A COMMUNICATION SUPPORT FOR REAL-TIME DISTRIBUTED COMPUTER CONTROLLED SYSTEMS

Eduardo Tovar, Francisco Vasques [1]

Department of Computer Engineering, Polytechnic Institute of Porto, Rua de São Tomé, 4200 Porto, Portugal, E-mail: emt@dei.isep.ipp.pt
[1] Department of Mechanical Engineering, University of Porto, Rua dos Bragas, 4099 Porto Codex, Portugal, E-mail: vasques@fe.up.pt

Keywords
P-NET, Real-Time Communications, Fieldbus Networks, Scheduling.

Abstract

In this paper, we analyse the ability of P-NET [1] fieldbus to cope with the timing requirements of a Distributed Computer Control System (DCCS), where messages associated to discrete events should be made available within a maximum bound time. The main objective of this work is to analyse how the network access and queueing delays, imposed by P-NET's virtual token Medium Access Control (MAC) mechanism, affect the real-time behaviour of the supported DCCS.

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 factory automation hierarchy (figure 1).

Among other characteristics, these hierarchical levels have dissimilar message flows. It is possible to classify [2] such flows, carried by the communication systems, according to:

- the required response time, that is, how quickly messages must be transferred;
- their length, that is, the amount of information to be transferred;
- the required reliability, which means, for instance, the importance of error-free or guaranteed delivery;
- the message rate, in other words, how frequently an application task sends a particular type of message, for instance, from a sensor to the process controller.

In a rough way, one can say that 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 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).

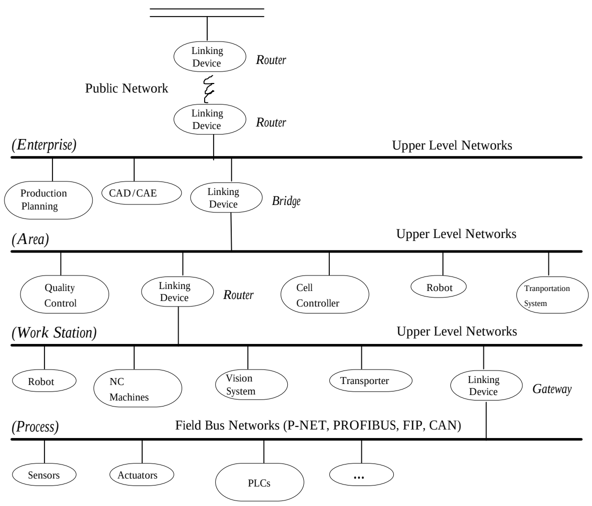

Figure 1: Automation Hierarchy and Communication Networks

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 queueing 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 queueing delays,

Proc. of the Fourth Workshop on Discrete Event Systems

which heavily depend on the *Medium Access Control* mechanism.

Different approaches for the Medium Access Control mechanism have been adopted by fieldbus communication systems. As significative examples, we can mention the timed token protocol in Profibus [1], the centralised polling in FIP [1] and the CSMA/CA in CAN [3].

Recently, several studies on the ability of fieldbus networks to cope with real-time requirements have been presented, such as [4] on CAN, [5] and [6] on FIP and finally [7] and [8] on Profibus.

In this paper, we analyse the ability of the P-NET [1] fieldbus network to cope with the timing requirements of a distributed computer control system (DCCS), where messages associated to discrete events should be made available within a maximum bound time.

2. P-NET Networks

2.1. MAC Characteristics

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 2 summarises these Virtual Token Passing procedures.

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).

