RELIABILITY IMPROVEMENT OF THE DUAL-PRIORITY PROTOCOL UNDER UNRELIABLE TRANSMISSION

N. Navet **Y.-Q. Song**
LORIA - INPL
ENSEM - 2, Avenue de la forêt de Haye
F-54516 Vandoeuvre-lès-Nancy
Tel: 33.3.83.59.57.46 *Fax: 33.3.83.44.07.43*
{*nnavet, song*}*@loria.fr*

Abstract: The dual-priority is a scheduling policy providing the guarantees needed by periodic or sporadic hard real-time tasks while decreasing the response time for aperiodic soft real-time tasks. This scheduling policy can be applied to message scheduling and its performance on CAN (Controller Area Network) will be assessed. Nevertheless, when used in an electromagnetic stressed environment (e.g. automotive communication) leading to transmission errors, this scheduling strategy could lead to serious disappointments. It will be explained why the hard real-time traffic is highly sensitive to transmission errors. The risks of deadline failure will be quantified and a simple mechanism that provides probabilistic guarantees to prevent hard real-time frames from missing their deadlines, will be proposed. This mechanism is compared in terms of performance to the original dual-priority strategy. The chosen performance metrics are the deadline failure probability for hard real-time traffic, the average response time and the variance in response time for soft real-time traffic.

Keywords: Scheduling algorithms; local area networks, fault tolerance; real-time systems.

1. INTRODUCTION

In real-time systems, one generally identifies two types of timing requirements: "hard" and "soft". It is assumed that the Hard Real-Time traffic (HRT) is periodic or sporadic (the minimum interarrival time is then taken as the period), with deadlines that must be guaranteed, while the Soft Real-Time traffic (SRT) is aperiodic, with timing constraints that could occasionally be missed without major consequences. In this paper, the problem of scheduling these two types of traffic with different performance objectives will be addressed.

Originally, the dual-priority scheme was introduced for the preemptive scheduling of tasks in (Davis, 1994). Many other approaches have appeared in the literature, aimed at improving the responsiveness of SRT tasks (Chetto and Chetto, 1989; Sprunt *et al.*, 1989; Lehoczky and Ramos-Thuel, 1992; Spuri and Buttazzo, 1996) but the dual-priority scheme is, for key practical reasons, applicable over a wide range of problems with low overheads and less restrictive hypotheses than most other approaches (Davis, 1994). For instance, knowledge of the first release time of a periodic source is not mandatory, thus making dual-priority scheme usable for message scheduling on a CAN network where nodes are not synchronised. In essence, this policy facilitates the responsive execution of SRT tasks by running all HRT tasks immediately, when there are no SRT tasks ready to run, or as late as possible where SRT tasks are ready to run. Tindell and Hansson (1995) proposed applying the dual-priority scheme for message scheduling (termed "dual-priority protocol") on CAN without evaluating its performance and without addressing the problem of transmission errors. In an earlier work, Tindell and Burns (1994) proposed a solution for calculating an upper boundary on HRT message response times on a CAN network. Knowledge of the worst-case response time, defined as the maximum interval between the transmission request and the complete reception of the message, is a requirement for the implementation of the dual-priority protocol. This paper's goal is, on the one hand, to evaluate the performance of dual-priority protocol on CAN with a reliable medium, and on the other hand, to propose a simple mechanism that provides probabilistic guarantees to prevent hard real-time frames from missing their deadlines when transmission errors occur. It is worth noting that the problem of transmission errors has to be taken into account, in the light of the existence of such disturbances (mainly caused by electromagnetic fields) in industrial and automotive communication systems, and the potentially disastrous consequences of failure to respect the time constraints. Throughout this paper, a basic transmission error model will be used: the occurrence of errors obeys a Poisson law. The same analysis could be applied with a more powerful error model, making assumptions not only about error frequency but also about error gravity (single errors and bursts of errors), see (Navet and Song, 1998; Navet *et al.*, to appear in 1999). In this paper, the problem of scheduling two types of traffic with different performance objectives will be addressed:

(1) ensuring that the timing requirements of HRT traffic are met with a chosen probability for a given perturbation level;
(2) minimising as far as possible the average response time of SRT traffic, while satisfying the above objective.

