MULTI-MASTER PROFIBUS DP MODELLING
AND WORST CASE ANALYSIS-BASED EVALUATION

Salvatore Cavalieri[1], Salvatore Monforte[1], Eduardo Tovar[2], Francisco Vasques[3]

[1] *Department of Computer Science and Telecommunications Engineering,*
Faculty of Engineering, University of Catania
Viale A.Doria, 6 95125 Catania, Italy
Tel.: +39 095 738 2362, Fax: +39 095 738 2397
e-mail: {salvatore.cavalieri, salvatore.monforte}@diit.unict.it

[2] *Department of Computer Engineering, ISEP, Polytechnic Institute of Porto*
Rua de São Tomé, 4200 Porto, Portugal
Tel.: +351.2.8340500, Fax: +351.2.8321159
e-mail: emt@dei.isep.ipp.pt

[3] *Department of Mechanical Engineering, FEUP, University of Porto*
Rua Dr.Roberto Frias, 4200-465 Porto, Portugal
e-mail: vasques@fe.up.pt

Abstract: This paper provides an analysis of the real-time behaviour of the multi-master Profibus DP network. The analysis is based on the evaluation of the worst-case message response time and the results obtained are compared with those present in literature, pointing out its capability to perform a more accurate evaluation of the performance of the Profibus network. *Copyright © 2002 IFAC*

Keywords: Fieldbuses, Profibus, Real-Time Systems, Worst Case Response Time.

1. INTRODUCTION

Fieldbuses are serial digital communication systems used to interconnect process controllers, sensors and actuators, at the lower levels of the factory automation hierarchy. These hierarchical levels have dissimilar message flows, especially in terms of time constraints (Decotignie and Pleinevaux, 1993). Time constraints are more stringent as we go down in the automation hierarchy. In the context of this paper, we consider time constraints or deadlines as the maximum allowable time span between a message transfer request and its transmission on the communication system.

The presence of a shared communication medium in Fieldbus systems introduce an increase in the total message delay with the respect of the traditional point-to-point communication systems. This increase results from multiple factors, such as the access and queuing delays and the protocol processing time.

In particular the access delay and queuing delay factors depend on the Medium Access Control (MAC) mechanism used by the Fieldbus network. Different approaches for the MAC mechanism have been adopted by Fieldbus communication systems. As significant examples, the Timed Token protocol in Profibus (EN 50170, 1996), the centralised polling in FIP (EN 50170, 1996) and the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) in CAN (EN 50325, 1999), can be mentioned.

Recently, several studies on the ability of Fieldbus networks to cope with real-time requirements have been presented. One of this is (Tovar and Vasques, 1999), where the authors addressed the Profibus MAC ability to schedule messages according to their real-time requirements, in order to support real-time distributed applications. A worst-case response time

analysis has been done, allowing defining analytical conditions to guarantee the respect of the high priority time constraints. This work may be surely considered a first step forward the pre-run-time schedulability analysis of Profibus networks. However it presents some limitations. First of all the analysis is based on the assumption that each master station is able to transmit only one message per token visit. This assumption is correct but seems too pessimistic. The other limitation is due to the value of the real token rotation time, i.e. the effective time needed to the token to perform a complete cycle among all the nodes in the Profibus network. It was assumed that this value was always equal to the worst one for every token cycle. Also in this case this assumption is correct, as it is relevant to a lower bound of the rotation time, but it seems a very pessimistic assumption and not much realistic. Finally, in the message model adopted in (Tovar and Vasques, 1999) some kind of message streams generated by stations are not represented, as will be pointed out in this paper.

Overcoming these limitations allows obtaining a worst-case response time analysis less pessimistic than that presented in (Tovar and Vasques, 1999), able to take into account more realistic information flow scenarios.

In (Monforte, et al., 2000) some of these limitations have been overcome, but the analysis presented considered only mono-master Profibus communication systems. In this paper the analysis is extended to the more general multi-master scenario.

