Worst-case deadline failure probability in real-time applications distributed over CAN (Controller Area Network)

N. Navet[1] Y.-Q. Song[1]
[1] LORIA - INPL
ENSEM
2, Avenue de la forêt de Haye
54516 Vandoeuvre-lès-Nancy
France

F. Simonot[2]
[2] Institut E. Cartan
ESSTIN
Parc R. Bentz
F-54500 Vandoeuvre-lès-Nancy
France

March 16, 1999

Abstract

Real-time applications distributed over the CAN network are generally characterised by stringent temporal and dependability constraints. Our goal is to take account of transmission errors in the design of such applications because the consequences of such disturbances are potentially disastrous. In this study, the concept of worst-case deadline failure probability (WCDFP) is introduced. The motivation of the probabilistic approach is that, in practice, the number of errors occurring during a given time period can with difficulty be bounded. To evaluate the WCDFP, we propose, on the one hand, a method of computing for each message the tolerable threshold of transmission errors under which timing constraints are guaranteed to be met. On the other hand, we also suggest an error model enabling us to consider both error frequency and error gravity. Our error model follows a generalized Poisson process and its stochastic parameters have been derived. We then propose a numerically efficient algorithm to compute the probabilities and apply the analysis to an industrial case-study of the automotive field.

keywords : CAN, Embedded Systems, Error Model, Fault Tolerance, Real-Time Systems.

1 Introduction

Real-Time Distributed Applications (DRTA, e.g. embedded applications, process control) increasingly use the CAN network for transmitting real-time information between sensors, actuators and control devices (computers, PLC, ...).

Such applications are generally characterised by the obligation to respect stringent temporal and dependability constraints and consequently a significant part of the design process of such systems is devoted to the verification of the constraints respect.
It appears essential to us to seriously take account of transmission errors in the validation of DRTA using CAN, considering on the one hand the existence of such disturbances (mainly caused by electromagnetic fields) and, on the other hand, the potentially disastrous consequences of failing to respect the time constraints (i.e. the transmitted data in a vehicle frequently represent information vital to the safety of the passengers). The *Wall Street Journal* (edited 8 September 1997 - quoted in [13]) and the *NRC Handelsblad* (edited 25 and 26 September 1997 quoted in [14]), talked about accidents involving vehicles, which to all appearances were caused by electromagnetic disturbances. The criticality of the constraints led us to assess the 'worst-case' performance.
Assuming that only periodic or sporadic messages have stringent timing requirements, Tindell et al [16] have proposed a solution for calculating an upper bound on frame response times on CAN. Tindell and Burns have extended their previous works in [17, 15] to transmission errors by considering a deterministic error model with which they can actually calculate an upper limit to the response time. Their error model makes the assumption that the number of errors can be upper bounded during a given time period. However, in our opinion, this is in general not possible because transmission errors are a random phenomenon and in essence, a random phenomenon tends to better obey probabilistic laws rather than deterministic ones.
Our goal is to take into account the transmission errors in the design of DRTA but we will focus only on the effect of EMI (electromagnetic interferences) on the data transmission because the transmission support is a particularly "weak link" although EMI could also affect the correct functioning of all the electronic devices in the vehicle. For this purpose, the concept of Worst-Case Deadline Failure Probability (WCDFP) is defined and a flexible error model making assumptions on both error frequency and gravity is proposed. The occurrence of error obeys a Poisson Process. When an error occurs, it can be either a single error or a burst of errors. For instance, for in-vehicle networks, single and burst errors respectively correspond to a normal and to a strongly disturbed area (such as the surroundings of an airport that are scanned by radars). In the context of process control applications, burst errors can also occur for instance when surrounding machines are switched-on/off creating electromagnetic disturbances.
In section 2, we describe the CAN protocol and emphasising its way of handling transmission errors. Section 3 is devoted to the calculus of the worst-case response time and maximum error threshold. Section 4 describes the error model and gives the WCDFP definition and as well as an efficient computing method. Finally, in section 5, our analysis is applied to an industrial case-study.