In Section 2, existing work on dual-priority scheme applied to message scheduling is recalled, as well as the basic principles of the CAN protocol. The performance of the dual-priority protocol is then assessed in an industrial case-study. Section 3 is devoted to a proposal for improving the reliability of the dual-priority protocol, and the evaluation of its performance.

2. DUAL-PRIORITY SCHEDULING ON CAN

2.1 *The dual-priority protocol*

Under dual-priority scheduling, as proposed in (Davis, 1994), the priority range must be partitioned into three bands: "low hard", "soft", "high hard" in increasing level of priority. An HRT task is first queued with a priority within the "low hard" band and later, when it becomes urgent, it will be promoted to the "high hard" range. The same applies to message scheduling (Tindell and Hansson, 1995): instead of transmitting HRT frames as soon as they are available, they can be deferred in favour of SRT data until they become urgent, thus lowering the average SRT traffic response time without a deterioration in HRT traffic.

Let t be the release time of an HRT message m, let D_m be its deadline (usually equal to the period P_m) and let R_m be its worst-case response time. The latest priority promotion time (from the "low-hard" to the "high-hard" range) such that the frame will meet its deadline, is R_m before its deadline, or is otherwise expressed as $(t + D_m - R_m)$ (Tindell and Hansson, 1995).

The dual-priority strategy could be applied to CAN if the assumptions on periodicities, priorities and message length made for the calculation of R_m are valid. Even the lowest-priority periodic frame could lead to missed deadlines if sent with characteristics other than those assumed. Tindell and Hansson (1995) suggest a "smart" CAN controller which permits messages to be sent on the bus only when they meet the assumptions made for the mathematical analysis, and thus preserve the calculated timing performance. Although a hardware solution is needed for critical systems, this work could be done by a software layer in less constrained systems. Tindell and Hansson also proposed that the frame priority be boosted at the controller level, using a priority "step-up" timer for each transmission buffer.

2.2 *CAN networks*

CAN is a multi-access broadcast bus with a priority-based access control to the medium. When two or more messages are competing for the bus, the one with the highest priority gains access to the bus, thus delaying the lower-priority message(s). Nodes do not possess an address, and none of them plays a preponderant role in the protocol. A message contains an identifier, 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) and allowing message filtering upon reception. In CAN, each station which detects an error sends an "error flag", and the corrupted frame will automatically re-enter into the next arbitration phase. 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 because it will further be assumed that all stations stay in the "error active" state and 23 bits is the maximum overhead with an "error active" transmitting node. Moreover, it will be assumed that the transmission error occurs at the last bit of the transmitted frame, in order to consider the worst-case hypothesis. The reader interested in the CAN protocol could consult (Kiencke and Kytölä, 1996), (ISO, 1994*a*) and (ISO, 1994*b*).

2.3 *Worst-case response time on CAN*

A periodic message with identifier m is characterized by (C_m, T_m, D_m) where C_m is the transmission time, T_m is the period and D_m the deadline. The worst-case response time R_m must be bounded for each frame by D_m, otherwise one cannot guarantee that the message has been sent before the next one is queued, and the set of messages of the application is said to be non-schedulable. To calculate R_m, as the maximum transmission time C_m is known, one just has to compute the maximum time needed by the message to gain the arbitration phase (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 delay, denoted by B_m, is the transmission time of the longest lower-priority message). Thus, (Tindell and Burns, 1994):

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

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

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

with τ_{bit} the transmission duration of one bit (termed 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 (Tindell and Burns, 1994) :

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

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 (ISO, 1994*b*).
I_m is computed starting with $I_m^0 = 0$ until convergence. The reader should refer to (Tindell and Burns, 1994) and (Tindell and Hansson, 1995) for more details.

2.4 *Performance of the dual-priority protocol with a reliable medium*

In the rest of the paper, an experimental embedded CAN-based application proposed by the PSA (Peugeot-Citröen) Automobiles Company will be considered (Navet and Song, 1996). Six devices exchange messages : engine controller, wheel angle sensor, AGB (Automatic Gear Box), ABS (Anti-Blocking System), device y (the name of which cannot be communicated because of confidentiality) and the bodywork gateway. The traffic consists of a set of periodic messages (e.g. speed and torque from the engine controller), see Fig. 1. This "time-triggered" traffic coexists with "event-triggered" one. The latter is assumed to be aperiodic with exponentially distributed interarrival times. Each periodic frame has its deadline equal to its period.

