Pre-Run-Time Schedulability Analysis of P-NET Fieldbus Networks

Eduardo Tovar
Department of Computer Engineering
Polytechnic Institute of Porto
Rua de São Tomé
4200 Porto, Portugal
emt@dei.isep.ipp.pt

Francisco Vasques
Department of Mechanical Engineering
University of Porto
Rua dos Bragas
4099 Porto Codex, Portugal
vasques@fe.up.pt

Abstract - P-NET is a fieldbus industrial communication standard, which uses a Virtual Token Passing MAC mechanism. In this paper we establish pre-run-time schedulability conditions for supporting real-time traffic with P-NET. Essentially we provide formulae to evaluate the minimum message deadline, ensuring the transmission of real-time messages within a maximum time bound.

I. INTRODUCTION

The information flow in a manufacturing environment is usually structured into different hierarchical levels that place different requirements on the communication technology. At the field level, the lowest level of the automation hierarchy, transactions take place frequently, involving short quantity of data and demanding bounded completion times.

The field level includes the process-relevant field devices, such as sensors and actuators. We are specially addressing what is commonly designated by small control loops within the factory automation hierarchy. This is the case of the so-called reflex functions (regulation and controlling) where the control level is hierarchically located directly above the field level (figure 1).

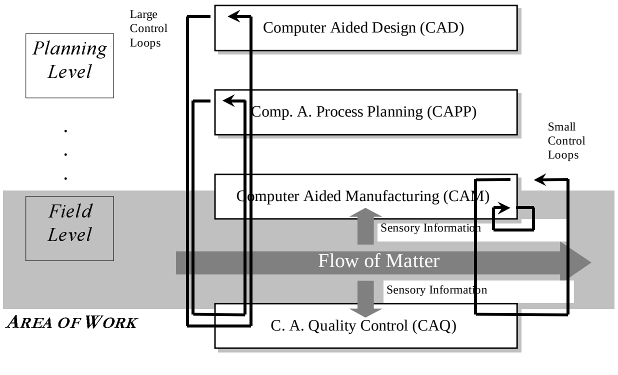

Fig. 1 Control Loops in the Factory Automation Hierarchy

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.

Among other characteristics, these hierarchical levels have dissimilar message flows, in terms of required response time, amount of information to be transferred, required reliability or message rate (how frequently an application task sends a particular type of message, for instance, from a sensor to the process controller) [1].

In a rough way, one can say that time constraints are more stringent as we go down in the factory 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).

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 for 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 [2,3], the centralised polling in FIP [3,4], the collision avoidance CSMA in CAN [5], the virtual token passing in P-NET [3,6] and the TDMA in TTP [7].

Recently, several studies on the ability of fieldbus networks to cope with real-time requirements have been presented, such as [8,9] on CAN, [10-12] on FIP, [13,14] on Profibus and finally [7] on TTP.

In this paper, we analyse the ability of the P-NET fieldbus network to cope with the timing requirements of a Distributed Computer Controlled System (DCCS), where messages associated to discrete events should be made available within a maximum bound time.

II. A BRIEF DESCRIPTION OF P-NET

The name P-NET is a derivation of “Process Network”. P-NET is designed as a communications link between distributed process control sensors, actuators and programmable controllers. Interested readers may find useful information in [15].

A. General Characteristics

P-NET uses the RS-485 multi-drop standard, with asynchronous transmission at 76800 bits/sec. This data rate resulted from weighing up the conflicting requirement for data to be transported as fast as possible, but not at such

---

This work was partially supported by FLAD under the project SISTER 471/97 and by ISEP under the project REMETER.

speed as to negate the use of standard microprocessor UARTS, or restrict the usable distance or cable type [16].

P-NET is a multi-master and a multi-net standard. All communication is based on a principle, where a master sends a request and the addressed slave immediately returns a response. For multi-master support, P-NET uses a Virtual Token Passing (VTP) approach.

In P-NET, several bus segments can be interconnected into a larger network through gateways, in a way that any master on the network can transparently access any node within the network without the need for special programs in the gateways or masters. Such segmentation allows the increase of data throughput in the total system, as it provides traffic isolation.

B. P-NET'S Data Link Layer

The Data Link Layer (DLL) tasks include the bus access control, the creation and recognition of frame boundaries and node addresses and the transmission error control. The bus access control will be explained in detail in §IIC.