2 The CAN network

Electronic control units began to equip vehicles in the early eighties. This has increased the need for real-time communication within a vehicle and the use of more dedicated signal lines was impossible because of cost, reliability and repair problems. To fulfil this need of multiplexed communication, "Robert Bosch Gmbh" designed the CAN network. Although specifically conceived for the automotive industry, CAN is also widely used in automation and this, mainly due to the low price of CAN communication solutions. The reader interested in CAN could consult the standard [7] and [8] for a good starting point.
CAN is a broadcast bus, with a priority-based access to the medium and non-destructive collision resolution. Nodes do not possess an address and none of them plays a preponderant role in the protocol. A message contains an identifier (Id), unique to the whole system, that serves two purposes : assigning a priority for the transmission (the lower the numerical value of the identifier, the greater the priority during the arbitration phase) and allowing message filtering upon reception. Data, possibly segmented in several frames, may be transmitted periodically, sporadically or on-demand ("Remote Transmission Request").
CAN has very efficient error detection mechanisms. In [18] Unruh et al. have estimated the expected number of undetected transmission errors during the lifetime of a vehicle to be lower than 10^{-12}, that is why we will further assume that all errors are correctly detected. Each node which detects an error sends an "error flag" (6 consecutive dominant bits) and the corrupted frame will automatically re-enter arbitration which can lead it to miss its deadline. The error recovery time (time from detecting an error until the possible start of a new frame) is 17 to 31 bit times. In the rest of the paper, an error recovery time of 23 bits will be considered as the maximum overhead because it will be further assumed that all stations stay in the "error active" state (see [7] for more details) and 23 bits is the maximum overhead with an "error active" transmitting node but the analysis remains valid whatever the value chosen. Note that any higher priority frame that became ready since the original frame won arbitration will gain the bus, extending thus the response time of the original frame by more time than the error recovery time.
The ISO standards [7] [6] specify a two-wire differential bus as in-vehicle transmission support with a data rate up to 1 Mbit/s. Under the same EMI conditions, the frequency of transmission errors depends on the bus transmission rate and on its electrical characteristics but it also greatly depends on the transmission support : for instance, measures published in [1] have shown the untwisted/unshielded pair to be 6 times more sensitive than the twisted/shielded pair. The use of all-optical network, which offers very high immunity to EMI, is not feasible yet because of low-cost requirement imposed by the automotive industry.
Due to the media access technique (priority based Carrier Sense Multiple Access/Collision Resolution), the maximum data rate which can be achieved in CAN networks is essentially dependent on the bus length, for example the maximum rate for 30 and 500 meters are respectively 1Mbit/s and 125kbit/s. CAN

uses Non-Return-to-Zero (NRZ) bit representation with a bit stuffing of length 5. Bit stuffing is an encoding method that enables resynchronisation when using Non-Return-to-Zero (NRZ) bit representation. Edges are artificially inserted into the outgoing bit stream by prohibiting the transmission of more than a maximum number (the length of the stuff) of consecutive equal level bit. The receiver will apply the inverse procedure and de-stuff the frame.

3 Worst-case response time and error threshold

A periodic frame m is characterised by (C_m, T_m, J_m, D_m) where T_m is the period, C_m the transmission time, D_m the deadline and J_m the maximum "jitter" (the variability in queueing times derived from the response time of the sending task). Note that sporadic messages can also be considered by taking the minimum interarrival time as the period.

3.1 Existing work

3.1.1 Worst-case response time with a reliable medium

The worst-case response time R_m, which is defined as the longest time between the start of the task queuing m and the latest time that the message arrives to the destination processor(s), must be bounded for each frame by D_m otherwise the timing constraint can not be guaranteed and the set of messages of the application is said to be non-schedulable. To calculate R_m, as the transmission time C_m and the jitter J_m can be upper bounded, we just have to compute the maximum time needed by the message to gain the arbitration (termed the "interference" time). A message m can be delayed by higher priority messages and by a lower priority message that has already obtained the bus (this time denoted as B_m is the transmission time of the biggest lower priority message). Thus, we have [16] :

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

where I_m is the interference time, i.e. the longest time that all higher priority messages can occupy the bus plus B_m :

$$I_m^n = B_m + \sum_{\forall j \in hp(m)} \left\lceil \frac{I_m^{n-1} + J_j + \tau_{bit}}{T_j} \right\rceil C_j \tag{2}$$

with τ_{bit} the bit time, $hp(m)$ the set of messages of higher priority than m and C_j the transmission time of a message j with d_j data bytes :

$$C_j = \left(47 + 8d_j + \frac{34 + 8d_j}{4}\right) \tau_{bit} \tag{3}$$

where 47 is the size of the fixed-form bit fields of the CAN frame and $\lfloor (34 + 8d_j)/4 \rfloor$ is the maximum number of "stuff" bits. I_m is computed starting with $I_m^0 = 0$ until convergence. The reader could refer to [16, 15, 17] for more details.

