Networked Control System (NCS) on a network CAN: on the Quality of Service (QoS) and Quality of Control (QoC) provided by different messages scheduling schemes based on hybrid priorities

Xuan Hung Nguyen, Guy Juanole, Gérard Mouney, Christophe Calmettes

Networked Control System (NCS) on a network CAN: on the Quality of Service (QoS) and Quality of Control (QoC) provided by different message scheduling schemes based on hybrid priorities

Xuan Hung Nguyen, Guy Juanole, Gérard Mouney[1,2]
[1]LAAS-CNRS ; 7 avenue du Colonel Roche, F-31077 Toulouse, FRANCE
[2]Université de Toulouse ; UPS, INSA, ISAE ; LAAS-CNRS : F-31077 Toulouse, FRANCE
xhnguyen@laas.fr, juanole@laas.fr, mouney@laas.fr

Christophe Calmettes
Université de Toulouse ; CUFR JF Champollion
Place de Verdun 81012 Albi Cedex 9, FRANCE
christophe.calmettes@univ-jfc.fr

Abstract

The message scheduling is an important mechanism in the context of Networked Control Systems. The message scheduling is, in several networks, mainly based on static priorities associated to the messages (for example the network CAN). This static priority scheme has intrinsic limitations and so here we will consider scheduling strategies based on hybrid priorities. Three hybrid priority schemes, resulting from our previous works are first presented. Then we compare the Quality of Service (QoS) offered by these schemes to a process control application and we evaluate the resulting Quality of Control (QoC).

1. Introduction

In the technological context of today, distributed systems (computers connected through a communication network) are more and more used, particularly, for implementing industrial applications like, for example, process control applications based on a closed loop structure [1]. The study and the design of such systems, called Networked Control Systems [2], represent a very important area of research because of its pluridisciplinary aspect. This aspect involves a deep knowledge in Automatic Control, Computer Science (mainly task scheduling) and Communication Networks (mainly message scheduling and communication protocols).
A process control application includes three remote tasks (the sensor task, the controller task and the actuator task) which are on different sites and which then require the periodic exchange of two message flows (the choice of the period depends on the application dynamic [1]) through the network: the sensor flow which concerns the transfer of the output samples, from the sensor of the process to the controller which computes the control law; the controller flow which concerns the transfer of the control samples from the controller (these samples result from the computation done after the reception of the samples coming from the sensor) to the actuator of the process.
The scheduling through the network of the messages of these two flows (which share the network with the flows of the other applications) is an essential mechanism which strongly influences the settling time and the stability [3] of a process control application. This paper is precisely concerned by this problematic by considering the network CAN [4] and the MAC layer which implements the scheduling of the frames (which carry the messages of the applications). The scheduling is done by means of priorities which are represented in the IDentifier (ID) field of the frames. Different types of priorities (static priority, hybrid priority) can be considered. Here we focus on hybrid priority schemes [5].

The goal of this paper is mainly to evaluate the Quality of Service (QoS) provided by three hybrid priority schemes to a process control application, to compare these QoS and to explain their differences. We also give the consequent Quality of Control (QoC) *i.e.* performances related to the process control application. This study is done by using the simulator TrueTime [6] which allows to represent NCS (both network and control aspects).

This paper includes the following two sections. The section 2 presents the context of the study (the process control application, the implementation through a network, the message scheduling on CAN) and the three message scheduling schemes which are considered. The section 3 presents the QoS obtained with these different strategies and the resulting QoC for the process control application.

978-1-4244-5461-7/10/$26.00 ©2010 IEEE

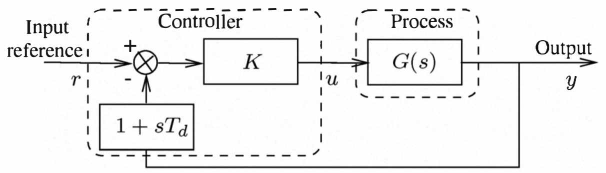

Figure 1. Model

2. Context of the study

The study considers the case of the implementation through the network CAN of one process control application and others applications which share the network with the process control application.
We present, at first, the process control application, which is considered, then the general view of its implementation through a network (model, choice of the sampling period, information transmission rate requested to the network) and finally the different priority schemes for the frame scheduling.

2.1. The process control application which is considered

This application is represented on the figure 1: the process control application has a transfer function $G(s) = \frac{1000}{s(1+s)}$; the controller is a proportional derivative controller which considers the output derivation (K is the proportional gain ($K = 1.8$); T_d is the derivative time constant ($T_d = 0.032$ s)).
The closed loop transfer function $F(s)$ of this application is

$$F(s) \quad = \frac{KG(s)}{1+KG(s)(1+T_d s)} \quad = \frac{1000K}{s^2+(1+1000KT_d)s+1000K}$$

We have an overshoot of 5% and the rise time t_r is about $40ms$.

2.2. General view of the implementation through a network

1/ The network operates both (figure 2):

- between computer 1 (C1) in association with the numerical information provided by the AD conversion (this computer includes a task that we call the sensor task and which generates the sensor flow addressed to the controller; we note fsc this flow) and computer 2 (C2) where we have the reference and the controller (in C2 we have a task called controller task which generates the controller flow addressed to the actuator; we note fca this flow); fsc goes from C1 to C2 and fca goes from C2 to the computer 3 (C3).

- and between C2 and C3 which provides numerical information to the DA conversion in front of the Zero Order Hold (ZOH).

The task which generates the flow fsc is time-triggered (the sampling is based on a clock) whereas the task which generates the flow fca is event-triggered (the controller waits for sensor sample reception before computing the control action and generating its flow).