The transmission rate of the CAN bus is 125kbit/s. The size of the periodic frames is assumed to be 125 bits, and the size of aperiodic frames to be 100 bits. The periodic part of the traffic represents a network load of 53.46%. All nodes are assumed to transmit from the starting of the engine.

priority	transmitter	T_m	$D_m - R_m$
1	engine controller	10	8
2	wheel angle sensor	14	11
3	engine controller	20	16
4	AGB	15	10
5	ABS	20	14
6	ABS	40	33
7	ABS	15	7
8	bodywork gateway	50	41
9	device y	20	10
10	engine controller	100	88
11	AGB	50	37
12	ABS	100	86

Fig. 1. Set of periodic messages (time unit : ms)

Simulations were performed, using the QNAP2 discrete event simulator (Simulog, 1993), with and without dual-priority scheduling. In both cases, the average response time (see Fig. 2) and the variance in response times (see Fig. 3) were measured for aperiodic traffic with a total network load varying from 60% to 95%. Nearer to the 100% network load, the system becomes unstable and the simulation results for aperiodic traffic lose accuracy.

2.4.1. *Average response time for SRT traffic*

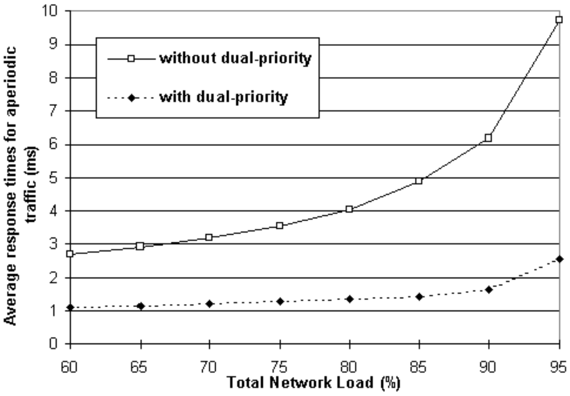

Fig. 2. Average response time for SRT traffic

From Fig. 2, it is obvious that the dual-priority protocol greatly reduces the average response time of aperiodic traffic. In this experiment, the observed gain ranges from a factor of 2.4 for a 60% load to a factor of 3.8 for a 95% load. It is also noteworthy that the dual-priority strategy offers very good resistance to an increase of the network load up to 90%.

2.4.2. *Variance in response times for SRT traffic*

The variance measures the dispersion of a sample of values relative to the mean value of the sample. In this case, the smaller the variance of the response times, the more deterministic the transmission of the aperiodic frames, because when transmitting a frame, one can expect its response

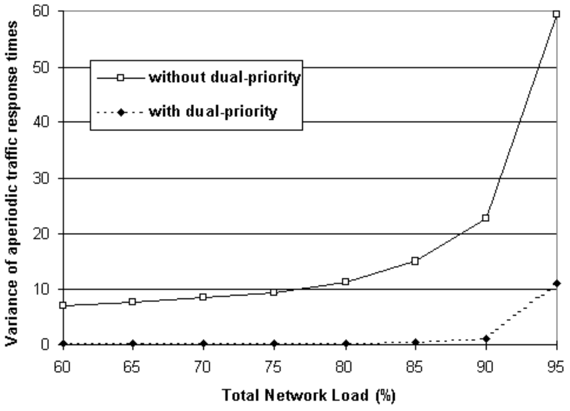

Fig. 3. Variance in response times for SRT traffic

time to be near the mean value. One clearly sees, from Fig. 3, that dual-priority hugely decreases the variance of aperiodic frame response times.

2.5 *The problem of transmission errors*

Local area networks often operate in an electromagnetically stressed environment. This is the case for in-vehicle networking, where CAN is widely used. Moreover, strong economic pressures could lead to the choice of an unshielded transmission support which does not provide very efficient immunity against electromagnetic disturbances (Barrenscheen and Otte, 1997). In this paragraph, it will be pointed out that the dual-priority scheduling is unsafe for HRT when transmission errors may occur.