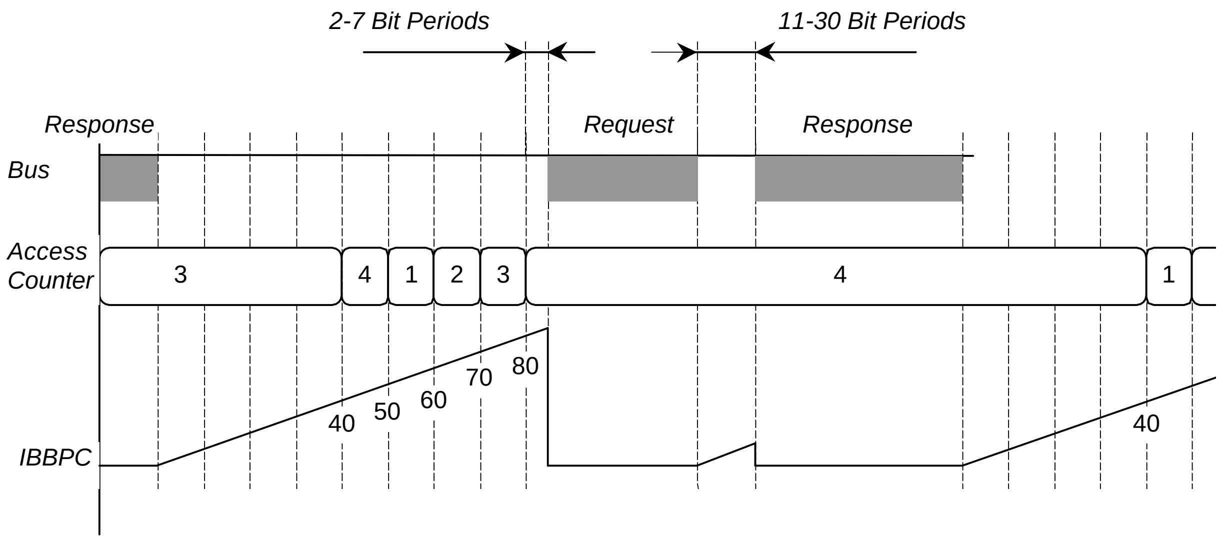

Figure 2: Virtual Token Passing Timings Example

3. Mapping DCCS Requirements into P-NET

Message types that can be found in Distributed Computer Controlled Systems (DCCS) include *command/response*, *alarms*, *logging messages*, *files* and *programs*. Typically a P-NET network will encompass the first three types of messages.

would implement master capabilities, whereas node 1 would implement slave capabilities.

The length of a communication frame is a significant parameter, as shorter frames avoid monopolisation of the shared transmission medium. In P-NET, frames are limited to 69 data bytes (with a 6 byte overhead, if we do not consider segmentation). In typical applications,

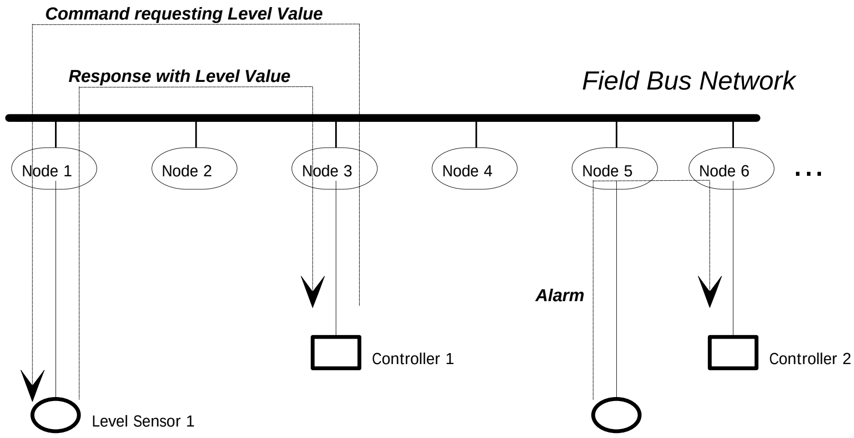

Figure 3: Command/Response and Alarm Transactions

A *command* is a message sent from one task to another (for example, from a controller to a sensor or actuator) requesting an action to be performed or a parameter be returned. A *response* is the reply confirming the performed action (write) or containing the requested information (read). These messages correspond to *STORE* and *LOAD* services. Both the command and response have, typically, small duration and normally require short deadlines. The command and the response form a single logical transaction, which in P-NET is reinforced by the fact that the requests have normally immediate response (exception is made if segmentation or fragmentation is used [9], which are not analysed in this paper).

