Scheduling Real-Time Communications with P-NET

Eduardo TOVAR

Department of Computer Science
School of Engineering
Polytechnic Institute of Porto,
Rua de São Tomé, 4200 Porto, Portugal
Tel.: +351.2.8340500, Fax.: +351.2.821159
e-mail: emt@dei.isep.ipp.pt

Francisco VASQUES

Department of Mechanical Engineering
School of Engineering
University of Porto,
Rua dos Bragas, 4099 Porto CODEX, Portugal
Tel.: +351.2.2041774, Fax.: +351.2.319280
e-mail: vasques@fe.up.pt

***Abstract:** In this paper we address the P-NET Medium Access Control (MAC) ability to schedule traffic according to its real-time requirements, in order to support real-time distributed applications. We provide a schedulability analysis based on the P-NET standard, and propose mechanisms to overcome priority inversion problems resulting from the use of FIFO outgoing buffers.*

1. Introduction

Within industrial communication systems, fieldbus networks are specially devoted for the interconnection of process controllers, sensors and actuators, at the lower levels of the automation hierarchy, where fulfilment of time constraints is often a mandatory requirement.

In the context of this paper, we consider time constraints or deadlines, as the maximum delay between sending a request and receiving the related response at the application level. In other words, we are emphasising the association of deadlines to messages cycles (request followed by response at the application level).

The message cycle delay is made up of multiple factors, such as transmission time (frame length / transmission rate), protocol processing time, propagation delay or access and queuing delay. As we are dealing with real-time communication across a shared transmission medium, the most relevant factors to our analysis are the access and queuing delays, which heavily depend on the Medium Access Control (MAC) mechanism.

Different approaches for the MAC mechanism have been adopted by fieldbus communication systems. As significant examples, we can mention the timed token protocol in Profibus [1], the centralised polling in FIP [1], the CSMA/CA in CAN [2] and *Virtual Token Passing* in P-NET [1].

Recently, several studies on the ability of fieldbus networks to cope with real-time requirements have been presented, such as [3] on CAN, [4] and [5] on FIP and finally [6] and [7] on Profibus.

In this paper we address the P-NET's MAC ability to schedule field level transactions according to its real-time requirements, in order to support real-time distributed applications. The proposed P-NET pre-run-time schedulability analysis is based on the knowledge of the field level transactions timing requirements, expressed by means of message cycles length and deadline. We highlight the drawback imposed by the *First In First Out* (FIFO) behaviour of the network buffer, suggesting then the use of a priority based implementation of the network buffer.

2. P-NET MAC Description

P-NET is a multi-master standard based on a *Virtual Token Passing (VTP)* scheme, without explicit token transmission between masters.

Each master contains two counters. The first one, the Access Counter (AC), holds the node address of the currently transmitting master. When a request has been completed and the bus has been idle for 40 bit periods (520μs @ 76,8Kbps), each one of the AC counters is incremented by one. The master whose AC counter value equals its own unique node address is said to hold the token, and is allowed to access the bus. When the AC counter is incremented as it exceeds the "maximum No of Masters", the AC counter in each master is pre-set to one. This allows the first master in the cycling chain to gain access again.

The second counter, the Idle Bus Bit Period Counter (IBBPC), increments for each inactive bus bit period. Should any transactions occur, the counter is re-set to zero. As explained above, when the bus has been idle for 40 bit periods following a transfer, all AC counters are incremented by one, and the next master is thus allowed access.

If a master have nothing to transmit (or indeed isn't even present), the bus will continue inactive. Following a further period of 130μs (10 bit periods), the IBBPC will have reached 50, (60, 70,…) and all the AC counters will again be incremented, allowing the next master access. The virtual token passing will continue every 130μs, until a master does require access.

P-NET standard also stands that each master is only allowed to perform a message transaction per token "visit".

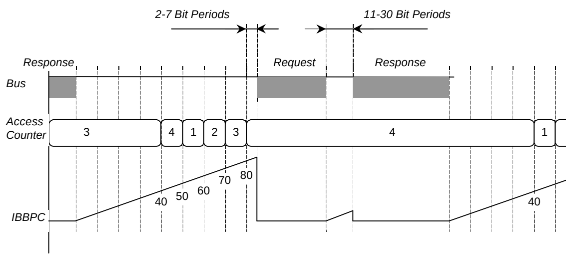

Figure 1

A slave is allowed to access the bus, between 11 and 30 bit periods after receiving a request, measured from the beginning of the stop bit in the last byte of the frame. The maximum allowed delay is then 390μs (corresponding to 30 bit periods).

If the IBBPC counter is higher than or equal to 360, the token master should send a normal frame or a *sync*. A *sync* is one byte that contains the node address of the token master. No device will receive the byte but all IBBPC counters will be cleared, thus resulting in AC counters synchronisation.

