a slow controller on the bus, it may request extra time between frames, in order to process the received frame. In such case, the controller is allowed to send two consecutive overload frames, preventing other stations from transmitting further frames. An Overload Frame has the same structure of an Error Frame (Section 2.2), and thus it means that with a slow controller on the bus, there is an extra overhead of 40 bits that must be considered for every message.

5.2. Response Time Analysis Considering Network Inaccessibility

In order to integrate the inaccessibility analysis presented in Section 4.1 in the response time analysis of CAN message streams, two factors must be added to equations (6) and (7) to account for the maximum inaccessibility time from bus (Ina_{bus}) and transceiver (Ina_{transc}) errors:

$$R_m = I_m + C_m \tag{10}$$

$$I_m = B_m + \sum_{\forall j \in hp(m)} \left(\left\lceil \frac{I_m + t_{bit}}{T_j} \right\rceil \times C_j \right) + Ina_{bus} + Ina_{transc} \tag{11}$$

The maximum number of errors (n_{errors}) that can interfere with the transmission of message *m* (considering the existence of *n* errors in a period *T*) is given by:

$$n_{errors} = n \times \left\lceil \frac{I_m + C_m}{T} \right\rceil \tag{12}$$

Hence, according to the considered failure assumptions (a maximum of n_{bus} errors during a time

interval T_{bus}), the network inaccessibility due to bus errors is:

$$Ina_{bus} = n_{bus} \times \left\lceil \frac{I_m + C_m}{T_{bus}} \right\rceil \times t_{ina} \quad (13)$$

The maximum inaccessibility due to an erratic transceiver is a consequence of 16 consecutive errors (since, the station with an erratic transceiver will go into Error-Passive state after 16 consecutive errors). Therefore, the maximum network inaccessibility due to transceiver errors is:

$$Ina_{transc} = 16 \times t_{ina} \quad (14)$$

5.3. Network Load Analysis Considering the Network Inaccessibility

The network load is given by the sum of the ratio transmission delay versus period of all message streams. Additionally, periods of temporary network inaccessibility (due to on-going error detection and recovery mechanisms) must also be considered.

Considering the set of failure assumptions presented in Section 3.2, the network load resulting from bus errors and transceiver errors is:

$$U_{ina} = \frac{n_{bus} \times t_{ina}}{T_{bus}} + \frac{16 \times t_{ina}}{T_{transc}} \quad (15)$$

Consequently, the overall network load is:

$$U = \left(\sum_{\forall m} \frac{C_m}{T_m} \right) + U_{ina} \quad (16)$$

6. Case study (SAE Benchmark)

This Section presents the analysis of a CAN network example, where temporary periods of network inaccessibility are considered. The chosen example is based on the SAE benchmark [7], which, although specified within the context of automotive industry, is an interesting option, since it allows the comparative analysis of the proposed methodology with previously available results [5].

This SAE benchmark specifies a set of messages that must be transferred, considering network data rates of: 125 Kbit/sec, 250 Kbit/sec, 500 Kbit/sec and 1 Mbit/sec. A simplification of this benchmark for the case of CAN networks was presented in [5], where the number of message streams is drastically reduced by piggybacking groups of data messages in single Data Frames, whenever possible. This simplification allows a reduction of the overall network load, due to the removal of the messages' overhead. Table 1 presents the resulting set of message streams, ordered by decreasing priorities.

Table 1. SAE benchmark

S_i	C_i (bytes)	T_i (ms)	D_i (ms)	S_i	C_i (bytes)	T_i (ms)	D_i (ms)
A	1	1000	5	J	2	10	10
B	2	5	5	K	1	100	20
C	1	5	5	L	4	100	100
D	2	5	5	M	1	100	100
E	1	5	5	N	1	100	100
F	2	5	5	O	3	1000	1000
G	6	10	10	P	1	1000	1000
H	1	10	10	Q	1	1000	1000
I	2	10	10				

Table 2 presents the response time and the network load considering the message stream set of Table 1 (evaluated using equations (10) and (16), respectively). The *0 errors* assumption is the assumption considered in [5], although with a slight difference: in [5] the authors assume that a message could be blocked by messages with 8 data bytes, although there is no such message in the benchmark. Thus, the response times presented in the 1st column of Table 2 are slightly smaller than those presented in [5].