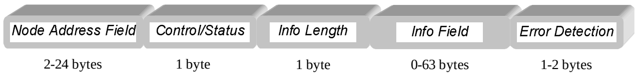

Fig. 2 P-NET Frame Structure

Communication frames (figure 2) contain a node address field, (the first address being the node address of the receiver), Control/Status, the length of the info field (may be 0), the Info itself (if Info Length > 0), and finally the error detection code(s).

All the frame bytes are sent asynchronously, with one start bit (logical zero), 8 data bits with LSB first, one address/data bit and one stop bit (logical 1). Within a frame, a start bit must immediately follow a stop bit. The frames are separated by an idle period of 11 bits periods or more, and the first byte of a frame has the address/data bit = 1. The rest has the address/data bit = 0.

Every received byte is checked for the correct Start Bit, Address/data bit and Stop bit. If this is not correct, Overrun/Framing error is set in Control/Status Field. The complete frame error check uses one, in the case of reduced error detection, or two error detection codes. If the checking fails, Error Detect Failure is set in Control/Status Field.

Bits 0-6 in the first byte of a frame contain the sender node address. Bit 7 indicates if the frame contains a request ("0") to the slave program or a response ("1") to the master program. The normal node address (NA) range is 1-125 decimal. NA 0, 126 and 127 are used, respectively, for internal applications, for broadcasting without acknowledge and for test purposes.

C. P-NET'S Medium Access Control (MAC)

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 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, allowing the first master to access the bus 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 to access the bus.

If a master has nothing to transfer (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 to access the bus. 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”.

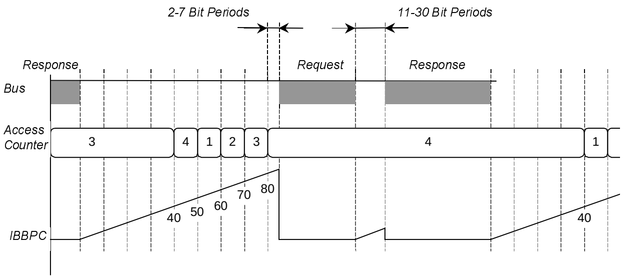

Fig. 3 P-NET Virtual Token Passing

Figure 3 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).

The sending node address in each frame is used to synchronise the access counters in all the masters. When a master is out of synchronisation (because a transmission error or reset/power up) it is not allowed to access the bus before being re-synchronised.

If the IBBPC counter is higher than or equal to 360, the token master should send a normal frame or a **sync** frame, in order to allow master's re-synchronisation. A **sync** is one byte that contains the node address of the token master, with bit 7 = 1. No device will receive the byte but all IBBPC counters will be cleared, thus resulting in AC counters synchronisation. Figure 4 illustrates the synchronisation.

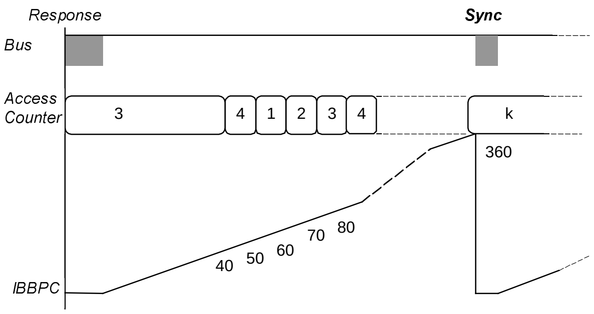

Fig. 4 Synchronisation Detail

D. Multihoping in P-NET Networks

The P-NET multinet system feature allows for routing through up to 10 gateways. These multihoping capabilities are based on simple rules for address conversion inside gateways.

P-NET supports four types of addresses: simple, complex, extended and response address types. The simple and response addresses use only 2 bytes. The extended address uses 4 bytes (2 destination and 2 source address bytes). The complex address may use up to 24 bytes. Figure 5 is an example of complex address type. P-NET uses the complex addressing scheme to route frames through gateways.

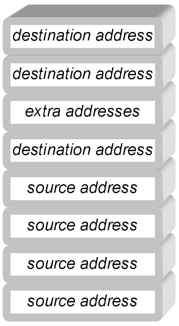

Fig. 5 P-NET Example of Extended Address

III. BASIC P-NET PRE-RUN-TIME 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 in the same message stream is longer than the message stream deadline. This means that in the outgoing queue there will not be two waiting messages from the same stream.

A. Network and Message Models

A network is composed of **nm** master stations. Each **k** master station has $\mathbf{ns}^{(k)}$ associated 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 $\mathbf{S}_i^{(k)} = (\mathbf{C}_i^{(k)}, D_i^{(k)})$, where $\mathbf{C}_i^{(k)}$ 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. Such relative deadline is the maximum admissible time to complete a message cycle. Additionally, we denote a bit period as **bp**.