3.1.2 Worst-case response time with transmission errors

Tindell and Burns, in [15], have introduced errors by considering the following error model :

- during a time t, we have exactly one burst of errors whose size is n_{error}.
- Except for this burst of errors, the error period is T_{error}

The number of transmission errors during time t is thus : $\left(n_{error} + \left\lceil \frac{t}{T_{error}} \right\rceil - 1\right)$.
The response time for frame m with the above error model is obtained by changing equation 2 as follows :

$$I_m^n = E_m(I_m^{n-1} + C_m) + B + \sum_{\forall j \in hp(m)} \left\lceil \frac{I_m^{n-1} + J_j + \tau_{bit}}{T_j} \right\rceil C_j \qquad (4)$$

with $E_m(t)$ the error recovery overhead function which gives an upper limit to the overheads due to error recovery that could occur in an interval of duration t :

$$E_m(t) = \left(n_{error} + \left\lceil \frac{t}{T_{error}} \right\rceil - 1\right) \cdot \left(23\tau_{bit} + max_{\forall j \in hp(m) \cup \{m\}} C_j\right) \qquad (5)$$

3.2 Tolerable errors threshold

We propose a different approach which is to determine for each frame m, the maximum number of errors, denoted K_m, for which $R_m \leq D_m$ is true. $E_m()$, the error recovery overhead function which was introduced by Tindell as a function of time becomes here a function of the number of errors n :

$$E_m(n) = n \cdot (23\tau_{bit} + max_{\forall j \in hp(m) \cup \{m\}} C_j) \qquad (6)$$

The calculus of the response time for frame m with K_m errors, denoted by $R_{m_{max}}$, is performed iteratively, starting with $K_m = 0$ and incrementing this at the end of each step. This continues until equation 4, with $E_m()$ defined by equation 6, does not converge or until $R_m > D_m$. In order not to take account of the last unsuccessful iteration, we have then to subtract 1 from K_m.
K_m is needed to calculate the WCDFP of frame m (see paragraph 4.3). This will enable us to estimate the reliability of our application in an electromagnetic stressed environment and to optimise our set of messages by changing their priorities and periods.

4 Towards a realistic error model

From observations carried out on board some vehicles, two types of errors are identified : single error (the vehicle is found in a zone that is not particularly disturbed) and burst errors (see figure 1). The latter type of error corresponds to strongly disturbed zones such as areas that are scanned by radars near an

airport [11] [19] or the surroundings of high tension electrical lines and in this case, the size of the burst directly depends on the time needed to cross the zone. In fact, strong disturbance appears as the inaccessibility of the medium during a time period, leading to systematic transmission errors followed by the corresponding retransmissions. It can also be envisaged that this model can be applied to process control applications for which burst errors correspond to the switching-on/off of surrounding machines. To fit the reality of transmission errors as well as possible, we propose a model shown in figure 1 which takes into consideration both the frequency of the errors (error occurrences follow a Poisson law) and their gravity (burst of errors or single errors) :

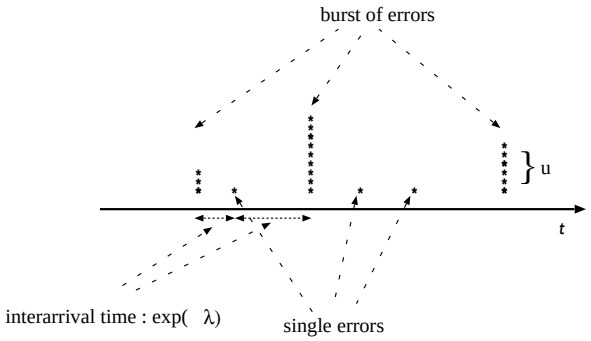

Figure 1: Error model

4.1 Model description

From a mathematical point of view, the above model can be seen as a Generalized Poisson Process (GPP, see [12]). Let $X(t)$ be the number of errors within the interval $[0, t]$, we have :

$$X(t) = \sum_{i=0}^{N(t)} y_i \qquad \text{with } y_0 = 0 \tag{7}$$

where:

- $N(t)_{t\geq 0}$ obeys the Poisson law with parameter λ with $P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$
- $y_i = \begin{cases} u, \text{ with probability} & \alpha \\ 1, \text{ with probability} & 1-\alpha \end{cases}$

u is the discrete random variable (r.v.) giving the burst size and y_i the r.v. giving the error size (single error and burst errors). The probability distribution

of u and the value of parameters α and λ should be defined according to the knowledge of the environment in which the embedded system is used. This should be done during the design phase of the vehicle with measurements carried out on board a prototype. Note that some new CAN controllers possess features that facilitate this task such as readable error counters (NEC controllers, see [5]) or better, thanks to the possibility to trigger an interrupt when an error occurs (Philips controllers, see [4]). This could even help us to conceive on-line adaptive scheduling politics (see [10] for a first step in that direction).