2. A BRIEF OVERVIEW OF THE PROFIBUS PROTOCOL

Profibus was recently considered as one of the Fieldbus solutions of the General-Purpose Fieldbus Communication System European Standard, the EN 50170 (EN 50170, 1996).

The Profibus is based on a token passing procedure used by master stations to grant the bus access to each one of them. After receiving the token, the measurement of the token rotation time begins. This measurement expires at the next token arrival and results in the real token rotation time (T_{RR}). A target token rotation time (T_{TR}) must be defined in a Profibus network. The value of this parameter is common to all masters. When a master station receives the token, the token holding time (T_{TH}) timer is given the value corresponding to the difference, if positive, between T_{TR} and T_{RR}.

An important Profibus concept is the message cycle. A message cycle consists of a master's action frame (request or send/request frame) and the associated responder's acknowledgement or response frame. User data may be transmitted in the action frame or in the response frame. Profibus defines two categories of message cycles: high priority and low priority. These two categories use two independent outgoing queues.

If at the arrival, the token is delayed, that is, the real rotation time (T_{RR}) was greater than the target rotation time (T_{TR}), the master station may execute, at most, one high priority message cycle. Otherwise, the master station may execute high priority message cycles while $T_{TH}>0$. T_{TH} is always tested at the beginning of the message cycle execution. This means that once a message cycle is started it is always completed, including any required retries, even if T_{TH} expires during the execution. We denote this occurrence as a T_{TH} overrun. The low priority message cycles are executed if there are no high priority messages pending, and while $T_{TH}>0$ (also evaluated at the start of the message cycle execution, thus leading to a possible overrun of T_{TH}).

All the stations, except the token holder (initiator), shall in general monitor all requests. The acknowledgement or response must arrive within a predefined time, the slot time, otherwise the initiator repeats the request. At the network set-up phase, the maximum number of retries, before a communication error report, must be defined in all master stations.

The Profibus real-time analysis presented in this paper is based on the knowledge of the message cycle duration. This duration must include the time needed to transmit the action frame and receive the related response, and also the time needed to perform the allowed number of message retries.

3. NETWORK AND MESSAGE MODELS

We consider a bus topology containing n master stations. A special frame (the token) circulates around the logical ring formed by the masters. We denote the logical ring latency, i.e. the token walk time, including node latency delay, media propagation delay, etc. as λ.

We assume that two kinds of message cycles are present: high and cyclic low priority, both modelled as periodic processes. Requests for message cycles are placed in high priority and cyclic low-priority outgoing queues.

Let $Sh^k_i=(Ch^k_i, Dh^k_i, Th^k_i)$ and $Sl^k_i=(Cl^k_i, Dl^k_i, Tl^k_i)$ be high-priority and cyclic low-priority message streams in the generic master k (k=1,..., n), respectively. A message stream is a temporal sequence of requests for message cycles.

Ch^k_i and Cl^k_i are maximum message cycle duration for a request of message stream Sh^k_i and Sl^k_i, respectively. This duration includes the time needed to transmit the request frame and completely receive the related response, and also the time needed to perform the allowed number of message retries.

Th^k_i and Tl^k_i are the periodicity of streams Sh^k_i and Sl^k_i, respectively. We assume that this periodicity is the minimum interval between two consecutive arrivals of the related requests to the outgoing queue.

Dh^k_i and Dl^k_i are the relative deadlines of the related message cycles. They are defined as the maximum admissible time interval between the instant when the message request is placed in the outgoing queue and the instant when the related response is completely received at the master's incoming queue. In the paper, it will be assumed that $Dh^k_i= Th^k_i$ and $Dl^k_i=Tl^k_i$, as done in (Monforte, *et al.*, 2000). Further, we assume

that nh^k and nl^k are the number of high priority and cyclic low priority message streams, respectively.