An HRT message m is always released with a priority in the lowest range. If a sufficient number of consecutive aperiodic frames are queued, the message m will await its priority promotion time before having a chance to gain the bus.

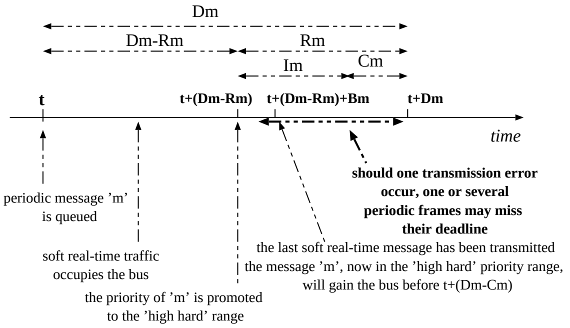

Fig. 4. The problem of transmission errors

The promotion to the highest priority range for a frame m occurs at $(t + D_m - R_m)$. In the worst case, the last SRT message releases the bus at $(t + D_m - R_m + B_m)$. Should a transmission error occur after $(t + D_m - R_m + B_m - 23\tau_{bit})$, where $23\tau_{bit}$ is the transmission time of the error frame, one or several HRT messages may miss their deadlines (see Fig. 4). In the worst case, a single transmission error may even cause all the HRT frames of the application to miss their deadline.

To illustrate this problem, Fig. 5 shows the percentage of missed deadlines for the twelve periodic frames of this application with a Frame Error Rate (FER) equal to 2.5%, 5% and 7.5% when dual-priority scheduling is turned on with a 100% network load. In a "real-world" application, it is very unlikely that the network load, on average, would reach 100%. Nevertheless, bursts of SRT messages may happen, and the network, over a small period of time, may be fully charged. For instance, if message 1 is queued during such a burst of SRT frames, it has a probability of missing its deadline equal to 3.06%, with an FER of 5%.

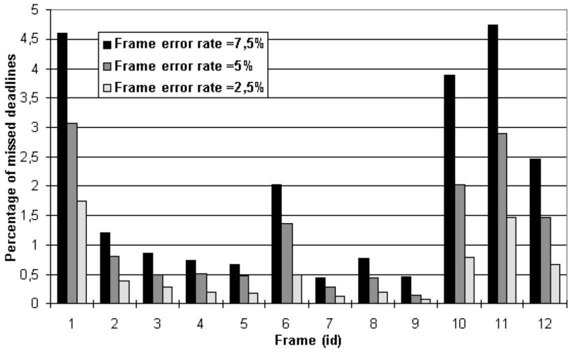

Fig. 5. Percentage of missed deadlines for HRT traffic with the dual-priority scheme

3. RELIABILITY IMPROVEMENT

It is a conventional belief that HRT traffic must have a 100% guarantee. In practice, this cannot be achieved when the environment or the behavior of some components of the system are not fully predictable. Should transmission errors occur randomly on the bus, then it is not unlikely that in the lifetime of the system, missed deadlines could happen. This is true with or without dual-priority scheduling. It is therefore crucial for the designer to assess the risks and, if necessary, take the appropriate fault-tolerance strategies (oversampling, mode change . . .). The reader might refer to (Kopetz, 1997) and (Ziegler *et al.*, 1994) for a good starting point on this topic.

This section proposes a scheduling strategy that provides probabilistic guarantees to prevent hard real-time frames from missing their deadlines whilst using the dual-priority protocol. The application designer has to define the required safety level for his particular application, through a parameter denoted α, which is defined as the upper bound on the Deadline Failure Probability (DFP). For instance, $\alpha = 0.001$ means that no periodic frame will, on average, miss its deadline more than once every 1000 transmissions.