Table 2. Response Time of Messages (125 Kbit/sec)

Msg.	Response Time (ms)				
	0 errors	1 error	2 errors	3 errors	Transc. error
A	1.368	2.416	3.464	4.512	18.136
B	1.952	3.000	4.048	5.096	18.720
C	2.456	3.504	4.552	6.184	21.560
D	3.040	4.088	5.136	7.272	24.160
E	3.544	4.592	7.312	8.360	28.672
F	4.128	5.176	8.400	9.448	33.952
G	4.864	8.672	9.720	10.768	43.712
H	5.368	9.176	10.224	14.920	54.176
I	8.712	9.760	14.960	18.768	60.040
J	9.296	10.344	18.888	19.936	78.536
K	9.800	18.928	19.976	29.104	99.288
L	10.456	19.584	20.632	29.760	100.448
M	19.040	20.088	29.216	30.264	110.272
N	19.544	28.672	29.720	38.848	119.360
O	20.048	29.176	30.224	39.352	120.368
P	28.632	29.680	38.808	39.856	128.952
Q	28.656	29.704	38.832	39.880	128.976
U (%)	80.279	81.327	82.375	83.423	80.280

In this table, all the message streams that may miss their deadlines are highlighted. A network date rate of 125 Kbit/sec is considered (which leads to the highest network load) together with the following set of error assumptions:

- from 0 to 4 bus errors in each 100 ms time interval, resulting from a bit error rate of approximately 10^{-4} (for a data rate of 125 Kbit/sec, this results in considering 0-4 errors within 12500 bits), which is an expectable range for bit error rates in aggressive environments;
- a single transceiver failure (causing 16 failed transmissions), leading the related station to an Error-Passive state.

As it can be seen, a set of message streams that is completely schedulable without considering temporary periods of network inaccessibility, is no longer schedulable even assuming low bit error rates. The simple consideration of one bit error within an interval of 100 ms leads to a faulty timing behaviour in two of the message streams. Network load does not increase significantly since just 0-4 bus errors are considered within each interval of 100 ms.

An interesting result is that, conversely to what is common in priority driven systems, the first message stream to miss its deadline is not the lowest priority one, but one with an intermediate priority (message streams **F** and **J**). The reason for this unusual behaviour is that the occurrence of a bus error results in the same inaccessibility period, whatever the message stream being considered. Therefore, message streams with smaller response times will have the larger percentage increase on its message's duration, resulting that the most penalised message streams will be the ones with the smallest slack time (smallest difference between response time and deadline).

This unusual behaviour is present even in the case of errors during the transfer of lower priority messages. In this case, the mechanism needed to recover from the error prevents higher priority messages from being transmitted.

Thus, in the presence of bus errors, a CAN fieldbus network *is not able to provide different integrity levels* to the supported applications, since errors in low priority messages interfere with the response time of higher priority messages. This result proves that the scheduling of messages in the presence of errors *is not equivalent* to the scheduling of fixed priority systems in overload conditions (where tasks/messages with lower priorities do not interfere with the response time of higher priority tasks/messages).