B. Maximum Virtual Token Cycle

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

$$v\mathbf{tcycle} = \sum_{i=1}^{nm}\left(7\times\mathbf{bp} + \max_{j=1..ns^{(i)}}\left(\mathbf{C}_j^{(i)}\right) + 40\times\mathbf{bp}\right) \qquad (1)$$

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

C. 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:

$$\mathbf{ns}^{(k)}\times v\mathbf{tcycle} \qquad (2)$$

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

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

Thus, we may conclude that other queuing strategies, such as priority queues, rather than FIFOs would be advisable. In [17] the authors give some guidelines concerning the implementation of deadline based priority queues for P-NET fieldbus networks.

IV. SCHEDULABILITY CONDITIONS FOR MULTIHOP P-NET NETWORKS

Suppose that we have a P-NET network composed by 4 masters (M1, M2, M3, M4) and 4 slaves (e1, e2, e3, e4), all connected to one network segment. Each of the masters deals with 2 message streams, as shown in figure 6.

The figures in this section reflect a regular bus topology. In fact, P-NET bus is a physical multi-drop ring without terminators.

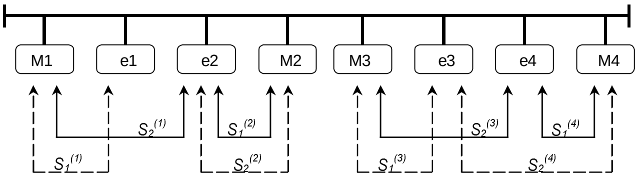

Fig. 6 P-NET Network Example

If we consider that the maximum token "holding" time in each station is 250x**bp** (this means that the maximum message lengths is equal in each master), then the schedulability condition (expression 1) stands that the relative deadlines of the messages should be greater than:

$$D \geq 2 \times 4 \times 250 \times \mathbf{bp} = 2000 \times 1/76800 \cong 26\,\text{ms}$$

Figure 6 illustrates perfectly, in a reduced scale, the advantages of segmentation. In fact, the whole network could be composed of two segments, grouping M1, M2, e1 and e2 in one segment and M3, M4, e3 and e4 in another segment (figure 7). Just for simplification we are not suggesting the use of an existent master (M1, M2, M3 or M4) with multi-port capabilities.

As there is no need for sending frames (related to different message streams) through the gateway, relative deadline of messages may go down to:

$$D \geq 2 \times 3 \times 250 \times \mathbf{bp} = 1500 \times 1/76800 \cong 19{,}5\,\text{ms}$$

However, in more complex systems, it is not likely that all the information flows are restricted to their own segment. As slave nodes tend to group I/O in racks, it is possible that specific information flows demand interoperation between masters and slaves belonging to different network segments.

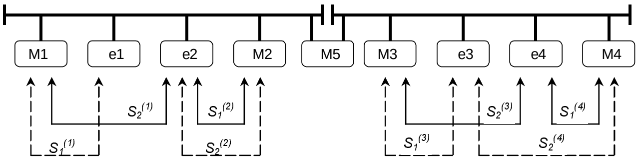

Fig. 7 P-NET Network with 2 Segments

Figure 8 illustrates the sequence corresponding to master/slave transactions through 2 gateways.

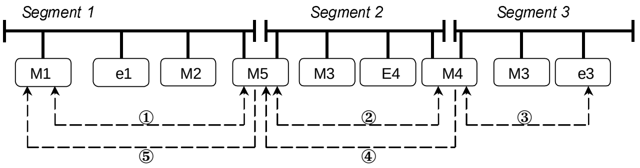

Fig. 8 Sequence of Transactions with 3 Segments

If master M1 (segment 1) wants to read a value corresponding to a sensor associated with the slave e3 (segment 3), gateways M5 and M4 will be used. The sequence of transactions is the following (note that in this case the P-NET frames use the complex addressing, with the explicit addresses of the intermediate nodes):

1. when M1 gains access to the network (in particular to segment 1), and the referred message is the first one in the outgoing queue, M1 sends the request and M5 responds immediately "answer comes later";
2. when M5 gains access to segment 2, and the referred message is the first one in the outgoing queue, M5 sends the request and M4 responds also "answer comes later";
3. when M4 gains access to segment 3, and the referred message is the first one in the outgoing queue, M4 sends the request and finally e3 responds with the required information;
4. when M4 gains access to segment 3, and the referred message is the first one in the outgoing queue, M4 sends a request (containing the required information) without response to M5;
5. when M5 gains access to segment 1, and the referred message is the first one in the outgoing queue, M5 sends a request (containing the required information) without response to M1.