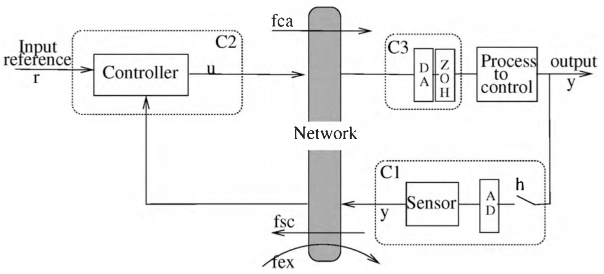

Figure 2. Implementation through a network

Generally a network is not dedicated to only one application but shared between different applications. In order to make a general study of the behaviour of the process control application, when it is implemented through a network, we have to see, in particular, the influence of the flows of the other applications. It is why we have, in figure 2, what we call the external flow, noted fex, which globally represents an abstraction of the flows of all other applications. We also consider that fex is periodic.

2/ The choice of the sampling period (h) of the process control application is a basic action. The sampling period has, from the control point of view, an upper bound. But from the network point of view, a value that is too small gives load that is too great. So, the choice results from a compromise. The relation $\frac{t_r}{10} \leq h \leq \frac{t_r}{4}$, which has been given in [1] is generally used. We consider here the bound $\frac{t_r}{4}$. As $t_r \approx 40$ ms, we have $h = 10$ ms.
Taking into account for the implementation through a network, the controller, defined in the subsection 2.1, needs to be discretized with the sampling period. With this discretization, the measured dynamic characteristics are now an overshoot less than 1 %, a rise time $t_r \simeq 34$ ms and a settling time (at 5 %) $t_s \simeq 50$ ms. The characteristics (overshoot and settling time) will be our references to analyze the performances of the control application through the studied networks.

3/ In a general way, the information transmission rate requested by the applications to a network is the pertinent parameter to compare the efficiency of the message scheduling. We call here this parameter the User Request Factor (URF).
Concerning the network CAN, the scheduling is done by the MAC layer and concerns the frame scheduling.
By calling:

- D_{ca}, D_{sc}, D_{ex} the duration of the fca frame, the fsc frame and the fex frame, respectively,

- h the sampling period of the process control application (the period of fsc and consequently of fca) and T_{ex} the period of the external flow.

we have $URF = \frac{D_{ca}}{h} + \frac{D_{sc}}{h} + \frac{D_{ex}}{T_{ex}}$.

In the context of this work, we will consider the following numerical values:

- bit rate in the physical layer of CAN: 125 Kbits/s,
- length of 10 bytes for the fsc frames and fca frames (thus a duration of $D_{sc} = D_{ca} = 640\mu$s),
- length of 15 bytes for the fex frames (thus $D_{ex} = 960\mu$s).

The component $\frac{D_{ca}}{h} + \frac{D_{sc}}{h}$ of the URF, which concerns the process control application and which represents the network capacity used by this application, has the value 12.8 %. The use by the external frame of the network capacity will depend on its period T_{ex}. It is this parameter that we will vary during our study in order to analyze the robustness of the scheduling of the process control application frames.
The frame scheduling in the MAC layer of CAN [4] is done by comparing the field ID bit by bit (we start from the Most Significant Bit (MSB)). In CAN the bit 0 is a dominant bit and the bit 1 is a recessive bit. The lower the numerical value of the field ID, the higher the priority. We consider here the standard length of 11 bits for the field ID.

2.3. Static priority scheme

This scheme is the scheme defined in the standard: the priority represents the priority of the flow to which the frame belongs.
We have done studies [5] in such a context and we have shown that we have to take the priority of the flow fca higher than the priority of the flow fsc in order to get the best results (it is what we will consider here). However, in these studies we have also shown that the static priority scheme is unable to provide the necessary Quality of Service to the flows of a process control application (in particular, if these flows have not the highest priority and if the flows of the others applications, which have the highest priority, request very frequently the network use, the process control application will have very bad performances in the transient behaviour). An example of this fact is demonstrated by the step response given on the figure 3 (Priority of fex > Priority of fca > Priority of fsc). By considering the period Tex of the external flow such that $Tex = h/9$ (then we have $URF = 99.2\%$), we get an overshoot of 31% and response time of 420 ms. Then, in the NCS context, the scheduling of the frames of the flows of a process control application, must integrate, in more of the static priority of the concerned flows, the needs, in terms of transmission urgency, of the frames.
Hence the concept of the hybrid priority scheme which was first defined by [3] and that we adapted to the NCSs; we present now the three priority schemes that we have defined [5].

2.4. General idea on hybrid priorities

1- *The field ID and the scheduling execution.*
The identifier field of a frame is divided into two levels (figure 4): the first level (n nits) represents the priority of a flow (it is a static priority specified off-line); the second level (m bits) represents the priority of the transmission urgency (the urgency can be either constant or variable). The idea of structuration of the ID is present in the Mixed Traffic Scheduler [7] which combines EDF (dynamic field) and FP (static field). In [3] the authors propose encoding the weighted absolute value of the error in the dynamic field (this idea is also presented in [8]) and to resolve the collisions with the static field.

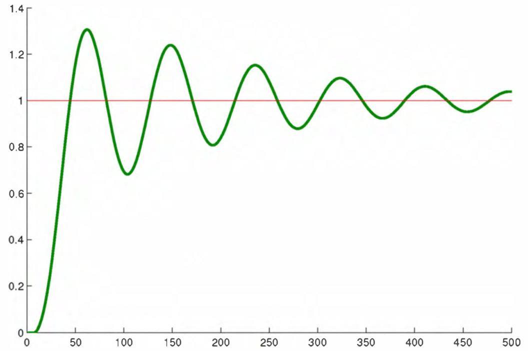

Figure 3. static scheme: step response (t_s = 420 ms, overshoot greater than 30%)

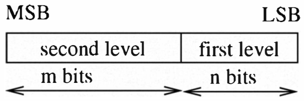

Figure 4. Field ID (Hybrid Priority)

A constant transmission urgency is characterized by a static priority (one m bit combination) specified off-line. A variable transmission urgency is characterized by a dynamic priority (which can take, generally speaking, m-bit combination among a subset of the m-bit combinations).
The frames of the flows fsc and fca of a process control application have variable needs (strong urgency in a transient behavior after an input reference change (in order to follow the change quickly) or after a disturbance (in order to make the regulation quickly); weak urgency in a permanent behavior). That is why, in this study, we consider that the dynamic priority of the frames of the flows fsc and fca of a process control application can take any m-bit combination of the set of the m-bit combinations. The scheduling is executed by, first, comparing the second level (needs predominance), and, secondly, if the needs are identical, by comparing the first level (flow predominance).
Remark: for the first level of the field ID we will consider here: Priority fex > Priority fca > Priority fsc.