3.1 *Principles*

Basically, the idea enabling this Quality Of Service (QOS) to be achieved, is to bring forward the priority promotion time if it is necessary regarding α. That amounts to including in the calculation of the response time (see eq. 1) the possibility of transmission errors. For each frame m of the application, one has to find the smallest tolerable number of errors n_m (starting with $n_m = 0$) such that the requested QOS will be achieved :

$$P[X\left(R_m(n_m)\right) > n_m] \leq \alpha \qquad (4)$$

which can be rewritten as :

$$\left(1 - \sum_{n=0}^{n_m} P[X(R(n_m) = n]\right) \leq \alpha \qquad (5)$$

where $X(t)$ is the stochastic process giving the number of errors during time t (see eq. 9) and $R_m(n)$ the worst-case response time with n errors for frame m. If fewer than n_m errors occur during $R_m(n_m)$ with $R_m(n_m) \leq D_m$, the frame m will meet its deadline; otherwise it could miss it. The calculus is performed iteratively, starting with $n_m = 0$ and incrementing this at the end of each unsuccesful step until eq. 4 is satisfied or until $R_m(n_m)$ is larger than the deadline. In the latter case, the requested QOS cannot be achieved. For the calculation of $R_m(n_m)$, the following relations are needed:

$$R_m(n) = C_m + I_m(n) \qquad (6)$$

with :

$$I_m^n(n) = E_m(n) + B_m + \sum_{\forall j \,\in\, hp(m)} \left\lceil \frac{I_m^{n-1} + \tau_{bit}}{T_j} \right\rceil C_j \qquad (7)$$

where $E_m(n)$ is the error recovery overhead function defined as:

$$E_m(n) = n \left(23\tau_{bit} + \underset{\forall j \,\in\, hp(m) \cup \{m\}}{max} C_j\right). \qquad (8)$$

The concept of an error recovery overhead function $E()$ was introduced in (Tindell and Burns, 1994) as a function of time; it becomes here a function of the number of errors, n.

Note that $P[X\left(R_m(n_m)\right) > n_m]$ is an upper bound on the DFP because underlying it is the assumption that each error occurs exactly at the last bit of a transmission, which is why it is termed the "worst-case DFP", see (Navet *et al.*, to appear in 1999).

In the case where the error model is assumed to obey the Poisson law with parameter λ, one has:

$$P[X(t) = k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \qquad (9)$$

Parameter λ can be considered as the measurement of the degree of perturbation of the environment, and is defined as the average number of errors per second. The relation between between λ and the FER is given by eq. A.1 (see Appendix A). The application designer has to chose a suitable value for parameter λ for his particular application. This should be done using measurements carried out on a prototype, or from the experience gained on similar systems. Some new CAN controllers possess features that facilitate this task, such as readable error counters (Hausmann and Gebing, 1997) or the possibility of triggering an interruption when an error occurs (Hank, 1997). From the resulting FER, which is a percentage, one must derive the parameter λ giving the average number of errors per second (see eq. A.1). Note that if the application frames are not all of the same size, they are not equally affected by transmission errors. One must thus derive, from a global λ, the λ_m corresponding to frame m (see eq. A.2).

Considering the application introduced in Section 2.4, with an FER of 5% (λ equal to 53.13) for HRT traffic and α set to 0.001, the results shown in Fig. 6 are found for n_m, for the corresponding $R_m(n_m)$ and for the Priority Promotion Delay (denoted PPD and equal to $D_m - Rm(n_m)$).

Frame (id)	n_m	$R_m(n_m)$ (ms)	PPD (ms)
1	3	5.55	4.45
2	3	6.55	7.45
3	3	7.55	12.45
4	4	9.74	5.26
5	4	10.74	9.26
6	4	12.74	27.26
7	5	14.92	0.08
8	5	18.92	31.08
9	5	19.92	0.08
10	6	26.10	73.90
11	6	27.10	22.90
12	7	30.29	69.71

Fig. 6. Results for n_m, $R_m(n_m)$ with $\alpha = 0.001$ and FER=5%

For the sake of simplicity, the analysis only considers a unique DFP tolerance parameter α for all frames, but this parameter could be adjusted to reflect the criticality of each frame; for example, by weighting the global parameter α with the frame's relative priority.

3.2 *Performance evaluation*

The proposal outlined in the previous paragraph gives probabilistic guarantees on deadlines for HRT traffic, depending on the chosen α value. By bringing forward the priority promotion time, this proposal has the negative aspect of degrading the performance compared to the "pure" dual-priority protocol. In this paragraph, the loss of performance for SRT traffic will be quantified in terms of the average response times and the variance in response times.