Figure 1 summarises these virtual token passing procedures.

3. P-NET Schedulability Analysis

In this section, we establish a pre-run-time schedulability condition for the P-NET fieldbus network. Essentially, we provide formulae to evaluate the minimum message deadline, as function of message lengths, number of different message streams and number of P-NET master stations.

Our pre-run-time schedulability analysis is based on the assumption that the inter-arrival time between two consecutive messages at the same message stream is longer than the deadline of that stream. This means that in the outgoing buffer there will not be two messages from the same stream.

3.1. Network and Message Models

A network is composed of *nm* master stations. Each *k* master station has associated $ns^{(k)}$ message streams, each one being a temporal sequence of message cycles (pair of messages constituted by a request and a response, when applicable), concerning, for instance, a specific process variable. A message stream is characterised as $S_i^{(k)} = (C_i^{(k)}, D_i^{(k)})$, where $C_i^{(k)}$ denotes the length of the message cycle (time for sending the request and receive the response) and $D_i^{(k)}$ denotes the relative deadline of the message. The message relative deadline is the maximum admissible time to deliver it. Additionally, we denote a P-NET bit period as *bp*.

3.2. Maximum Virtual Token Cycle

Our analysis is based on the knowledge of the maximum virtual token cycle time (*vtcycle*). This time is given by the sum of each station maximum token holding time:

$$vtcycle = \sum_{i=1}^{nm}\left(7\times bp + \max_{j=1..ns^{(i)}}\left(C_j^{(i)}\right) + 40\times bp\right) \quad (1)$$

where $7\times bp$ corresponds to the master reaction time and $40\times bp$ to the implicit token passing delay. The message cycle time $\max_{j=1..ns^{(i)}}\left(C_j^{(i)}\right)$ includes the request and response message lengths and the responder turn-around time.

3.3. *Deadline Constraint*

The standard stands that the master requests are passed to the network layer buffer, which behaves as a FIFO. Thus, in the worst case, the message cycle with the earliest deadline may be the last one to be transferred, that is, we may have a priority inversion with a length:

$$ns^{(k)} \times vtcycle \tag{2}$$

Thus, the P-NET traffic is schedulable, that is real-time requirements are met, if, and only if, at each station k we have:

$$\min_{l=1..ns^{(k)}} \left\{D_l^{(k)}\right\} \geq ns^{(k)} \times \sum_{i=1}^{nm} \left(47 \times bp + \max_{j=1..ns^{(i)}} \left\{C_j^{(i)}\right\}\right) \tag{3}$$

Thus, we may conclude that other queuing strategies, such as priority queues, rather than FIFOs would be advisable.

4. Outgoing Priority Queues

With FIFO outgoing queues, the schedulability condition very much depends on the length of priority inversions we may have. These priority inversions are as much important as more high priority flows are associated to a particular master station.

This shortcoming can be avoided if priority queuing is used instead of FIFO queuing. We propose the use of a deadline based priority algorithm, where the outgoing queue is dynamically managed being the highest priority message, the one with the earliest deadline.

Our P-NET high priority message streams ($Sh_i^{(k)}$) are characterised by the relative deadline ($Dh_i^{(k)}$) and time length ($Ch_i^{(k)}$) of their messages. If we assume their worst-case inter-arrival time, they can be converted into periodic arrivals.

It can be shown that for periodic messages, there exists a feasible schedule if and only if there exists a feasible schedule for the LCM (the least common multiple) of the periods [8]. Moreover, it can be shown that if the messages share a common request time, it is a schedulability sufficient condition that messages are schedulable for the longest period [9]. Thus we must analyse the schedulability of a high priority message streams set in a time span ($T_q^{(k)}$) corresponding to:

$$T_q^{(k)} = \max_i \left\{Dh_i^{(k)}\right\} \tag{4}$$

It results also from our approach that we should evaluate the minimum worst case number of token visits during $T_q^{(k)}$. Considering that in the worst case the token will visit the station k each *vtcycle*, the number of token visits during $T_q^{(k)}$ will be given by the following expression:

$$N_q^{(k)} = \left\lfloor \frac{T_q^{(k)}}{vtcycle} - 1 \right\rfloor \tag{5}$$

where the subtraction of one unit results from the worst case scenario where message cycles are passed to the outgoing buffer just after the token release by the station, and the floor function results from $T_q^{(k)}$ not being a multiple of *vtcycle*. Figure 2 illustrates these assumptions.