4.2 Model characterisation

In this section, we first give some results concerning the two r.v u and y_i, and then the stochastic characteristics of $X(t)$.

4.2.1 Characteristics of u

In this study, we will consider that the burst size follows the r.v u whose distribution $P(u)$ is given by equation 8 with parameter p ($q = 1 - p$).

$$P(u = k) = kp^2q^{k-1} \tag{8}$$

We constructed this new law based on the geometric distribution in order to obtain firstly $P(u = 0) = 0$ (the burst size never being nil) and secondly a relatively important queue distribution when $k \to \infty$. It can been shown that the expected value of u is $E(u) = (2p^{-1} - 1)$ and the variance of u is $Var(u) = \frac{2q}{p^2}$. Note that the analysis developed in this paper remains valid whatever the distribution of u may be.

4.2.2 Characteristics of y_i

As $\forall i \in \mathbb{N}$, y_i follows the same probability law, for the sake of simplicity we can ignore the index i. From the definition of the r.v. y_i which gives the error size, either a single error or a burst of errors, we know that :

$$\begin{aligned} P(y = k) &= P[y = k \mid u]\, P[u] + P[y = k \mid 1]\, P[1] \\ &= \alpha k p^2 q^{k-1} + P[y = k \mid 1]\,(1 - \alpha) \end{aligned}$$

Since $P[y = k \mid 1] = \begin{cases} 0 & k \geq 2 \\ 1 & k = 1 \end{cases}$ we obtain

$$P(y_i = k) = \begin{cases} \alpha k p^2 q^{k-1} & k \geq 2 \\ 1 - \alpha + \alpha p^2 & k = 1 \\ 0 & k = 0 \end{cases} \tag{9}$$

Other results for the r.v. y are :

- The generating function (g.f. for short) $y(z)$ of the r.v. y is :

$$y(z) \doteq E[z^y] = (1 - \alpha)z + \frac{\alpha p^2 z}{(1 - qz)^2} \tag{10}$$

- The expected value of y is $E[y] = 1 + \frac{2\alpha q}{p}$ and the variance of y is $Var(y) = \frac{2\alpha q\,(p + 3q - 2\alpha q)}{p^2}$

4.2.3 Characteristics of $X(t)$

$X(t)$ being a generalized Poisson process has then a g.f. of the form [12] $X(z) = E[z^{y(z)-1}]$, with $y(z)$ as the g.f. of y (see equation 10). We have then :

$$X(z) \doteq \sum_{k=0}^{\infty} z^k P[X(t) = k] = e^{\lambda t\left[(1-\alpha)\,z + \frac{\alpha p^2 z}{(1-qz)^2} - 1\right]} \tag{11}$$

The expected value of $X(t)$ is $E[X(t)] = \lambda t\, E[y]$ and its variance is $Var[X(t)] = \frac{\lambda t}{p^2}\left(1 + (6\alpha - 2)\,q + q^2\right)$.

4.3 WCDFP with efficient calculus method

In this paragraph, we first give the definition of the WCDFP, then a general calculus method to compute this probability. By a general method, we mean that the procedure we proposed can be applied whatever the distribution of u may be and moreover with good numerical accuracy.
Recalling to mind that K_m is the maximum tolerable number of transmission errors for frame m during $R_{m_{max}} \leq D_m$ (with $R_{m_{max}}$ being the response time with K_m errors). If less than K_m errors occur during $R_{m_{max}}$, the frame m will meet its deadline, otherwise it could miss it. The WCDFP is the probability that more than K_m errors occur during time $R_{m_{max}}$ (see equation 12). It is termed worst-case because the underlying assumption is that the time needed by the frame m to gain the bus is the maximum possible (see equation 4) and that each error is detected at the last bit of the transmission and introduces the longest error recovery time (see equation 6).

$$P[X(R_{m_{max}}) > K_m] = 1 - \sum_{k=0}^{K_m} P[X(R_{m_{max}}) = k] \tag{12}$$