In the following analysis we will consider the maximum message cycle duration for a request of message stream Sh^k_i for every message stream inside the master k (i=1,.., nh^k and k=1,..., n). This value will be indicated as Ch^k_{max}. Similarly, Cl^k_{max} will indicate the maximum message cycle duration for a request of message stream Sl^k_i for the generic master k and for every cyclic low priority message stream inside the master k (i=1,.., nl^k and k=1,..., n).

It's important to point out that the message model here proposed seems more realistic and accurate than that presented in (Tovar and Vasques, 1999), where no production period was associated to the low priority information flow. Moreover, in (Tovar and Vasques, 1999) the maximum message cycle duration for a request of high and low priority message streams was considered for every master and for every message stream inside the master. According to the model presented in this paper, maximum message cycle durations are considered separately for each master, as said before. In this way, it's possible to take into account different behaviour of the masters from the length of the message point of view.

4. TIMING ANALYSIS OF MULTI-MASTER PROFIBUS SYSTEMS

The aim of this section is to give definitions and theorems used for the computation of the worst-case response time.

Definition 1 – Profibus Critical Instant – Considering that requests for all high and low-priority message streams are simultaneously placed on the respective outgoing queues, we define a Profibus critical instant, as the time instant at which a request for a given message stream has the longest response time.

Definition 2 – Initial Blocking – We define the initial blocking as the delay that the first request made at the critical instant may suffer until starting to be processed.

Definition 3 – Critical Load – We define the critical load for a given priority class, as the time interval between a critical instant and the time instant when the last request (made at the critical instant) for that priority class has been completely processed.

Assuming that the critical instant for master k occurs immediately after the token has been released for its tc^{th} token visit (or token cycle), let us find the conditions that lead to the maximum initial blocking.

It should be noted that an increase of the initial blocking is attended only by the token holding periods of the masters following master k in the tc^{th} token cycle and up to master k in the $(tc+1)^{th}$ one.

For simplification of the analysis and without loss of generality it is possible to consider that the master k is the first one receiving the token at the tc^{th} token visit. Thus, let us introduce the following:

Definition 4 – Relative Token Cycle – We define the tc^{th} relative token cycle for the master k as the set of all the masters from master k (included) up to master n receiving the token in the tc^{th} absolute token visit and all the masters, from the first one up to the master which precedes master k, receiving the token in the $(tc+1)^{th}$ absolute token visit. Moreover we denote with k_i the ith master in a relative token cycle, where i=0,.., n-1, assuming that master k_0 coincides with master k.

Henceforth, unless explicitly specified, token cycles are to be considered relative to master k and master indexes are ordered starting from master k.

Moreover, the following notation is used:

- $nh\pi_{tc}^k$ as the number of high-priority messages processed by master k in its tc^{th} token cycle;
- $nl\pi_{tc}^k$ as the number of low-priority messages processed by master k in its tc^{th} token cycle;
- Δh_{tc}^k as the value of the token holding time timer at token arrival to master k for its tc^{th} token cycle;
- Δl_{tc}^k as the value of the time interval given by the difference between the token holding time timer at token arrival to master k for its tc^{th} token cycle and the number of high-priority messages processed at the tc^{th} token cycle, i.e. $\Delta l_{tc}^k = \Delta h_{tc}^k - nh\pi_{tc}^k \cdot Ch^k_{max}$;
- H_{tc}^k as the value of the token holding period at master k, which can be greater than Δh_{tc}^k, if an overrun occurs.
- $Cl_{max} = \max_{1 \le k \le n} \left\{ Cl^k_{max} \right\}$ the maximum low priority message cycle to be processed in the system.

For this analysis we consider the following theorem.

Theorem 1 – In a multi-master Profibus system, the maximum initial blocking follows as a result from the simultaneous occurrence of the following conditions:

- each master k_i (i=1,..,n-1) does not use the token for processing message cycles in the $(tc-1)^{th}$ relative token cycle and the master k does not use the token for processing message cycles in the tc^{th} token cycle. That is $\sum_{i=1}^{n-1} H_{tc-1}^{k_i} = 0 \wedge H_{tc}^k = 0$;

- a critical instant occurs just after releasing the token for every master k_i (i=1,.., n-1) in the $(tc-1)^{th}$ token cycle and for master k in the tc^{th} token cycle.