2- *Cohabitation of flows with constant needs and flows of process control applications (variable needs)*
We have the objective of good performances for the pro-

cess control applications in transient behavior. This means the urgent needs of these flows must be satisfied very quickly. For that, we impose a maximum with constant needs for the priority of these needs (concept of priority threshold Pr_th for the constant needs). In this way, a strong transmission urgency of a process control application flow (dynamic priority with a very high value *i.e.* higher than Pr_th) will be scheduled first.
Remark: the external flow fex will have in this study constant needs (characterized by Pr_th).

3- *Toward making the dynamic priorities*
The concept of the dynamic priorities requires specifying, at first, the characteristic of a process control application which gives informations on the needs, and, secondly, how these needs can be translated into a dynamic priority (computation of a dynamic priority, instants of re-evaluation of a dynamic priority). We propose to express the needs with a signal which aptly characterizes the behavior of a process control application: it is the control signal u.

2.5. Three hybrid priority schemes

We have defined three schemes. The first is what we call the strict hybrid priority (*hp*) scheme (computation of the dynamic priority directly from a function of the control signal u; re-evaluation after each sampling instant). The second is the hp scheme extended with a static time strategy (sts) for the re-evaluation of the dynamic priority (re-evaluation not always after each sampling time). This scheme is noted *hp+sts*. The third is a scheme which does not compute the dynamic priority directly from the control signal u (definition of a timed dynamic priority reference profile and trip in this profile by means of an on-line temporal supervision based on a function of the control signal u). The dynamic priority is re-evaluated after each sampling instant. This third scheme, which implements a dynamic time strategy for the trip in the timed dynamic reference profile, is noted *hp+dts*.
We now detail these three schemes.

hp scheme

The needs are translated into a dynamic priority by considering an increasing function of $|u|$ (call it $f(|u|)$) characterized by a saturation for a value of $|u|$ less than the maximum of $|u|$ (noted $|u|\text{max}$). We do not want the dynamic priorityto take its highest value only when $|u|$ is maximum but already for values before the maximum, in order to react quickly as soon as the needs begin to become urgent. So we decide (it is an arbitrary choice) to take $\frac{2}{3}|u|\text{max}$ as the value of $|u|$ where the dynamic priority reaches its highest value $P\text{max}$.
Several functions $f(|u|)$ have been studied, for this work we consider the function $f(|u|)$ represented in figure 5. This function is defined by:

$$f(|u|) = \begin{cases} P\text{max}\sqrt{\frac{|u|}{\frac{2}{3}|u|\text{max}}}, & 0 \leq |u| \leq \frac{2}{3}|u|\text{max} \\ P\text{max}, & |u| > \frac{2}{3}|u|\text{max} \end{cases}$$

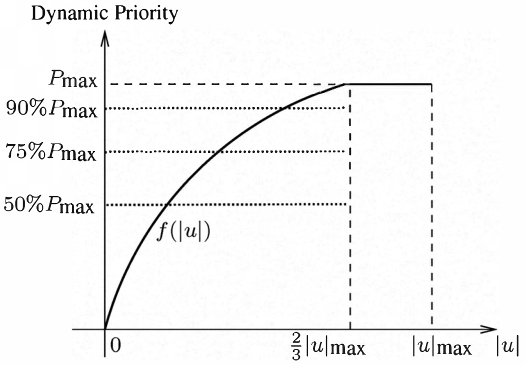

Figure 5. The considered non linear function

The computation of the dynamic priority is done by the controller each time it receives a frame that the sensor sends after each sampling instant (dynamic priority re-evaluated after each sampling instant). Then, after the reception of a frame from the sensor, the controller sends a frame with the value of the new dynamic priority. This frame reaches all the sites (CAN is a bus) and as the sensor site knows the first level of the ID of fca (it is a constraint for our implementation), it will learn the dynamic priority that it will put in the next frame that it will send (the dynamic priority is then used by the two flows of a process control application).
Taking into account the task implementation (sensor task is time-triggered, controller task is event-triggered), it is the sensor task which transmits the first frame at the start of the application. For this first frame, the sensor site has no information about the dynamic priority and thus we consider that it uses the maximum priority. This way, the first fsc reaches the controller site as quickly as possible.

(hp+sts) scheme

A criticism of the hp scheme is that we can have oscillatory behavior of the dynamic priority values (resulting from a damped sinusoidal transient behavior of u). We can have, for example, this scenario for the dynamic priority values at three successive re-evaluation instants [9]: the highest value at the first re-evaluation instant, then an intermediary value at the second, and again the highest value at the third re-evaluation instant ... Such an oscillatory behavior shows that the control of a situation requiring a big value of the dynamic priority is inadequate in terms of the maintenance of this big value, since after leaving this value for an intermediary one, at the second re-evaluation instant, we come back to this big value at the third re-evaluation instant. The observation of this phenomenon suggests increasing the duration of the dynamic priority with a big value in order to improve transient behavior.
The (hp+sts) scheme is then the following. Contrary to the scheme hp, where the dynamic priority is re-evaluated in the controller site, after each reception of a fsc frame, the instant of the re-evaluation is no longer so closely related