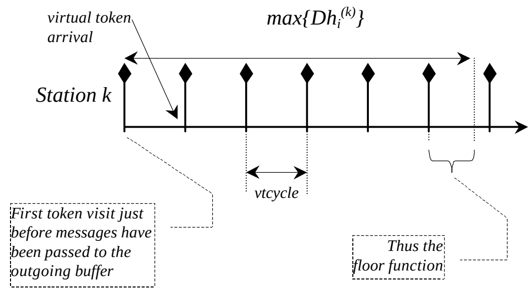

Figure 2

Conversely to the case of outgoing FIFO queues, it can be shown that if the number of message transactions requests during $T_q^{(k)}$ is limited to $N_q^{(k)}$, then using a earliest deadline first based algorithm, the message deadlines are guaranteed. In fact, if we define the concept of token use rate by a station k as being the percentage of token visits that the station uses to transmit one high priority message, analogies to the case of task scheduling in a monoprocessor environment [9] can be made.

In [9] Liu showed that for a given set of m tasks, the deadline driven scheduling algorithm is feasible if and only if:

$$\sum_{i=1}^{m} \frac{C_i}{T_i} \leq 1 \tag{6}$$

where C_i represents the processor running time of task i and T_i represents its periodicity. In other words, the set of tasks is schedulable if the processor utilisation factor (fraction of processor time used to process tasks) is bounded to 1 (in fact the actual maximum bound limit).

So, at a one high priority message per token visit basis, we can formulate a deadline constraint as follows:

$$\sum_{i=1}^{ns^{(k)}} \left\lceil \frac{T_q^{(k)}}{Dh_i^{(k)}} \right\rceil \leq N_q^{(k)}, \ \forall_{\ Master\ k} \qquad (7)$$

that is, the number of high priority message cycles transactions requests, in a station k, that arrive within the time corresponding to the longest deadline (period), should be lower than the worst case number of token visits during the same time.

Figure 3 illustrates an example of a station with 4 high priority message streams both using priority and FIFO queues.

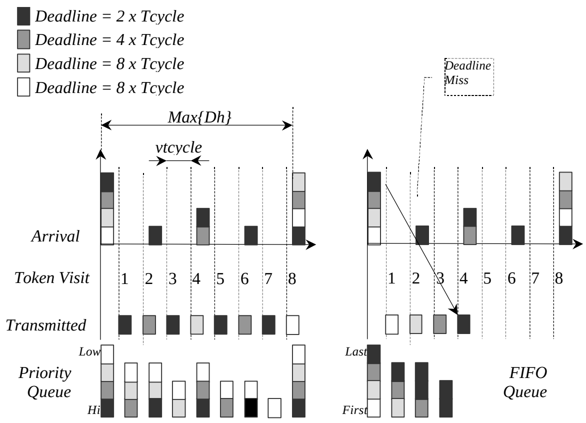

Figure 3

5. Conclusions

In this paper we provided basic pre-run-time schedulability conditions for supporting real-time communications with P-NET.

We propose the development of a deadline based priority mechanism for the outgoing buffer, as it will allow the support of real-time traffic with tighter deadlines.

References

[1] EN 50170, "General Purpose Field Communication System", European Standard, CENELEC, July 1996.

[2] SAE J1583, "Controller Area Network (CAN), an In-Vehicle Serial Communication Protocol", SAE Handbook, Vol. II, 1992.

[3] K. Tindell, H. Hansson and A. Wellings, "Analysing Real-Time Communications: Controller Area Network (CAN)", *in Proceedings of the IEEE Real Time Systems Symposium (RTSS'94)*, pp. 259-263, S.Juan, Puerto Rico, December 1994.

[4] P. Pedro and A. Burns, "Worst Case Response Time Analysis of Hard Real-Time Sporadic Traffic in FIP Networks", *in Proceedings of 9th Euromicro Workshop on Real-time Systems*, Toledo, Spain, June 1997.

[5] F. Vasques, "Sur L'Intégration de Mécanismes d'Ordonnancement et de Communication dans la Sous-Couche MAC de Réseaux Locaux Temps-Réel", *PhD Thesis, available as technical Report LAAS Nº 96229*, Toulouse, France, June 1996.

[6] E. Tovar and F. Vasques, "Guaranteeing Real-Time Message Deadlines in Profibus Networks", *in proceedings of the 10th Euromicro Workshop on Real-time Systems*, Berlin, Germany, June 1998.

[7] E. Tovar and F. Vasques, "Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications", *submitted to the 15th IFAC Workshop on Distributed Computer Control Systems (DDCS'98)*, Como, Italy, September 1998.

[8] E. Lawler and C. Martel, "Scheduling Periodically Occurring Tasks on Multiple Processors", *Information Processing Letters*, 12(1), February 1981.

[9] C. Liu and J. Layland, "Scheduling Algorithms for Multiprogramming in Hard-Real-Time Environment", *Journal of the ACM*, Vol. 20, No. 1, pp.46-61, January 1973.