Proof. Let us assume that no message streams are to be processed by all the masters k_i following master k (i.e. i=1,..,n-1), when each of them releases the token in the $(tc-1)^{th}$ token cycle. This condition is depicted in Figure 1, in which it was assumed that master k is master 2 (i.e. k_0=2), k_1=3, k_2=4 and k_3=1. The figure highlights the $(tc-1)^{th}$ relative token cycle. The same figure allows better understanding the maximum value that the initial blocking can assume. As can be seen, a critical instant occurs for every master k_i (i=1, 2 and 3) just after releasing the token in the $(tc-1)^{th}$ token cycle and for master 2 in the tc^{th} token cycle. In this token cycle, on the receipt of the token, master 3 uses the token until it expires. Assuming that the expiration of T_{TH} occurs an instant after the master has started to transmit the longest message stream, it performs this last transmission before releasing the token. In the worst case, the time needed to complete this last transmission is $C^{k_1}_{max} = \max\left\{Ch^{k_1}_{max}, Cl_{max}\right\}$,

i.e. the maximum message cycle length which should be processed by the master k_1.

The other masters (number 4 and 1) receiving the token in the tc^{th} token cycle are allowed to transmit only one high-priority message stream lasting $Ch_{max}^{k_2}$ and $Ch_{max}^{k_3}$, respectively. When the tc^{th} token cycle is concluded, master 2 receives the token and can transmit the high-priority message streams.

It is clear that if no message stream is to be processed by all the masters following master k this leads to the lowest possible initial blocking for the master k, B^k:

$$\lambda \le B^k \qquad (1)$$

About the upper bound, it should be considered the overruns due to processing of the longest message streams. Taking into account the conditions leading to the maximum initial blocking it is clear that at most only one overrun (T_{TH} expiration) may take place before master k processes its first request. In particular, the worst case initial blocking occurs when master k_1 overruns due to the maximum messages cycle. Therefore, the maximum initial blocking in a Profibus multi-master system can be bounded as:

$$\lambda \le B^k \le T_{TR} - \lambda + C_{max}^{k_1} + \sum_{i=2}^{n-1} Ch_{max}^{k_i} \qquad (2)$$

An expression for the maximum blocking the first request issued at the critical instant in master k may suffer can be expressed as:

$$B^k = \lambda + \sum_{i=1}^{n-1}\left(nh\pi_{tc}^{k_i} \cdot Ch_{max}^{k_i} + nl\pi_{tc}^{k_i} \cdot Cl_{max}\right) \qquad (3)$$

where:

$$nh\pi_{tc}^{k_j} = min\left(nh^{k_j}, \lfloor \Delta h_{tc}^{k_j} / Ch_{max}^{k_j} \rfloor + 1\right) \qquad (4)$$

represents the number of high-priority messages which can be processed by master k_i at its tc^{th} token visit and

$$nl\pi_{tc}^{k_i} = \begin{cases} 0 \text{ if } \Delta l_{tc}^{k_i} < 0 \\ min\left(nl^{k_i}, \lfloor \Delta l_{tc}^{k_i} / Cl_{max}^{k_i} \rfloor + 1\right) \text{ if } \Delta l_{tc}^{k_i} \ge 0 \end{cases} \qquad (5)$$

represents the number of low-priority messages which can be processed by master k_i at its tc^{th} token visit.

We have to prove that under the hypotheses of the theorem, (3) represents the maximum allowed value for the initial blocking. Let us assume that exists $\overline{B}^k > B^k$. Therefore, at least for a given j (j=1,..,n-1), it follows that $\overline{nh\pi_{tc}^{k_j}} > nh\pi_{tc}^{k_j}$ or $\overline{nl\pi_{tc}^{k_j}} > nl\pi_{tc}^{k_j}$.