Table 3. Response Time of Messages (250 Kbit/sec)

Msg.	Response Time (ms)					
	0 errors	1 error	2 errors	3 errors	4 errors	Transc. error
A	0.684	1.208	1.732	2.256	2.780	9.068
B	0.976	1.500	2.024	2.548	3.072	9.360
C	1.228	1.752	2.276	2.800	3.324	9.904
D	1.520	2.044	2.568	3.092	3.616	10.992
E	1.772	2.296	2.820	3.344	3.868	11.828
F	2.064	2.588	3.112	3.636	4.160	12.624
G	2.432	2.956	3.480	4.004	4.528	13.576
H	2.684	3.208	3.732	4.256	4.780	14.272
I	2.976	3.500	4.024	4.548	5.072	14.816
J	3.268	3.792	4.316	4.840	6.744	16.780
K	3.520	4.044	4.568	5.092	6.996	17.324
L	3.848	4.372	4.896	6.800	7.324	17.652
M	4.100	4.624	6.528	7.052	7.576	17.904
N	4.352	4.876	6.780	7.304	7.828	18.156
O	4.604	5.128	7.032	7.556	8.080	18.408
P	4.856	6.760	7.284	7.808	8.332	18.660
Q	4.868	6.772	7.296	7.820	8.344	18.672
U (%)	40.140	40.664	41.188	41.712	42.236	40.140

Table 3 analyses the same scenario for the case of a network data rate of 250 Kbit/sec. Obviously, as the duration of messages is reduced by 50%, the overall network load is also reduced by 50%. As a consequence, considering such reduced network load for this particular set of message streams (with harmonic periodicities), the message stream set is now schedulable for the considered failure assumptions.

Also included in Tables 2 and 3 is the consideration of a single transceiver failure. In this situation, higher priority messages miss their deadlines. It is interesting to notice that the response time of message stream **A** increases 13 times when a transceiver error is considered, but the network load does not suffer any increase. That is due to the assumption of an extremely low failure rate for transceivers, leading to a negligible increase in the network load.

It is also clear that transceiver errors are extremely penalising for the scheduling of message stream sets (14), since a station with an erratic transceiver may signal up to 16 errors, preventing other stations from accessing the bus.

Finally, Table 4 analyses a scenario where there are no bus errors; instead, a single transceiver error for different network date rates is considered. It can be seen that, even without bus errors, the message stream set is only schedulable at 1Mbit/sec, that is, it is only schedulable for a network load as low as 10%.

Table 4. Response Time of Messages considering one transceiver error

Msg.	Response Time (ms)			
	1 Mbit/sec	500 Kbit/sec	250 Kbit/sec	125 Kbit/sec
A	2.267	4.534	9.068	18.136
B	2.340	4.680	9.360	18.720
C	2.403	4.806	9.904	21.560
D	2.476	4.952	10.992	24.160
E	2.539	5.496	11.828	28.672
F	2.612	5.768	12.624	33.952
G	2.704	6.098	13.576	43.712
H	2.767	6.224	14.272	54.176
I	2.840	6.370	14.816	60.040
J	2.913	6.516	16.780	78.536
K	2.976	6.642	17.324	99.288
L	3.058	6.806	17.652	100.448
M	3.121	6.932	17.904	110.272
N	3.184	7.058	18.156	119.360
O	3.247	7.184	18.408	120.368
P	3.310	7.310	18.660	128.952
Q	3.313	7.316	18.672	128.976
U (%)	10.035	20.070	40.140	80.280

7. Pessimism Analysis

Up to this moment, a set of worst case error assumptions has been assumed. It is, therefore, important to evaluate how severe is the pessimism inherent to the proposed approach. Considering the proposed worst-case

analysis (equations (13) and (15)), some sources of inaccessibility-related pessimism can be identified:

- It has been assumed that the worst case error assumptions are always present. That is, that all the n_{bus} and n_{transc} are present in one round of messages;
- It has been assumed that bus errors are always detected in the last bit of the longest Data Frame;
- It has been also assumed that an Error Frame has always the maximum number of bits.