$X(t)$ being a generalized Poisson process has then a generating function (g.f) of the form [12] $X(z) = E[z^{y(z)-1}]$. As $X(t)$ only takes integer values, the probability for $X(t)$ taking k can be obtained by differentiating its g.f :

$$P[X(t) = k] \doteq \frac{1}{k!}\frac{d^k X(z)}{dz^k}|_{z=0} \tag{13}$$

$P[X(t) = k]$ can be calculated with equation 13 using some widely-used symbolic calculus software like Maple. But in practice, we are faced with the numerical calculus accuracy problem. For instance, in our experiments with Maple and the default precision (10 digits), for big values of k ($k \geq 40$), if we add the

first k probabilities, we may get a value greater than 1. If we increase the precision, the computation time becomes excessive. These difficulties led us to conceive of another calculus method. Additional reasons which motivated us are the facts that the g.f of y is not always easy to obtain (depending on the burst size distribution u) and its derivative may rapidly become so big that it is impossible to handle it with symbolic calculus software. The proposed calculus method does not need the generating function of the stochastic process, it can cope with burst-size frequency histograms coming directly from measures carried out on a prototype which, from a practical point of view, is very important.
According to the definition of $X(t)$ (see equation 7), we have :

$$P[X(t)=k] = \sum_{m=0}^{k} P(X(t)=k \mid N(t)=m)\, \frac{e^{-\lambda t}(\lambda t)^m}{m!} \tag{14}$$

- if $k=0$, we have $N(t)=0$ and as $y_i \geq 1$, thus $P[X(t)=0] = e^{-\lambda t}$
- if $k \geq 1$, we have $P[X(t)=k] = \sum_{m\geq 1}^{k} P(X(t)=k \mid N(t)=m)\, \frac{e^{-\lambda t}(\lambda t)^m}{m!}$

Since $P[X(t)=k \mid N(t)=m]$ is the probability of having k errors within m package of size y_i and since $N(t)$ and the r.v. y_i are mutually independent, if we note $S_m = y_1 + y_2 + \ \ldots \ + y_m$, then :

$$P[X(t)=k] = \sum_{m=1}^{k} P[S_m=k]\, \frac{e^{-\lambda t}(\lambda t)^m}{m!} \tag{15}$$

S_m being a Markov chain, we can write :

$$P[S_{m+1}=j] = \sum_{1\leq i\leq j} P[S_{m+1} \mid S_m=i]\, P[S_m=i] \tag{16}$$

We note $a_k = P(S_1=k) = P(y_1=k) \doteq P(y=k)$, according to the *Chapman-Kolmogorov* equation [12] :

$$P[S_{m+1}=j] = \sum_{i=1}^{j} a_{j-i}\, P[S_m=i] \tag{17}$$

Noting that for $m>k$, $P[S_m=k]=0$ since $y_i \geq 1$. Equation 17 gives us the recursive algorithm for calculating $P[S_m=k]$:
Where computed[,] is a two-dimensional array, used to store already computed values, whose elements must be initialised to -1. Finally, according to equation 15 the algorithm for computing $P[X(t)=k]$ is :
with $Poisson(i,t,\lambda)$ defined as $\frac{e^{-\lambda t}(\lambda t)^i}{i!}$.

```
1  funct real P_S(integer m, integer k)
2     real r := 0
3     if (m > k)  r := 0;
4  else if (m = 1)  r := a_k;
5     else if (computed[m, k] = −1)
6             for i := 1 to k do
7                 r := r + (a_{k−i} · P_S(m − 1, i))
8             od
9             computed[m, k] := r;
10       else r := computed[m, k];
11          fi
12       fi
13    fi
14    return r;
15 end
```

Table 1: Algorithm for computing $P[S_m = k]$.

```
1  funct real compute_Xt(integer k, real t, real λ)
2     real r := 0;
3     for i := 1 to k do
4         r := r + P_S(i, k) · Poisson(i, t, λ);
5     od
6     return r;
7  end
```

Table 2: Algorithm for computing $P[X(t) = k]$.

5 Case study

We consider an experimental embedded CAN-based application, provided by PSA (Peugeot-Citröen Automobiles Company), that was implemented in a prototype car. Six devices exchange messages : the engine controller, the wheel angle sensor, the AGB (Automatic Gear Box), the ABS (Anti-Blocking System), device y[1] and the bodywork gateway. The traffic consists of a set of 12 periodic messages (e.g. speed and torque from the engine controller) listed in figure 2. The transmission rate of the CAN bus is 250kbit/s and we neglect jitters. It will be further assumed that the deadline of each frame is equal to its period. The Data Length Code (DLC) denotes the number of bytes of each frame. Equations 1, 2 and 6 enable us to compute the tolerable error threshold (K_m) and the maximum response time with K_m errors (denoted as $R_{m_{max}}$) for each message. The dependability of this system will be assessed in terms of WCDFP and we will investigate the influences of frame periods and priorities.