Let assume that $\overline{nh\pi_{tc}^{k_j}} > nh\pi_{tc}^{k_j}$ occurs. As said before, $nh\pi_{tc}^{k_j}$ represents the number of high-priority message streams processed in master k_j within its tc^{th} token cycle. That is, the minimum between the number of requests issued at the critical instant which have not been processed yet and the number of requests, which can be processed having received a given token holding time ($\Delta h_{tc}^{k_j}$) including the

overrunning one, if any. Hence, if $\overline{nh\pi_{tc}^{k_j}} > nh\pi_{tc}^{k_j}$ this would mean that message streams have been either received while all nh^{k_j} requests were queued yielding a deadline violation, or processed after the completion of the overrunning one, yielding a protocol violation.

In the same way it is possible to demonstrate that $\overline{nl\pi_{tc}^{k_j}} > nl\pi_{tc}^{k_j}$ cannot occur. $\square$
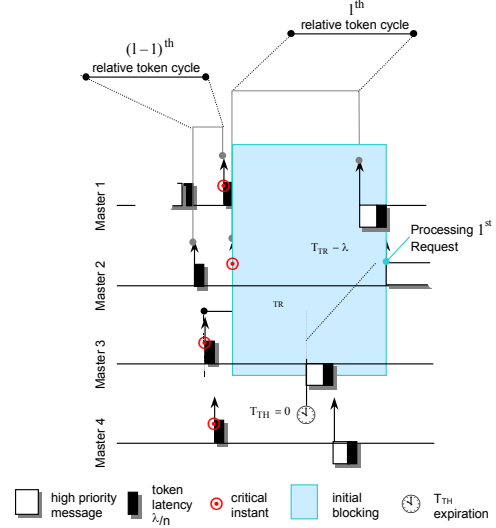

Fig. 1. Conditions leading to the maximum initial blocking.

5. WORST CASE RESPONSE TIME

Concerning the response time of a given message stream, two main components can be identified: the initial blocking and the critical load. It is clear that the worst-case response time occurs if both components assume their maximum value i.e. the worst-case delay the first request issued at the critical instant may suffer (maximum initial blocking) and the longest span of time between the critical instant and the instant at which the last request is processed (maximum critical load). Theorem 1, presented in the previous section, introduces the hypotheses under which in a multi-master Profibus network, the maximum initial blocking occurs. It should be noted that occurrence of such an event doesn't necessarily lead to the maximum critical-load and thus to the worst-case response time. It's possible to verify that if the following conditions occur:

$$\sum_{i=1}^{nh^{k_1}} Ch_i^{k_1} + nl^{k_1} \cdot Cl_{max} > T_{TR} - \lambda \qquad (6)$$

$$\sum_{j=1}^{nh^{k_i}} Ch_j^{k_j} + nl^{k_i} \cdot Cl_{max} > T_{TR} - \sum_{j=0}^{n-1} Ch_{max}^{k_j} - \lambda + i \cdot Ch_{max}^{k_i} \qquad (7)$$

where 1<i<n,

$$Th_j^{k_i} \le B + Ch_{max} + \sum_{i=2}^{n-1}(T_{TR} - \lambda + max\{0, Cl_{max} - Ch_{max}^{k_i}\}) \qquad (8)$$

where $0<i<n$ and $1 \le j \le nh^{k_i}$, the maximum initial blocking leads to the maximum critical load and thus to the worst-case response time evaluation.

In view of the notation previously introduced, the token holding period and the token holding time timer at token arrival to master k can be expressed as $H_{tc}^{k_i} = Ch_{max}^{k_i} \cdot nh\pi_{tc}^{k_i} + Cl_{max}^{k_i} \cdot nl\pi_{tc}^{k_i}$ and $\Delta h_{tc}^{k} = T_{TR} - \lambda - \sum_{i=0}^{n-1} H_{tc-1}^{k_i}$, respectively.