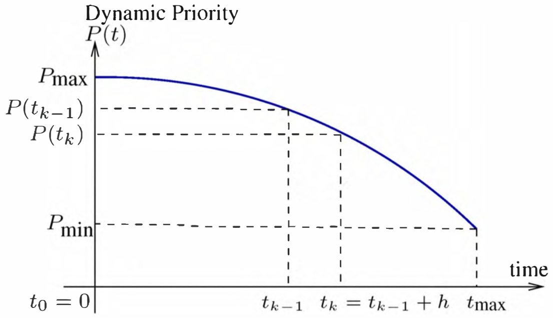

Figure 6. Reference profile

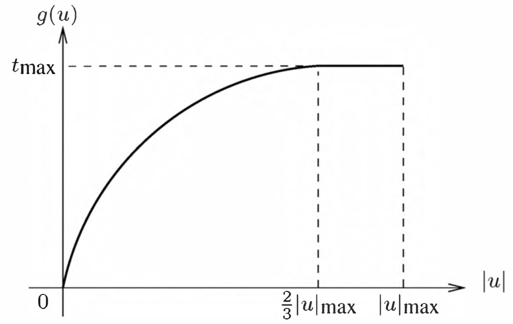

Figure 7. Exemple of $g(u)$

to the sampling instants. Here the duration of the time interval between two successive re-evaluations depends on the value of the dynamic priority at the beginning of the time interval. This duration must be relevant, in particular, from the point of view of the transfer function of the process control application and more precisely, of its transient behavior (defined before its implementation through the network). We considered the following algorithm:

- if the dynamic priority has a value between the highest priority ($P_{\max}$) and half the highest priority ($\frac{1}{2}P_{\max}$), we keep this value for 4 sampling intervals and we re-evaluate the dynamic priority afterwards; this duration is equal to the rise time t_r (we have chosen $h = \frac{tr}{4}$) which represents a good characteristic of the transient behavior).

- if the dynamic priority has a value inferior to half the highest priority, we re-evaluate it after each sampling instant as in the previous algorithm.

(hp+dts) scheme

Main ideas We define, at first, a reference profile of dynamic priorities for apprehending with efficiency one transient behavior (*i.e.* an input change or a disturbance). It consists in a continuous decreasing time function from a priority $P_{\max}$ (start of the transient behavior) to a priority $P_{\min}$ (end of the transient behavior and then the beginning of the permanent behavior), which gives the values of the dynamic priorities at all the sampling times (these values are decreasing).

However the only consideration of the reference profile is not enough to handle the actual behavior. In the actual behavior, we have to take into account for the influence of the network and also the possibility of successive input changes and/or disturbances which lengthen the transient behavior with respect to the one considered in the reference profile. Then, actually, the temporal evolution of the dynamic priorities, cannot be always decreasing *i.e.*, at a sampling instant, we can, by considering the reference profile curve, move back to values higher than the value of the previous sampling instant.

So, in order to take into account for an actual behavior, we add a component, called on-line temporal supervision based on a function $g(u)$ which will allow to do, with respect to the reference profile, the temporal repositioning of the values of the dynamic priorities.

Reference profile We consider the reference profile represented in figure 6. The function $P(t)$ is defined by:

$$P(t) = P_{\max} - (P_{\max} - P_{\min})(\frac{t}{t_{\max}})^2 \quad 0 \le t \le t_{\max}$$

$P_{\min}$ is the priority used in the permanent behavior.

The dynamic priority decreases slowly at the beginning of the transient behavior (we need several successive sampling instants with high priority in order to be reactive) and more quickly towards the end. The reference profile expresses that the priority, related to the sampling instants, t_k, is lower than the priority related to the previous sampling time, t_{k-1}.

Concerning the time $t_{\max}$: as our objective is to tend towards a transient behavior guided by the transient behavior of the process control application without the network, we take $t_{\max}$ equal to the settling time at 5% of the process control application without the network.

On line temporal supervision We have defined several functions $g(u)$ which allow, at the sampling instant t_k, to move back in the reference profile with respect to the previous sampling instants t_{k-1}. These functions $g(u)$ give the time values which must be subtracted to the value $t_{k-1} + h$ to come back more or less towards the beginning of the reference profile (then using, at the instant t_k, a priority higher than at the instant t_{k-1}). Note that the maximum of this time value can be $t_{\max}$. Here we use the function $g(u)$ represented in figure 7 and defined by:

$$g(u) = \begin{cases} t_{\max}\sqrt{\frac{|u|}{\frac{2}{3}|u|_{\max}}}, & 0 \le |u| \le \frac{2}{3}|u|_{\max} \\ t_{\max}, & |u| > \frac{2}{3}|u|_{\max} \end{cases}$$

Then $g(u) \in [0, t_{\max}]$.

Algorithm for computing the dynamic priority at the sampling instant t_k The operations of the algorithm consist in positioning, when the controller receives the fsc frame, at an instant in the interval of time $[0, t_{\max}]$ of the definition of the reference profile and in detecting the value of the dynamic priority to use from this instant.

The value of the instant depends on the value of $g(u)$ at the reception of the fsc frames.

Initially (configuration of the system) the reference profile is at point value ($t_{\max}$, $P_{\min}$), then upon the reception of a fsc frame the controller:

1. computes $g(u)$
2. computes $x = t_k - \alpha g(u)$
   α is a coefficient, defined by $\alpha = \frac{t_k}{t_{\max}}$ $(0 \leq \alpha \leq 1)$ which balances the influence of $g(u)$ by increasing this influence even more because the dynamic priority is low (when the priority is low, a large value of $g(u)$ must induce greater feedback; it is not as necessary when the priority is already high).
   - if $x \leq 0$, we go to the time 0 on the reference profile ($P_{\max}$)
   - if $0 \leq x \leq t_{\max}$, we go to the time x in the reference profile
3. reinitializes the virtual time for the next sampling $t_k = x + h$; if $t_k > t_{\max}$, $t_k = t_{\max}$. This value will be used for computing the dynamic priority on the reception of the next fsc frame.

3. Study of the three schemes based on hybrid priorities

3.1. Study conditions

We consider the process control application which was presented in the subsections 2.1 and 2.2. The input is a position step which starts at time 0 and we study the transient behavior until it reaches permanent behavior.

The QoS parameters, which need to be taken into consideration, are the mean delay $\bar{D}$ of the control loop and its standard deviation σ. The QoC parameter is the settling time at 5% (t_s) which is obtained directly from the tool TrueTime.