Although this set of assumptions is necessary for the worst case evaluations, it is also correct to say that it contains an important level of pessimism. In order to assess the impact of each one of these factors in the pessimism of the response time analysis, the following set of equations has been used:

$$Ina_{bus} = \mathbf{A} \times n_{bus} \times \left\lceil \frac{t_m + C_m}{T_{bus}} \right\rceil \times \times \left(\mathbf{B} \times C_{MAX} + (0.7 + 0.3 \times \mathbf{C}) \times C_{error} + C_{IFS}\right) \quad (17)$$

$$U_{ina\_bus} = \mathbf{A} \times n_{errors} \times \times \frac{\left(\mathbf{B} \times C_{MAX} + (0.7 + 0.3 \times \mathbf{C}) \times C_{error} + C_{IFS}\right)}{T_{bus}} \quad (18)$$

where **A** stands for the percentage of assumed errors in a period of T_{bus} (maximum of 4 errors), **B** stands for the percentage of the longest message to be transmitted and **C** is the percentage of the error frame length. As Error Frames have at least 14 bits, **C** can only be applied to the remaining 6 bits.

Fig. 2 and 3 illustrate the impact of each one of these factors on the network load and on the response time of Message Stream **F** (Message Stream **F** is chosen for the analysis, since it is the one with the smallest slack time). The variation of parameter **A** is made considering a value of 1 for parameters **B** and **C**. Variation of parameters **B** and **C** is made considering the existence of 3 bus errors.

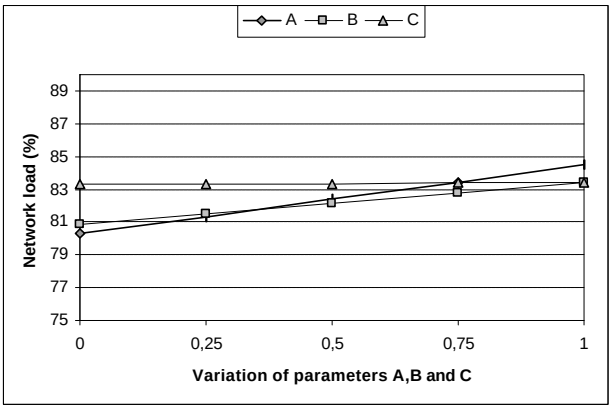

Fig. 2. Variation of the network load with parameters A, B and C

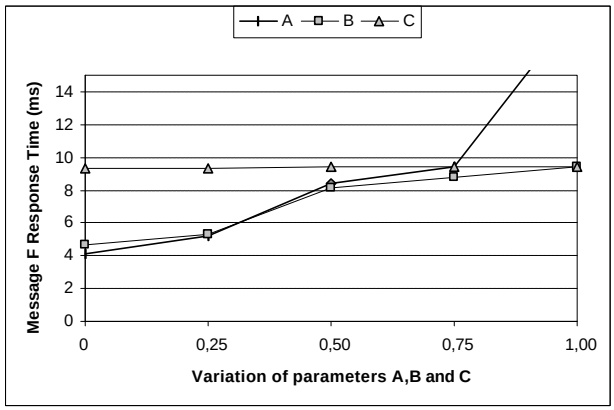

Fig. 3. Variation of message stream F response time with parameters A, B and C

As it can be seen, the parameter that has the strongest influence is the bus error rate. However, network load is only slightly penalised by errors. That is due to the assumption of a low failure rate in the network, since just 0-4 bus errors are considered within each interval of 100 ms.

The analysis presented in Section 6 showed that message stream **F** is only schedulable in the absence of errors (Table 2). In Fig. 3, such non-schedulability is reflected in the sudden increase of its response time, which is due to the increasing interference of higher-priority message streams (with 5 ms period). As shown in Fig. 3, the response time of this message stream is highly dependent on the assumed error rate, and also on the assumed inaccessibility time caused by such errors. However, with smaller periods of temporary inaccessibility, the message stream is schedulable even for larger error rates.

In order to assess the pessimism of considering that errors always occur in the last bit of the largest message, Fig. 4 shows the impact of parameter **B** for different bus errors assumptions (parameter **A**).