Priority (Id)	Transmitter node	DLC	Period	K_m	$R_{m_{max}}$
1	engine controller	8	10 ms	15	9.92 ms
2	wheel angle sensor	3	14 ms	20	13.96 ms
3	engine controller	3	20 ms	28	19.70 ms
4	AGB	2	15 ms	20	14.61 ms
5	ABS	5	20 ms	26	19.46 ms
6	ABS	5	40 ms	53	39.42 ms
7	ABS	4	15 ms	18	14.56 ms
8	bodywork gateway	5	50 ms	62	49.41 ms
9	device y	4	20 ms	23	19.54 ms
10	engine controller	7	100 ms	124	99.11 ms
11	AGB	5	50 ms	59	49.58 ms
12	ABS	1	100 ms	123	99.92 ms

Figure 2: PSA message set

From figure 2, we see that the tolerable threshold depends on both the period and the priority. With identical periods, a higher priority message can logically tolerate more errors, as illustrated by messages 3, 5 and 9. However, the period has more influence than the priority since the transmission time of a message is small ($< 0.54ms$ at 250kbit/s) compared to the period differences.

5.1 Worst Case Deadline failure probability

Applying equation 12 for each message of the system with parameters $\alpha = 0.1$, $p = 0.04$, $\lambda = 10$ and 30, gives the results illustrated in figure 3. From figure 3, we note that globally a high priority ensures a relatively low deadline failure probability. Meanwhile, the priority alone can not explain this result and it is obvious that the period greatly influences the deadline failure probability.

[1]The name of this device can not be communicated because of confidentiality.

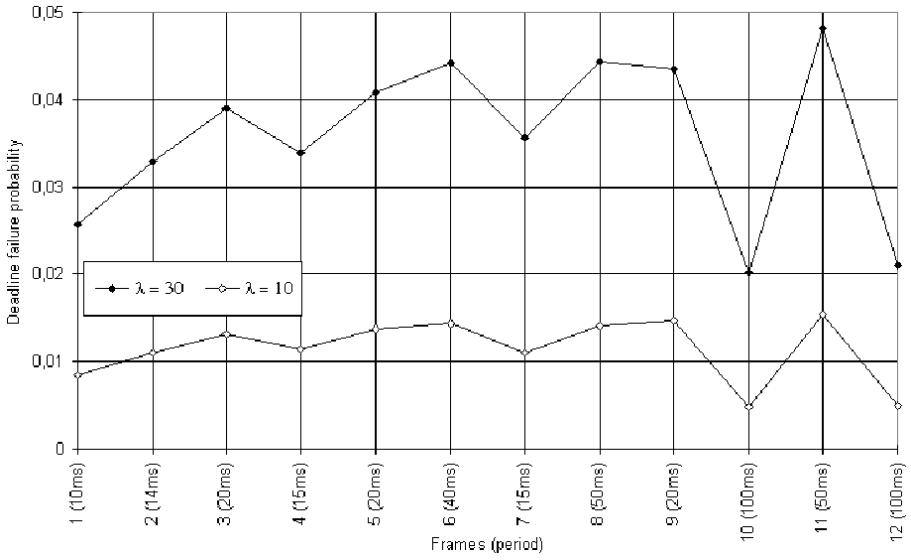

Figure 3: Worst-Case Deadline Failure Probability

Now, observing messages 4 and 7 which have much smaller periods than their neighbours, it seems that a small period increases reliability. But how can we explain so high a probability for message 11 knowing that its period is much smaller than that of its neighbours (i.e. messages 10 and 12)? Figure 4 gives us the explanation, we observe that there is a range of periods (from 20 to 50ms) within which the deadline failure probability is high. The above remarks have led us to examine in detail the influence of both period and priority on the reliability. We point out that this examination can help us to best choose periods and priorities during the design step.