In order to evaluate the QoS parameters, we use, on the one hand, the message exchange temporal diagrams which are also provided by TrueTime, and the value of t_s.

From the message exchange temporal diagrams, we can get the delay in the control loop (delay of the message of the flow fsc + delay of the message of the flow fca+ $Dsc + Dca$) for each sampling period (call D_i this delay for the sampling period i). Counting the number n of sampling periods in the settling time t_s, we deduce the value of $\bar{D}$ and σ by these formulas: $\bar{D} = \frac{\sum_{i=1}^{n}(D_i)}{n}$ and $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(D_i - \bar{D})^2}{n}}$.

In order to make a quantitative analysis, we cause a variation in the network load (URF) by varying the period T_{ex} of the external flow: we consider an external flow, the frequency of which (noted $\frac{1}{T_{ex}}$) is a multiple of the sampling frequency ($\frac{1}{h}$). The different URFs being considered are given in table 1.

URF (%)	Multiple of $\frac{1}{h}$	T_{ex} (ms)
99.2	9	1.1111
89.6	8	1.25
80	7	1.4286
70.4	6	1.6667
60.8	5	2.0
51.2	4	2.5
41.6	3	3.3333
32	2	5.0
22.4	1	10.0

Table 1. Different URFs

The following important points must still be emphasized:

- the flows fsc (which are generated at the sampling times) and fex are synchronous (starting at the same time) and as we consider the cases where the frequency of fex is a multiple of the sampling frequency, then their medium access attempts coincide at every sampling time;
- up to the value 70.4% of the URF (value of 1.6667 ms for Tex), we can see that during Tex, one frame of each flow can access the medium: 0.96 ms + 0.64 ms = 1.6 ms < 1.6667 ms (the third flow can begin to be transferred and then can not be interrupted). This remark is very important for the analysis which is done in section 3.3;
- a last point must be still noted: at the beginning of a transient behavior, as the control signal is at a maximum, the dynamic priority of the flows of the process control application is $P_{\max}$. This point also is important for the analysis in subsections 3.2, 3.3 and 3.4.

3.2. hp scheme

As concerns the process control application, we give $\bar{D}$ and σ in table 2 and t_s in table 3. The values depend on the network load URF (which depends on the frequency fex), and on the priority threshold Pr_th (which depends on the importance we give to fex).

	Pr_th					
URF	$0.9P_{\max}$		$0.5P_{\max}$		$0.25P_{\max}$	
(%)	$\bar{D}$	σ	$\bar{D}$	σ	$\bar{D}$	σ
99.2	5.333	1.680	3.743	2.262	1.804	1.380
89.6	3.264	1.286	2.240	1.228	1.629	0.846
80.0	2.48	0.887	1.978	0.828	1.5418	0.592
70.4	1.891	0.462	1.716	0.478	1.472	0.384
51.2	1.891	0.462	1.716	0.478	1.472	0.384
22.4	1.891	0.462	1.716	0.478	1.472	0.384

Table 2. hp scheme (QoS): $\bar{D}$ and σ (ms)

Concerning the values of $\bar{D}$, we observe the following main points:

- For each value of Pr_th:
  For $URF \leq 70.4$ %, we note that we have the same

URF	Pr_th		
(%)	$0.9P_{\max}$	$0.5P_{\max}$	$0.25P_{\max}$
99.2	359	228	105
89.6	148	110	103
80.0	111	108	101
70.4	107	105	99
51.2	107	105	99
22.4	107	105	99

Table 3. hp scheme (QoC): t_s (ms)

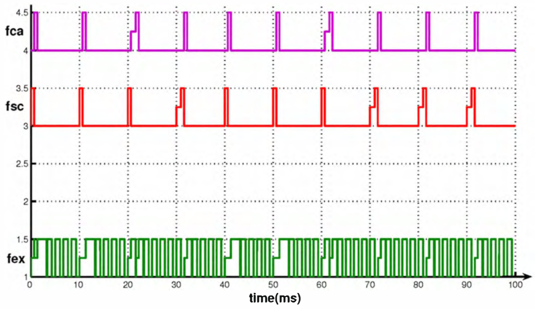

Figure 8. hp scheme ($URF = 70.4\%$, $Pr\_th = 0.9P_{\max}$): time exchanges

values of $\bar{D}$ and σ whatever the value of URF is. This is a consequence of the fact that (*cf.* remark in the study condition) the two frames of fsc and fca, during each sampling period, can be sent during the period of fex, which is not the case with $URF >$ 70.4 % where $\bar{D}$ and σ increase with the value of URF (see in table 2 $URF = 80\,\%, 89.6\,\%, 99.2\,\%$).

We explain the difference ($URF \leq 70.4\ \%$ and $URF > 70.4\ \%$) by means of two exchange temporal diagrams provided by TrueTime (figures 8 and 9 for the case of $Pr\_th = 0.9P_{\max}$). On the figure 8, we see that the frames fsc or fca can be delayed, during a sampling period, for the duration of one frame of fex (0.96 ms). On the figure 9, we see that the two frames of fsc and fca can be delayed and the delays for the frame of fca can be more than the duration of one frame of fex.

Note then, when $URF > 70.4\%$ and for increasing values of URF, $\bar{D}$ increases because the network load increases (thus more chances to delay the frames of fsc and fca).

- For increasing values of Pr_th, $\bar{D}$ also increases because the dynamic priorities of the frames of fsc and fca have fewer chances of being higher (except at the beginning of a transient behavior) than the threshold.