Within P-NET, master stations should support entities needing to issue commands, whereas slave stations would normally support sensors and actuators.

Alarms are messages sent from sensors to process controllers, indicating error conditions. They normally require shorter deadlines and normally involve short quantity of data. As in P-NET the communication initiative is restricted to master nodes, alarm sensors should then be mapped into masters, to avoid unnecessary polling operations.

Figure 3 illustrates typical DCCS message transactions. In terms of P-NET, nodes 3, 5 and 6

data concerning commands, responses and alarms will range from 1 to 16 bytes, which is fully compatible with the P-NET protocol.

Application	**Scan Time (s)**	**Alarm Response Time (s)**
Electricity Generation	2.0	0.1
Oil	1.0	0.1
Chemical Industry	0.1-0.6	0.01-1.0
Steel Industry	0.5	0.01
Food Industry	0.1-0.6	1.0
Paper Industry	0.6-300	0.6
Telecommunications	0.2	1.0

Table 1: Typical Scan Times and Alarm Response Times in Different Applications

Finally, it is important to highlight some typical figures concerning applications required response times and number of I/O points [2,10], which may impact the real-time compliance of a fieldbus communication network.

Application	Inputs		Outputs	
	analogue	digital	analogue	digital
Electricity Generation	2000	6000	800	200
Oil	160	1800	100	1200
Chemical Industry	400	500	50	600
Steel Industry	100	500	50	100
Paper Industry	40	50	5	30

Table 2: Typical Number of I/O Points in Different Applications

4. P-NET Schedulability Analysis

In this section, we establish a pre-run-time shedulability 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.

4.1. 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 $Si^{(k)} = (Ci^{(k)}, Di^{(k)})$, where $Ci^{(k)}$ denotes the length of the message cycle (time for sending the request and receive the response) and $Di^{(k)}$ denotes the relative deadline of the message. The message relative deadline is the maximum admissible time to deliver it. Additionally, we denote a bit period as *bp*.

4.2. Maximum Virtual Token Cycle (*vtcycle*)

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 $7\times bp$ correspond 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.

4.3. 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:

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

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

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

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

5. Numerical Analysis

In this section we present some numerical examples using the proposed pre-run-time schedulability condition.

Consider a DCCS example constituted by 800 I/O points, each one corresponding to a single message stream. For simplification we also assume that for all *i* master stations, $\max_{j=1..ns^{(i)}}\left(C_j^{(i)}\right)$ is $200\times bp$, that is message length is bounded. Considering a $30\times bp$ maximum slave turn-around time and $90\times bp$ of non-information field bits, the $200\times bp$ value would correspond to about 71 useful information field bits (in P-NET each frame character has an associated 9th bit).

For this example, the pre-run-time schedulability condition is:

$$\min_{l=1..ns^{(k)}} \left\{D_l^{(k)}\right\} \geq ns^{(k)} \times nm \times 247 \times bp, \quad \forall_{k,k=1..nm} \qquad (4)$$

As a matter of fact, $247 \times bp$ is just an example of the maximum virtual token holding time (t_{mtht}). So, expression 4 can be generalised as:

$$\min_{l=1..ns^{(k)}} \left\{D_l^{(k)}\right\} \geq ns^{(k)} \times nm \times t_{mtht}, \quad \forall_{k,k=1..nm} \qquad (5)$$

Two further results can be inferred:

a) if the number of message streams by master is the same, then the pre-run-time schedulability condition may be simplified to:

$$\min_{l=1..ns^{(k)}} \left\{D_l^{(k)}\right\} \geq n_{points} \times t_{mtht}, \quad \forall_{k,k=1..nm} \qquad (6)$$

where n_{points} correspond to the total number of I/O points. This means that, in this case, the pre-run-time schedulability condition is independent of the number of master stations.

In the considered DCCS (800 points and with a token holding time of $247 \times bp$), deadlines should be greater than:

$$800 \times 247 \times bp = 800 \times 247 \times 1/76.800 = 2.57\text{s}$$

b) The deadline restriction (3) in one station depends on the number of streams in that station and also on the number of masters; this restriction does not depend on the number of streams located in the other nodes.