5.2 Influence of period

To estimate period influence, we vary the period of the messages 6 and 11 from 10ms to 100ms and calculate the corresponding values of k and R_{imax}. The probability of the non-respect for $\lambda = 10$ and $\lambda = 30$ is shown in figure 4. From figure 4, we see that decreasing or increasing the period can considerably

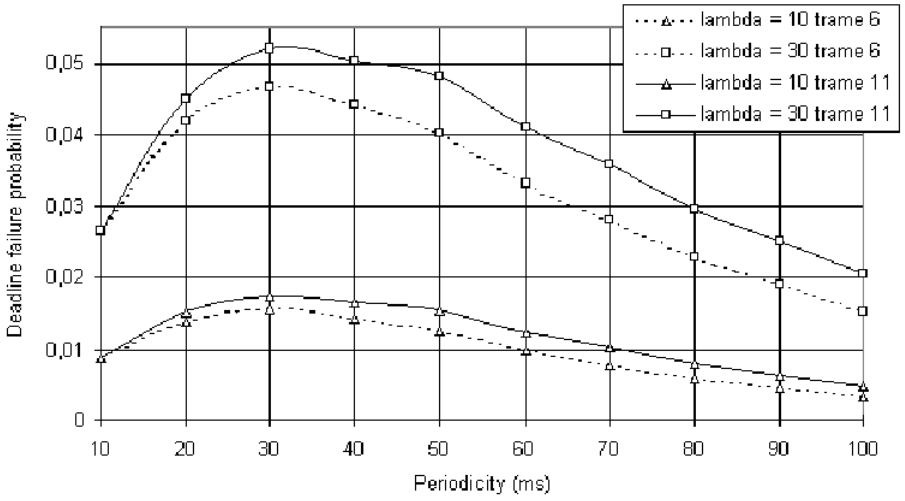

Figure 4: Influence of the period on deadline respect

reduce the deadline failure probability. Meanwhile, one must pay attention to the respect of the application constraints in order that a message source is neither under-sampling nor over-sampling, since under-sampling reduces the dependability while over-sampling generates more messages than needed and uselessly overloads the network.

5.3 Influence of priority

Taking as an example message 6, we change its priority to 1 (the highest) then to 12 (the lowest), the relative priority order of the other messages is conserved. The calculation is carried out for $\lambda = 10$ and 30 as shown in figure 5.

Logically, we confirm that the higher priority, the lower the probability of non-respect.

6 Concluding remarks

In this paper, we introduced the concept of WCDFP which conciliates worst-case response time and dependability analysis. To evaluate the WCDFP, we have proposed a method to compute the tolerable error threshold under which the

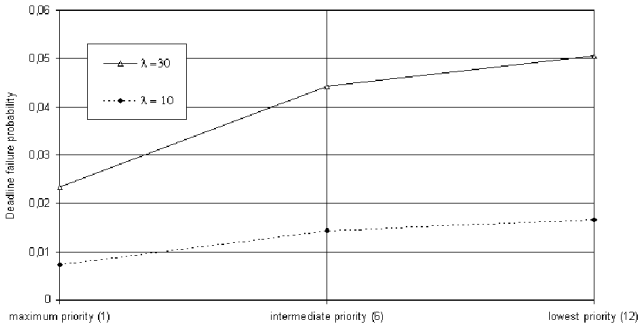

Figure 5: Influence of the priority on deadline respect

message deadlines are still met on a CAN network, and a flexible error model following a generalized Poisson process. To overcome some severe numerical computing difficulties, an efficient method for obtaining the WCDFP was also derived.
The WCDFP provides useful knowledge on the system's reliability which is considered as the main dependability indicator [20]. The WCDFP may help the designer to set the periods and priorities of messages. It can also help him to choose the right transmission support regarding dependability and cost objectives.
This work can be straightforwardly applied to other CSMA-CR based LAN such as VAN [2] or J1850 [3]. Although our study is carried out in the context of local area networking, it must be pointed out that the results could be applied to other contexts such as mobile communication systems.
Closer deadline failure estimation can be obtained by replacing the worst-case response time by its stochastic upper bound. This is our ongoing work.

Acknowledgement Part of this work was done under an industrial contract with PSA (Peugeot-Citroen) Automobiles [9]. The authors would like to thank J. Raymond for his constructive comments.

References

[1] J. Barrenscheen and G. Otte. Analysis of the physical can bus layer. In 4^{th} *international CAN Conference, ICC'97*, pages 06.02–06.08, Octobre 1997.

[2] Association Française de Normalisation (AFNOR). Véhicules routiers - transmission de données, December 1990. document R13-708.

[3] SAE Vehicle Network for Multiplexing and Data Communications Standards Committee. Class b data communications network interface. SAE J1850 Standard, Rev. NOV96.

[4] P. Hank. Pelican : A new can controller supporting diagnosis and system optimization. In *4th international CAN Conference, ICC'97*, pages 04.12–04.18, Octobre 1997.