- Concerning the values of σ, we have the following comments:
  For each value of URF, the variation of σ, when Pr_th increases, presents a maximum (which occurs for a value of Pr_th around $Pr\_th = 0.5P_{\max}$). The explanation is given by means of the figures 10, 11, 12 (which represent the dynamic priority variation for $Pr\_th = 0.25P_{\max}, Pr\_th = 0.5P_{\max}$ and $Pr\_th = 0.9P_{\max}$). These figures allow us to evaluate the number of times where, during the t_s, the frames of fca have a higher or lower priority than the threshold (a higher priority means a lower delay; a lower priority means a bigger delay). Then we can see that we have for $Pr\_th = 0.5P_{\max}$, the maximum value of σ (the number of times where the dynamic priorities are higher than the threshold $\approx$ the number of times where the dynamic priorities are lower than the threshold). For $Pr\_th = 0.25P_{\max}$ ($Pr\_th = 0.9P_{\max}$), the number of times where the dynamic priorities are higher (lower) than the threshold is much greater than the number of times where the dynamic priorities are lower (higher) than the threshold. Thus, we have values of σ smaller than with $Pr\_th = 0.5P_{\max}$ (in the case of $Pr\_th = 0.25P_{\max}$ with a small value of $\bar{D}$; in the case of $Pr\_th = 0.9P_{\max}$ with a higher value of $\bar{D}$).

  Obviously, for each value of Pr_th, σ increases with URF (the reason is still the increase of the network load).

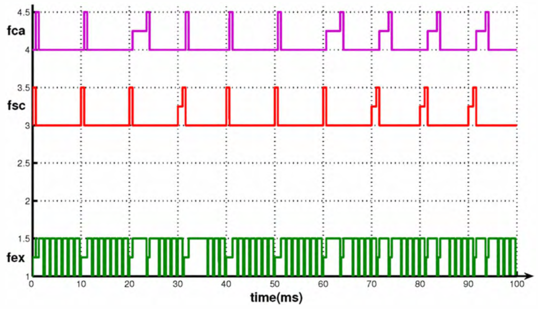

Figure 9. hp scheme ($URF = 89.6\%$, $Pr\_th = 0.9P_{\max}$)): time exchanges

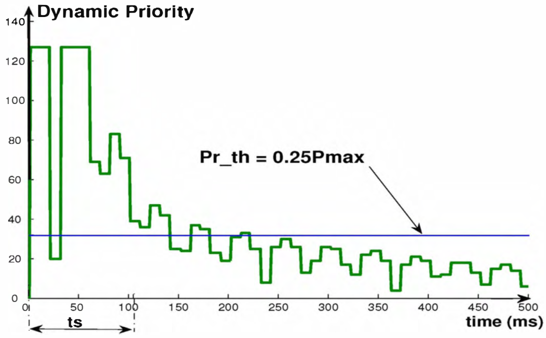

Figure 10. hp scheme, $URF = 99.2\%$, $Pr\_th = 0.25P_{\max}$

Important remark: for $Pr\_th \leq 0.15P_{\max}$ *i.e.* low threshold (we have not represented the results for reasons of limited space), we have the minimal value for $\bar{D}$ (1.28

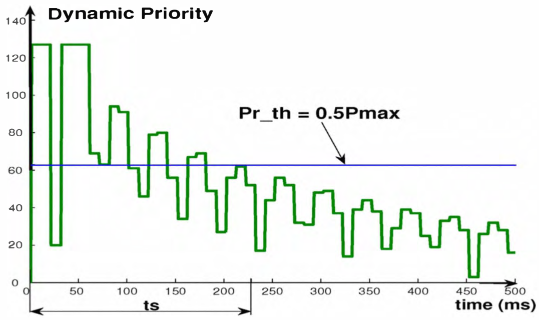

Figure 11. hp scheme, $URF = 99.2\%$, $Pr\_th = 0.5P_{\max}$

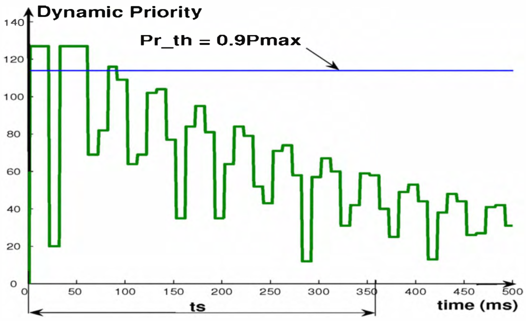

Figure 12. hp scheme, $URF = 99.2\%$, $Pr\_th = 0.9P_{\max}$

ms *i.e.* a frame of fsc (0.64 ms) and then a frame of fca (0.64 ms) that always use the medium before the frames of fex because the dynamic priority is always higher than Pr_th during the settling time). Then, of course, $\sigma = 0$.

3.3. hp+sts scheme

For the hp scheme, we give $\bar{D}$ and σ in table 4 and t_s in table 5. The values are obviously function of URF and Pr_th.

	Pr_th					
URF	$0.9P_{\max}$		$0.5P_{\max}$		$0.25P_{\max}$	
(%)	$\bar{D}$	σ	$\bar{D}$	σ	$\bar{D}$	σ
99.2	2.589	2.138	2.589	2.138	1.28	0.0
89.6	1.856	1.152	1.856	1.152	1.28	0.0
80.0	1.664	0.768	1.664	0.768	1.28	0.0
70.4	1.28	0.0	1.28	0.0	1.28	0.0
51.2	1.28	0.0	1.28	0.0	1.28	0.0
22.4	1.28	0.0	1.28	0.0	1.28	0.0

Table 4. (hp+sts) scheme (QoS): $\bar{D}$ and σ (ms)

We can see important differences with the hp scheme:

- for $URF \leq 70.4$ %, $\bar{D}$ is now always constant, whatever the Pr_th is (this is for two reasons: the first one is because of the consequence of the property ($URF \leq 70.4$ %) indicated in the subsection 3.1; the second is the fact that now, at the beginning of the transient behavior, the dynamic priority is used by the flows fsc and fca for a duration, at least, equal to $4h$. Obviously as $\bar{D}$ is constant, $\sigma = 0$.