Table 3 and figure 4 illustrate the minimum admissible deadline as a function of the number of streams in that particular station and of the total number of masters.

6. Conclusion

In this paper we provide a comprehensive study on how to use P-NET fieldbus networks to support real-time communications. The major contribution of this paper is the integration of a pre-run-time schedulability analysis with the typical Distributed Computer Controlled System requirements.

From the above analysis we can also conclude that, in order to support more stringent DCCS real-time requirements, other queueing strategies, such as priority queues based on the earliest deadline scheduling algorithm, would be advisable within the P-NET protocol. In [10] the authors analise the worst case response time using

ns*^(k)** ***nm	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
80	257.3	514.6	771.9	1029.2	1286.5	1543.8	1801.0	2058.3	2315.6	2572.9
40	128.6	257.3	385.9	514.6	643.2	771.9	900.5	1029.2	1157.8	1286.5
20	64.3	128.6	193.0	257.3	321.6	385.9	450.3	514.6	578.9	643.2
10	32.2	64.3	96.5	128.6	160.8	193.0	225.1	257.3	289.5	321.6
5	16.1	32.2	48.2	64.3	80.4	96.5	112.6	128.6	144.7	160.8

Table 3: Minimum Deadline (in msec)

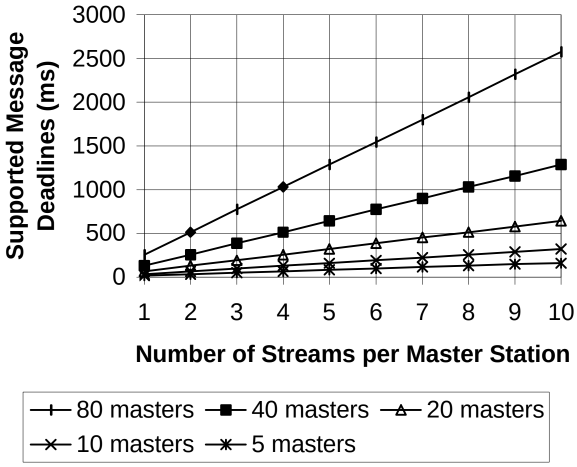

Figure 4: Minimum Deadline (msec)

a queuing stratagy based on the Earliest Deadline First algorithm.

Acknowledgements

This work was partially supported by FLAD, under project SISTER (Real-Time Sistems for Manufacturing Applications) 471/97 and by ISEP under de project REMETER (Real-Time Multi-point Networks).

The authors would also like to acnowledge the support from Chris Jenkins and John Johanssen from Procces Data A/S.

References

[1] EN 50170, "*General Purpose Field Communication System*", Vol. 1/3 (P-NET), CENELEC, December 1996.

[2] S. M. Prince, M. S. Soloman, "*Communication Requirements of a Distributed Computer Control System*", IEE Proceedings, Vol. 128, No. 1, pp. 21-34, January 1981.

[3] SAE J1583, "*Controller Area Network CAN, an In-Vehicle Serial Communication Protocol*", 1992-SAE-Handbook, pp.20341-20355.

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

[5] P. Pedro, 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.

[6] 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).

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

[8] E. Tovar, F. Vasques, "*Setting Target Toke Rotation Time in Profibus Based Distributed Applications*", in proceedings of the 15th IFAC Workshop on Distributed Computer Controlled Systems, Como, Italy, September 1998.

[9] E. Tovar, F. Vasques, "*Pre-Run-Time Schedulability Analysis of P-NET Fieldbus Networks*", in proceedings of IECON'98, Aachen, Germany, September 1998.

[10] ESPRIT Project 5206 – Fieldbus Integration into CIM (FICIM), "User Requirements Study", Document 92.015/D05/RD, November 1992.

[11] E. Tovar, F. Vasques, "*Scheduling Real-Time Communications with P-NET*", in Digest 98/306 of the IEE Colloquium on Real-Time Systems, York, England, April 1998.