[5] G. Hausmann and E. Gebing. The realisation of specific automotive applications with "full" can functionality at "basic" can cost on highly integrated 8-bit microcontroller of nec's 78k/0 family. In *4th international CAN Conference, ICC'97*, pages pp 4.02–4.11, 1997.

[6] International Standard Organization ISO. *Road Vehicles - Interchange of Digital Information - Controller Area Network for high-speed Communication.* ISO, 1994. ISO 11898.

[7] International Standard Organization ISO. *Road Vehicles - Low Speed serial data communication - Part 2: Low Speed Controller Area Network.* ISO, 1994. ISO 11519-2.

[8] W. Lawrenz. *CAN System Engineering.* Springer-Verlag (ISBN 0-387-94939-9), 1997.

[9] N. Navet and Y.-Q. Song. Evaluation de performances de la messagerie can du véhicule prototype psa - action 1 du contrat psa-crin. Technical report, Centre de Recherche en Informatique de Nancy (CRIN), 1996. Contract report 96-R-182.

[10] N. Navet and Y.-Q. Song. Une politique à changement de priorité pour l'ordonnancement de messages dans des environnements bruités. In *to appear in CFIP'99*, April 1999.

[11] I.E. Noble. Emc and the automotive industry. *Electronics & Communication Engineering Journal*, pages pp 263–271, October 1992.

[12] E. Parzen. *Stochastic Processes.* Holden-Day (ISBN 0-8162-6664-6), 1962.

[13] The Risks Digest. Gm car acceleration due to emi. *http://catless.ncl.ac.uk/Risks/19.38.html*, 19(38), September 1997.

[14] The Risks Digest. Mad bus disease. *http://catless.ncl.ac.uk/Risks/19.40.html*, 19(40), October 1997.

[15] K. Tindell and A. Burns. Guaranteed message latencies for distributed safety-critical hard real-time control networks. Technical report, Department of Computer Science, Univ. of York (UK), May 1994. Technical Report YCS229.

[16] K. Tindell and A. Burns. Guaranteeing message latencies on controller area network (can). In *1st International CAN Conference, ICC'94*, 1994.

[17] K. Tindell, A. Burns, and A.J. Wellings. Calculating controller area network (can) message response times. *Control Eng. Practice*, 3(8):1163–1169, 1995.

[18] J. Unruh, H.-J. Mathony, and K.-H. Kaiser. Error detection analysis of automotive communication protocols. Technical report, Robert Bosch GmbH, 1989.

[19] E. Zanoni and P. Pavan. Improving the reliability and safety of automotive electronics. *IEEE Micro*, 13(1):pp 30–48, 1993.

[20] C. Ziegler, D. Powell, and P. Desroches. Dependability of on-board automotive computer systems. In *IEEE Intelligent Vehicles 1994 Symposium*, pages pp 568–575, October 1994.

Nicolas Navet received the B.S. in Computer Science from the University of Mulhouse (France), the M.S. in Computer Science from the Technische Fachhochschule Berlin (Germany) and from the University of Nancy (France). Since 1996, he has been a PhD student at the LORIA Computer Science Laboratory in Nancy. His research interests include validation of real-time distributed applications and scheduling theory. He and his research advisor Mr Ye-Qiong Song jointly received the CAN in Automation International Users and Manufacturers Group (CiA) Research Award in 1997 for their work on CAN.

François Simonot received the B.Sc. degree in mathematics, the M.Sc. degree in statistics and the Ph.D. degree in applied mathematics from the University of Nancy 1, in 1970, 1971, 1981 respectively. Since 1974, he has been teaching pure and applied mathematics, especially probability, at University of Nancy 1 and Institut national polytechnique de Lorraine in Nancy. He has previously worked on probability metrics, robustness of stochastic systems and approximation of Markov chains. His current research include queueing systems, multiple access protocols and performance analysis of computer communication networks.

Ye-Qiong Song received the B.Sc. degree from the Beijing University of Posts and Telecommunications in 1984, the MASTERE degree from the Ecole Nationale SupĽrieure des TĽlĽcommunications de Paris in 1987, both in telecommunications engineering, the DEA degree from the University of Paris 6 in 1988, and the Ph.D. degree from the Institut National Polytechnique de Lorraine in 1991, in computer science. He has worked on the modeling and performance evaluation of FIP fieldbus using queueing theory and simulation. His current research interests include modeling and performance evaluation of local area networks and multiple access protocols when used in real-time environment. He is now an associate professor at the University of Nancy 1.