URF	Pr_th		
(%)	$0.9P_{\max}$	$0.5P_{\max}$	$0.25P_{\max}$
99.2	103	103	50
89.6	100	100	50
80.0	98	98	50
70.4	50	50	50
51.2	50	50	50
22.4	50	50	50

Table 5. (hp+sts) scheme (QoC): t_s (ms)

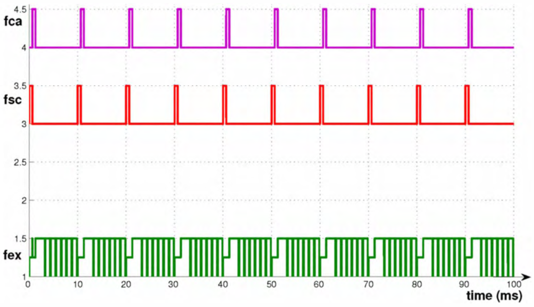

Figure 13. (hp+sts) scheme ($URF = 99.2\%$, $Pr\_th = 0.25P_{\max}$): time exchanges

- for $Pr\_th = 0.25P_{\max}$, we have $\bar{D}$ which is constant for all URF values (this means that, on all the network load conditions, the dynamic priority is higher than the threshold). The explanation is given by the exchange temporal diagram in figure 13.

- analysis of a row of the table 4 (in the case where $Pr\_th > 0.25P_{\max}$): we have the same values of $\bar{D}$ and σ whatever the value of Pr_th. The explanation is given by the exchange temporal diagrams of the figures 14 and 16 where we consider $URF = 99.2$ %. These diagrams are identical.

- analysis of a column of table 4 (in the case where $URF > 70.4$ %): we note an increase of $\bar{D}$ and σ with URF (the explanation is given by the figures 15 and 16); the delay of the frame fca (sampling periods 8 and 9) in the figure 16 is higher than in figure 15).

With respect to the hp scheme, all the improvements (which give best settling time for the process control application) result from the fact that the dynamic priority $P_{\max}$ is used a longer time. In figure 17, we have an example of the evolution of the dynamic priority (we have $P_{\max}$ during $8h$), compare the figure 17 with the figure 11.

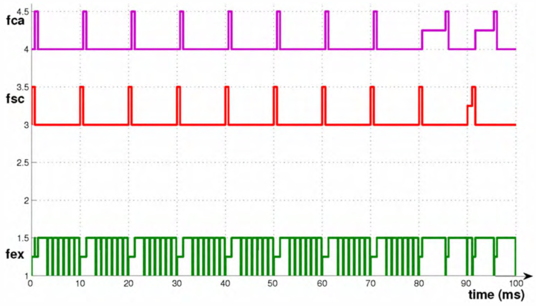

Figure 14. (hp+sts) scheme ($URF = 99.2\%$, $Pr\_th = 0.5P_{\max}$): time exchanges

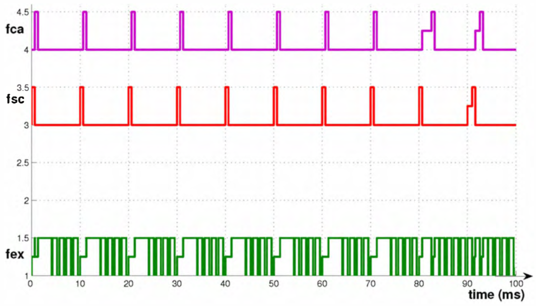

Figure 15. (hp+sts) scheme ($URF = 80\%$, $Pr\_th = 0.9P_{\max}$): time exchanges

3.4. hp+dts scheme

We give, as for the previous schemes $\bar{D}$ and σ in table 6 and t_s in table 7.

We can see now that we always have the minimum constant value $\bar{D}$ (duration of the fsc frame (0.64 ms) + duration of the fca frame (0.64 ms)), then $\sigma = 0$, and the best settling time (50 ms). This is a consequence of the fact that the dynamic priority is continuously controlled (by the control signal u) and that it is higher than the threshold for a time longer than the t_s (figure 18) .

	Pr_th					
URF	$0.9P_{\max}$		$0.5P_{\max}$		$0.25P_{\max}$	
(%)	D	σ	D	σ	D	σ
99.2	1.28	0.0	1.28	0.0	1.28	0.0
89.6	1.28	0.0	1.28	0.0	1.28	0.0
80.0	1.28	0.0	1.28	0.0	1.28	0.0
70.4	1.28	0.0	1.28	0.0	1.28	0.0
51.2	1.28	0.0	1.28	0.0	1.28	0.0
22.4	1.28	0.0	1.28	0.0	1.28	0.0

Table 6. (hp+dts) scheme (QoS): $\bar{D}$ and σ (ms)

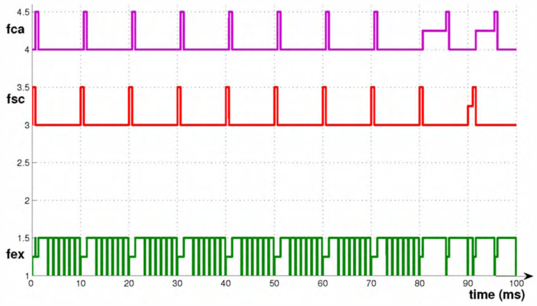

Figure 16. (hp+sts) scheme ($URF = 99.2\%$, $Pr\_th = 0.9P_{\max}$): time exchanges

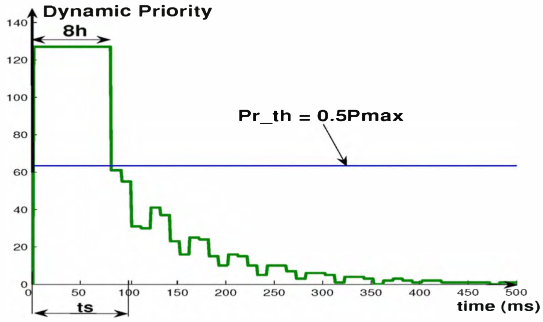

Figure 17. (hp+sts) scheme ($URF = 99.2\%$, $Pr\_th = 0.5P_{\max}$): time exchanges

URF	Pr_th		
(%)	$0.9P_{\max}$	$0.5P_{\max}$	$0.25P_{\max}$
99.2	50	50	50
89.6	50	50	50
80.0	50	50	50
70.4	50	50	50
51.2	50	50	50
22.4	50	50	50

Table 7. (hp+dts) scheme (QoC): t_s (ms)

3.5. QoC visualization

We represent, in figures 19 and 20 and 21, the step response for the three schemes in the following conditions $URF = 99.2\%$ and $Pr\_th = 0.9P_{\max}$. The oscillatory transient behavior clearly shows well the performances of the three schemes in terms of overshoot. The conditions of a big network load and a high threshold show the interest of the two schemes with a time strategy (hp+sts, hp+dts) to get good performances. The dynamic aspect of the time strategy in the scheme hp+dts shows in the end that it is the best scheme. The performances obtained with this message scheduling strategy are similar to those obtained without the network. These figures demonstrate the interest of these schemes with respect to the static scheme (see the figure 3).