The worst-case response time for nh^k high-priority message streams could be expressed as:

$$R^k = B^k + \sum_{tc=1}^{m-1}\left(H_{tc}^k + I_{tc}^k\right) + H_m^k + (m-1)\cdot\lambda =$$
$$= B^k + \sum_{tc=1}^{m-1}\left(nh\pi_{tc}^k \cdot Ch_{max}^k + I_{tc}^k\right) + nh\pi_m^k \cdot Ch_{max}^k + (m-1)\cdot\lambda \quad (9)$$

where:

- $m = \min\left\{tc \in N : \sum_{i=1}^{tc} nh\pi_i^k \geq nh^k\right\}$ represents the minimum number of token cycles needed by master k to accomplish the processing of all the nh^k high-priority message streams;

- $I_{tc}^k = \sum_{i=1}^{n-1} H_{tc}^{k_i} = \sum_{i=1}^{n-1}\left(Ch_{max}^{k_i} \cdot nh\pi_{tc}^{k_i} + Cl_{max}^{k_i} \cdot nl\pi_{tc}^{k_i}\right)$ represents the interference the master k may suffer at next token arrival due to the message streams processed by all the other master is the tc^{th} token cycle.

Note that, while computing the worst-case response time for the nh^k requests issued in master k at the critical instant, no new request for such message streams can be issued, otherwise this would mean that a deadline was missed. Hence, for the formulation of $nh\pi_{tc}^k$ in the previous expression there is no need to consider the periodicity of message streams.

On the other hand, for all the masters following master k, the periodicity of the message streams should be taken into account. It should be noted that, new message stream requests may be released at master k_i (i=1,..., n-1) while computing the worst-case response time relative to master k. Hence, the number of high and low priority messages processed in master k_i (i=1,..., n-1) within its tc^{th} token visit, must be rewritten taking into account also the number of new requests which can be issued for each message stream. Therefore we have for the high-priority:

$$nh\pi_{tc}^{k_i} = \min\left(nh\rho_{tc}^{k_i} - \sum_{j=0}^{tc-1}nh\pi_j^{k_i}, \left\lfloor\frac{\Delta h_{tc}^{k_i}}{Ch_{max}^{k_i}}\right\rfloor^+ + 1\right) \quad (10)$$

where $nh\rho_{tc}^{k_i}$ represents the number of high-priority message streams issued in master k_i starting from the critical instant up to the tc^{th} token visit to that master; it's given by:

$$nh^{k_i} + \sum_{j=1}^{nh^{k_i}}\left\lfloor\frac{\lambda + \Theta_{tc} + \sum_{z=0}^{i-1}\left(H_l^{k_z} + \frac{\lambda}{n}\right) + nh\pi_{tc}^{k_i} \cdot Ch_{max}^{k_i}}{Th_j^{k_i}}\right\rfloor \quad (11)$$

Note that the expression $\lfloor x\rfloor^+$ in (10) is equal to $\lfloor x\rfloor$ if x≥0, and is equal to 0 if x<0.

In (11), Θ_{tc} is the span of time relative to master k between the critical instant and the end of its tc^{th} token holding period; it's given by:

$$\Theta_{tc} = B + \sum_{m=1}^{tc-1}\sum_{i=0}^{n-1} H_m^{k_i} + (tc-1)\cdot\lambda \quad (12)$$

It should be noted that there is a mutual dependence between the evaluation of $nh\pi_{tc}^{k_i}$ and $nh\rho_{tc}^{k_i}$. Thus, in order to evaluate $nh\pi_{tc}^{k_i}$ it is necessary to know the value of $nh\rho_{tc}^{k_i}$, and vice-versa.