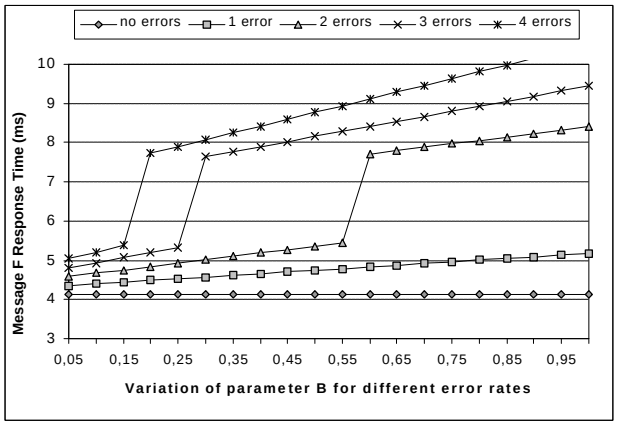

Fig. 4. Variation of message stream F response time with parameter B

Considering just one bus error per 100 ms, when parameter **B** is set to 0.5, the response time of message stream **F** will be just 4.744 ms, which compared to 5.176 ms (Table 2) gives a reduction of 8%. Furthermore, for this scenario, message stream **F** becomes schedulable. If greater error rates are assumed, the decrease of the response time is even more relevant.

The scenario is quite realistic since there is only one message that takes 6 bytes of data, while the majority of the messages have 1 or 2 bytes of data. Therefore, the inherent pessimism of worst case analysis can be reduced, if relaxed failure assumptions are accepted.

Conclusions

This paper addresses the response time analysis of CAN messages, considering temporary periods of network inaccessibility. It extends previous response time analysis, providing more accurate results on the timing behaviour of CAN networks. A benchmark was used to illustrate the relevance of the proposed analysis and also to evaluate its inherent pessimism.

From the achieved results, it can be concluded that message streams with smaller response times will have the larger relative increase on its duration, resulting that the most penalised message streams will be the ones with the smallest slack time.

An important conclusion is also that a CAN fieldbus network *is not able to provide different integrity levels* to the supported applications, since errors in low priority messages interfere with the response time of higher priority messages. This result proves that the scheduling of messages in the presence of errors *is not equivalent* to the scheduling of fixed priority systems in overload conditions (where tasks/messages with lower priorities do not interfere with the response time of higher priority tasks/messages). Another conclusion is that CAN is not resilient to transceiver errors, since they can lead to large inaccessibility periods.

The inherent pessimism of the proposed analysis has also been evaluated, and it is concluded that the message set response times' are highly dependent on the considered error rates and inaccessibility periods. It is also concluded that assuming smaller periods of temporary network inaccessibility, the system becomes schedulable even for greater bus error rates. This assumption is quite realistic, since the majority of the considered messages carry only 1 or 2 bytes of data.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful comments. This work was partially supported by FLAD (project SISTER 471/97), FCT (project DEAR-COTS 14187/98) and IDMEC.

References

[1] J. Rufino and P. Veríssimo, "A Study on the Inaccessibility Characteristics of the Controller Area Network", in *Proc. of the 2nd International CAN Conference*, London, United Kingdom, October 1995.

[2] ISO 11898, "Road Vehicle - Interchange of Digital Information - Controller Area Network (CAN) for High-Speed Communication", ISO, 1993.

[3] K. Zuberi and K. Shin, "Scheduling messages on Controller Area Network for Real-Time CIM Applications", in *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, pp 310-314, 1997.

[4] Rockwell Automation., "DeviceNet Product Overview", *Publication DN-2.5*, Rockwell, 1997.

[5] K. Tindell, A. Burns and A. Wellings, "Calculating Controller Area Network (CAN) Message Response Time", in *Control Engineering Practice*, Vol. 3, No. 8, pp. 1163-1169, 1995.

[6] N. Audsley, A. Burns, M. Richardson, K. Tindell and A. Wellings, "Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling", in *Software Engineering Journal*, Vol. 8, No. 5, pp. 285-292, 1993.

[7] SAE, "Class C Application Requirement Considerations", *Technical Report J2056/1*, SAE, 1993.