4. Conclusion

This study has presented the interest of an hybrid priority strategy for the message scheduling on a network where we have two distributed applications with different

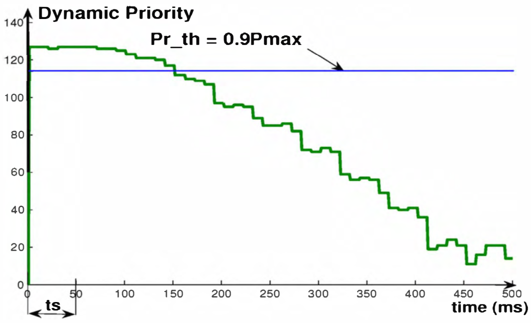

Figure 18. (hp+dts) scheme, $URF = 99.2\%$, $Pr\_th = 0.9P_{\max}$

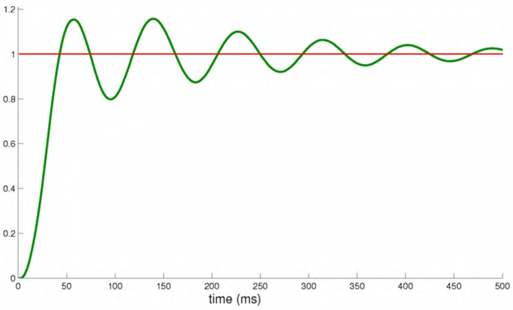

Figure 19. hp scheme: step response ($t_s = 359$ ms, overshoot about 20%)

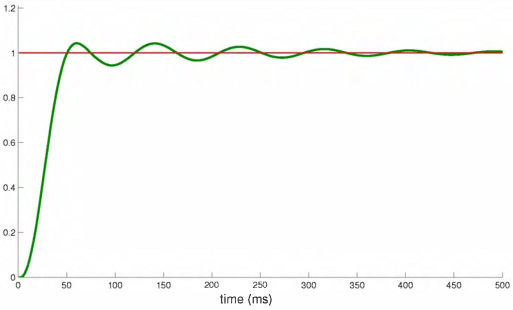

Figure 20. hp+sts scheme: step response ($t_s = 103$ ms, overshoot less than 5%)

needs in terms of transmission urgency in their messages flows (variable needs for the process control application). In particular, an important characteristic in a NCS context is the capacity to implement process control applications with good performances whatever the network load is. We have precisely shown that message scheduling strategies, based on hybrid priority schemes, allow the implementation of a distributed process control application even if the network load is important. We have considered three hybrid priority schemes and we have demonstrated the particular interest of a scheme, call (hp+dts), with a double

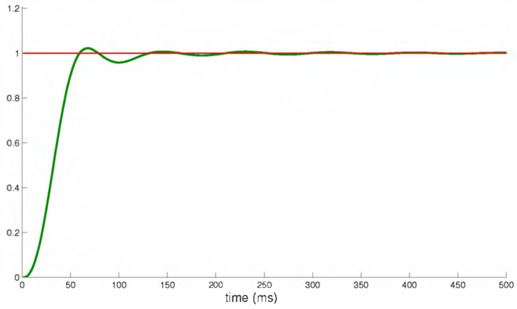

Figure 21. hp+dts scheme: step response ($t_s = 50$ ms, overshoot less than 1%)

aspect: dynamic priority based on a temporal supervision function of the control signal of the process control application and a reference profile. We have also evaluated, on the one hand, the QoS in terms of the mean delay and its standard deviation, and, on the other hand, the QoC in terms of settling time at 5%, and the relation between QoS and QoC (overshoot, damping).

References

[1] K. J. Astrom and B. Wittenmark, *Computer-controlled systems – Theory and design*, International Edition – Prentice Hall, 1997.

[2] W. Zhang, M. S. Branicky, and S. M. Philipps, “Stability of networked control systems”, *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, February 2001.

[3] G. Walsh and H. Ye, “Scheduling of networked control systems”, *Control Systems Magazine, IEEE*, vol. 21, no. 1, pp. 57–65, Feb 2001.

[4] Bosch GmbH, *CAN specification 2.0 (A)*, www.semiconductors.bosch.de/pdf/can2spec.pdf, 1991.

[5] G. Juanole, G. Mouney, and C. Calmettes, “On different priority schemes for the message scheduling in Networked Control Systems: Definition and Analysis of a Hybrid Priority Scheme for the Message Scheduling”, in *Proceedings of the 16^{th} Mediterranean Conference on Control and Automation, MED’08, Ajaccio, Corsica, France*, June 2008.

[6] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 – Reference Manual*, Lund Institute of Technology, Sweden, January 2007.

[7] K. Zuberi and K. Shin, “Scheduling messages on controller area network for real-time CIM applications”, *IEEE Transactions On Robotics And Automation*, vol. 13, no. 2, pp. 310–314, 1997.

[8] P. Marti, J. Yepez, M. Velasco, R. Villa, and J. Fuertes, “Managing quality-of-control in network-based control systems by controller and message scheduling co-design”, *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 6, pp. 1159–1167, Dec. 2004.

[9] G. Juanole and G. Mouney, “Networked Control Systems: Definition an Analysis of a Hybrid Priority Scheme for the Message Scheduling”, in *Proceedings of the 13^{th} IEEE conference on Embedded and Real-Time Computing Systems and Applications (RTCSA2007), Daegu, Korea*, August 2007.