For the low priority message stream, we have:

$$nl\pi_l^{k_i} = \begin{cases} 0 \text{ if } \Delta l_l^{k_i} < 0 \\ \min\left(nl\rho_l^{k_i} - \sum_{j=0}^{l-1} nl\pi_j^{k_i}, \left\lfloor\frac{\Delta l_l^{k_i}}{Cl_{max}^{k_i}}\right\rfloor + 1\right) \text{ if } \Delta l_l^{k_i} \geq 0 \end{cases} \quad (13)$$

and $nl\rho_{tc}^k$ is given by:

$$nl^{k_i} + \sum_{j=1}^{nl^{k_i}}\left\lfloor\frac{\lambda + \Theta_l + \sum_{z=0}^{i-1}\left(H_l^{k_z} + \frac{\lambda}{n}\right) + nh\pi_l^{k_i} \cdot Ch_{max}^{k_i} + nl\pi_l^{k_i} \cdot Cl_{max}^{k_i}}{Tl_j^{k_i}}\right\rfloor \quad (14)$$

6. NUMERICAL EXAMPLES AND FINAL REMARKS

The aim of this section is to compare the analysis presented in the paper with that proposed in (Tovar and Vasques, 1999). In that paper, Tovar and Vasques considered a scenario made up by six master stations featuring the time requirements shown in Table 1.

Table 1 Network Scenario

	Master 1	Master 2	Master 3	Master 4	Master 5	Master 6
Th^k_1	50 ms	90 ms	120 ms	60 ms	60 ms	80 ms
Th^k_2	100 ms	80 ms	130 ms	200 ms	100 ms	80 ms
Th^k_3	-	140 ms	110 ms	140 ms	100 ms	100 ms
nh^k	2	3	3	3	3	3

The maximum message length for both high and low priority message streams and for all the masters is equal to 2 ms (i.e. $Ch^k_{max}=Cl^k_{max}=2ms$, $\forall$ k). Using a 1-Mpbs network and assuming that request and response frames are made up by 400 bits, the frame duration is 400μs. Considering 260 μs for communication stack and propagation delay, each message cycle will take 660 μs. Configuring each master to support up to two message replies, we get the 2ms figure for the total length of the message cycle. T_{TR} was assumed equal to 8 ms, λ=0.1 ms and $Dl^k_i=Tl^k_i=T_{TR}$ $\forall$i, k.

It is important to point out the difficulties in the comparison of the approach here presented and that published in (Tovar and Vasques, 1999), due to the difference in the model of the low-priority traffic. The model here presented allows specifying the periodicity of the cyclic low priority traffic. The evaluation of the high priority worst case response time made in (Tovar and Vasques, 1999) is based on the assumption that low priority message streams are always present in the low priority-outgoing queue in each master. It's clear that this scenario can be obtained in our model assuming that the cyclic low priority message streams feature production periods less or equal to the T_{TR}. Table 2 shows the characterisation of the low priority information flow for each master, we have assumed in the comparison.

In order to have a more detailed evaluation of the scenario presented in (Tovar and Vasques, 1999), we analysed the responsiveness of the system on varying the target token rotation time around the 25% of its initial value, i.e. 8ms. Figure 2 shows the worst-case response time by means of the proposed analysis (R) and that presented in (Tovar and Vasques, 1999), SOA. The comparison is shown for masters 1 and 4, only for reason of space.

As shown in Figure 2, the computation of the worst-case response time by means of the proposed analysis (R) guarantees the minimum deadline (dotted line) and is slightly under the value (SOA). Moreover, it should be noted the expected graceful degradation of the system performance on increasing the T_{TR} as compared with the results of the approach presented in (Tovar and Vasques, 1999), where the higher the T_{TR} is the greater the increase of the response time leading to a violation of the minimum deadline.

Table 2 Network Scenario Considered

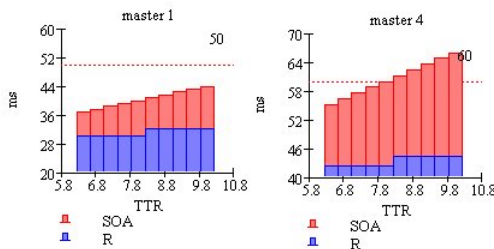|	Master 1	Master 2	Master 3	Master 4	Master 5	Master 6
Tl^k_1	200 ms	220 ms	200 ms	200 ms	200 ms	220 ms
Tl^k_2	210 ms	210 ms	220 ms	200 ms	220 ms	220 ms
Tl^k_3	220 ms	220 ms	220 ms	220 ms	220 ms	220 ms
nl^k	3	3	3	3	3	3

Fig. 2. Comparison between SOA and proposed analysis: scenario 1.