3.2.1. *Average response times for SRT traffic*

Average response times of SRT traffic for various α values, for a network load of 85% and 90% and for an FER equal to 5%, are presented in Fig. 7. From this figure, one can clearly see that the new proposal only slightly increases the average response time for SRT traffic compared to pure dual-priority scheduling, even with rather high probabilistic guarantees as to deadline. For instance, with $\alpha = 0.001$ and a 90% network load, the average SRT response time is still 2.8 times lower than without dual-priority scheduling and only 1.4 times longer than with pure dual-priority scheduling.

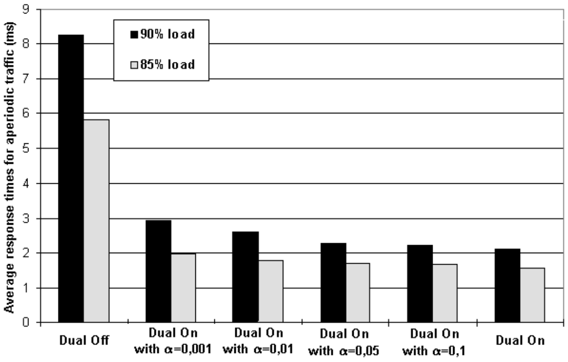

Fig. 7. Average response times for SRT traffic for different values of α with FER=5%

3.2.2. *Variance in SRT traffic response times*

From Fig. 8, it is clear that the performance loss for SRT response time variance is limited. With a 90% network load, the variance is equal to 5.59 with pure dual-priority, 12.61 with $\alpha = 0.001$, and 46.38 without dual-priority. From this figure, one can also note that the variance in SRT response time is very load-dependant; its value quickly decreases with a lower network load.

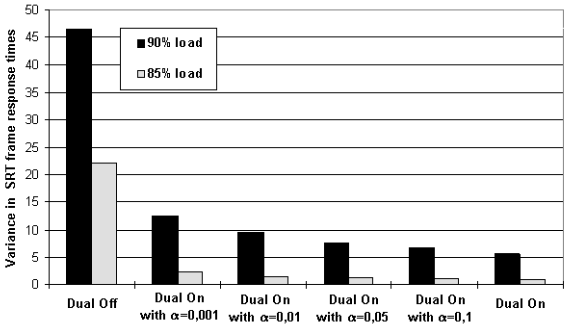

Fig. 8. Variance in SRT traffic response times for different values of α with FER=5%

4. CONCLUSION

In this paper, the performance of dual-priority scheduling on CAN with a reliable medium has been evaluated, showing that this scheduling strategy greatly diminishes both the response time and the variance in response times for SRT traffic. On the other hand, the problem of transmission errors has been pointed out, and a simple mechanism has been proposed, providing probabilistic guarantees to prevent hard real-time frames from missing their deadlines in the event that transmission errors occur. This proposal only slightly degrades the performance of the dual-priority protocol, and provides a guaranteed QOS for HRT traffic that can be chosen on an application-per-application basis, or even on a frame-per-frame basis.

Certain new CAN controllers have some interesting error-signalling features such as readable error counters or interrupt-triggering on transmission occurrence. Those features will enable the determination of an error model parameter-setting procedure that will dynamically change the parameter's values when these become improper in the light of the current bus perturbation level. Such an on-line adaptive parameter-setting procedure would be well suited for systems within which the bus perturbation level may vary greatly over time, such as automotive communication systems. Preliminary results show that calculating the FER periodically using the "exponential smoothing" technique (Wonnacott and Wonnacott, 1990) gives satisfactory results, both in terms of computing time (constant and very small) and in terms of the quality of the approximation.

Appendix A. USEFUL RELATIONS

The relation between FER and λ is :

$$FER = \frac{\lambda}{\widetilde{n}} \tag{A.1}$$

with λ the mean number of corrupted frames per second and $\widetilde{n}$ the mean number of frames per second:

$$\widetilde{n} = \frac{R\eta}{\widetilde{S}}$$

where R is the transmission rate (bit/s), η the mean network load (equal to the mean SRT load, η_a, plus the mean HRT load, η_p) and $\widetilde{S}$ the mean frame size :

$$\widetilde{S} = \left(\widetilde{S_a}\,\frac{\eta_a}{\eta}\right) + \left(\widetilde{S_p}\,\frac{\eta_p}{\eta}\right)$$

with $\widetilde{S_a}$ the mean size of SRT traffic, whose calculation is application-dependent, and $\widetilde{S_p}$, the mean size of HRT traffic calculated as follows:

$$\widetilde{S_p} = \frac{\sum\limits_{m \in \mathbf{M}} \frac{S_m}{T_m}}{\sum\limits_{m \in \mathbf{M}} \frac{1}{T_m}}$$

where $\mathbf{M}$ is the set of HRT messages and S_m is the size of HRT message m.

λ_m, the individual error arrival rate, can be derived from λ:

$$\lambda_m = \frac{\lambda}{\widetilde{S}} S_m. \qquad \text{(A.2)}$$

REFERENCES

Barrenscheen, J. and G. Otte (1997). Analysis of the physical CAN bus layer. *4^{th} international CAN Conference, ICC'97*. pp. 06.02–06.08.

Chetto, H. and M. Chetto (1989). Some results of the earliest deadline scheduling algorithm. *IEEE Transactions on Software Engineering* **15**(10), 1261–1269.

Davis, R. (1994). Dual priority scheduling : a means of providing flexibility in hard real-time systems. Technical report. Department of Computer Science, Univ. of York (UK). Technical Report YCS230.

Hank, P. (1997). PeliCAN : a new CAN controller supporting diagnosis and system optimization. *4^{th} international CAN Conference, ICC'97*. pp. 04.12–04.18.

Hausmann, G. and E. Gebing (1997). 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. *4^{th} international CAN Conference, ICC'97*. pp. 4.02–4.11.

ISO, International Standard Organization (1994*a*). *Road Vehicles - Interchange of Digital Information - Controller Area Network for high-speed Communication*. ISO. ISO 11898.

ISO, International Standard Organization (1994*b*). *Road Vehicles - Low Speed serial data communication - Part 2: Low Speed Controller Area Network*. ISO. ISO 11519-2.

Kiencke, U. and T. Kytölä (1996). CAN, a ten years' anniversarial review. *3th international CAN Conference, ICC'96*. pp. 2.2–2.7.

Kopetz, H. (1997). *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers. Boston.

Lehoczky, J. and S. Ramos-Thuel (1992). Aperiodic task scheduling for hard-real-time systems. *Proceedings IEEE Real-Time Systems*. pp. 110–123.

Navet, N. and Y.-Q. Song (1996). Evaluation de performances de la messagerie CAN du véhicule prototype PSA - Action 1 du contrat PSA-CRIN. Contract report 96-R-182.

Navet, N. and Y.-Q. Song (1998). Design of reliable real-time applications distributed over CAN (Controller Area Network). *IFAC Symposium on Information Control in Manufacturing (INCOM'98)*.

Navet, N., Y.-Q. Song and F. Simonot (to appear in 1999). Worst-case deadline failure probability in real-time applications distributed over can (controller area network). *Journal of Systems Architecture - the Euromicro Journal*.

Simulog (1993). Qnap2 Version 9.0 - Reference Manual.

Sprunt, S.P., L. Sha and J. Lehoczky (1989). Aperiodic task scheduling for hard-real-time systems. *Real-Time Systems* **1**(1), 27–60.

Spuri, M. and G. Buttazzo (1996). Scheduling aperiodic tasks in dynamic priority systems. *Real-Time Systems* **10**(2), 179–210.

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

Tindell, K. and H. Hansson (1995). Babbling idiots, the dual priority protocol, and smart CAN controllers. *2^{nd} international CAN Conference, ICC'95*. pp. 7.22–7–28.

Wonnacott, T.H. and R.J. Wonnacott (1990). *Introductory Statistics*. John Wiley and Sons. New York. 5th edition.

Ziegler, C., D. Powell and P. Desroches (1994). Dependability of on-board automotive computer systems. *IEEE Intelligent Vehicles 1994 Symposium*. pp. 568–575.