In order to point out the advantages of the proposed analysis, let us consider a different scenario, obtained varying the length of the low priority messages. Thus, maintaining all the hypotheses seen before, let us consider the value $Cl^k_{max}=7ms$, $\forall k$. Figure 3 depicts the results obtained by applying the proposed approach, comparing it with the results obtained according to the theory presented in (Tovar and Vasques, 1999). As can be seen, again the worst-case response time evaluation is less pessimistic. Moreover, it is very important to point out that the values obtained according with the proposal allow respecting the constrains on the minimum deadline.
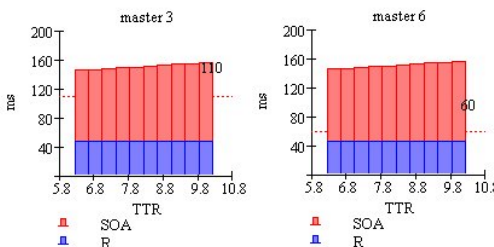

Fig. 3. Comparison between SOA and proposed analysis: scenario 2.

Finally let us consider the last scenario where only three masters are connected to the bus and $T_{TR}=12$ ms. Table 3, summarises the considered network scenario for this last example. The aim of this scenario is to demonstrate how high loaded masters will influence the response time of other masters. It should be noticed that the first master has just one message to be processed, whereas the other masters are slightly loaded if compared with the first one.

As depicted in Figure 4, also in this case the proposed analysis guarantee that all the masters meet their minimum deadline and highlight again as the approach (SOA) fails to satisfy the deadline constrains. Moreover there is no linear dependence on the value of T_{TR} as in the case of (Tovar and Vasques, 1999) where in spite of the real network load the higher the T_{TR}, the worst the response time evaluation.

Table 3 Considered Network Scenario

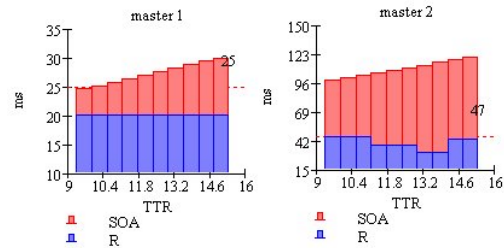|	Master 1	Master 2	Master 3
nh^k	1	4	3
Th^k_1	25 ms	50 ms	35 ms
Th^k_2	-	55 ms	36 ms
Th^k_3	-	47 ms	37 ms
Th^k_3	-	48 ms	-
Ch_{max}	5 ms	2 ms	2 ms
nl^k	1	1	1
Tl^k_1	110 ms	110 ms	110 ms
Cl_{max}	5 ms	5 ms	5 ms

Fig. 4. Comparison between SOA and proposed analysis: scenario 3.

REFERENCES

Decotignie, J.D. and P. Pleinevaux (1993). A Survey on Industrial Communication Networks. *Annales des Télécommunications*, pp. 435-448.

EN 50170 (1996). General Purpose Field Communication System. CENELEC.

EN 50325 (1999). Industrial Communication Subsystem based on ISO 11898 (CAN) for Controller-Device interfaces. Parts 1-2-3. CENELEC.

Tovar, E. and F. Vasques (1999). Real-Time Fieldbus Communications Using Profibus Networks. *IEEE Transaction on Industrial Electronics*. Volume 46 Number 6, pp.1241-1251.

Monforte, S., M. Alves and E. Tovar (2000). Designing Real-Time Systems Based on Mono-Master Profibus-DP Networks. In: *Proceedings of DCCS'2000*, IFAC, Sydney, pp.36-43.