In this sequence there are 5 situations corresponding to access and queuing delays. In general, if **h** represent the number of intermediate gateways, there will be:

$$2 \times \mathbf{h} + 1 \tag{4}$$

We will now re-define the schedulability condition, considering segmentation.

A. Complementary Notation

Consider that a P-NET network has **nseg** number of segments. As each station will be associated to a specific network segment, we define function *σ( )* as:

$$\sigma(\mathbf{k}) = \mathbf{i},_{k \in \{1,..,nm\},\ i \in \{1,..,nseg\}} \tag{5}$$

where *σ(***k***)* = **i**, gives the segment number (**i**$\in$*{1*,..,**nseg***}*) of master station **k**.

When supporting multi-hoping, a message stream may be relayed throughout several gateways. Each gateway will in fact correspond to **n** independent networks master stations, being **n** the number of different segments the gateway interconnects. Each message to be relayed by a gateway will in fact constitute a specific stream in two of the masters that constitute the gateway. So, the number of additional masters the message stream will use is given by function

$$\lambda(\mathbf{s}) = \mathbf{i} \tag{6}$$

where *λ(***s***)* = **i** gives the number of master stations (within gateways) a specific message stream **s** will use to be relayed. In fact, if **h** is the number of gateways used by stream s, this number will be $2 \times \mathbf{h}$.

We can thus define a **gtw** *(,)* function as:

$$\mathbf{gtw}(\mathbf{s}, \mathbf{g}) = \mathbf{k},_{g \in \{1,..,\lambda(s)\},\ k \in \{1,..,nm\}} \tag{7}$$

where **s** identifies a message stream $\mathbf{S}_a^{(b)}$, and **g** is the $\mathbf{g}^{th}$ master (belonging to a gateway) used to relay the message stream **s** through a multihop network.

B. ***D****e****a****dline* **C***o***nst***r***a***int*

Each P-NET segment has it’s own virtual token rotation procedure. Thus, we can define the maximum virtual token rotation time in a segment ξ as:

$$\boldsymbol{v}\mathbf{tcycle}_{seg\ \xi} = \sum_{\substack{i=1..nm,\\ \sigma(i)=\xi}} \left(7\times\mathbf{bp} + \max_{j=1..ns^{(i)}} \left(\mathbf{C}_j^{(i)}\right) + 40\times\mathbf{bp} \right) \tag{8}$$

If a message is to be relayed through *0* gateways, the deadline constraint will be given by:

$$\min_{l=1..ns^{(k)}} \left\{D_l^{(k)}\right\} \geq \mathbf{ns}^{(k)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma(k)}, \forall_{station\ k} \tag{9}$$

If a message stream is to be relayed through *1* gateway, the deadline constraint for that stream is:

$$\begin{aligned} D_l^{(k)} \geq {} & \left(\mathbf{ns}^{(k)} + \mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},1\right)\right)} \right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma(k)} + \\ & + \mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},2\right)\right)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma\left(gtw\left(S_{(l)}^{(k)},2\right)\right)} + 2\times\phi \end{aligned} \tag{10}$$

where ϕ stands for the time needed by the gateway to transfer frames between communication stacks.

If a message stream $\mathbf{S}_l^{(k)}$ is to be relayed through *2* or more gateways, the deadline constraint for that stream is given by expression (11).

$$\begin{aligned} D_l^{(k)} \geq {} & \left(\mathbf{ns}^{(k)} + \mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},1\right)\right)} \right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma(k)} + \\ & + \sum_{j=1}^{h-1} \left[\left(\mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},1+j\right)\right)} + \mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},2+j\right)\right)} \right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma\left(gtw\left(S_{(l)}^{(k)},1+j\right)\right)} \right] + \\ & + \mathbf{ns}^{\left(gtw\left(S_{(l)}^{(k)},\lambda\left(S_{(l)}^{(k)}\right)\right)\right)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ \sigma\left(gtw\left(S_{(l)}^{(k)},\lambda\left(S_{(l)}^{(k)}\right)\right)\right)} + 2\times\mathbf{h}\times\phi \end{aligned} \tag{11}$$

C. Nume*r***ic***a***l Ex***a***mple**

Figure 9 illustrates a network example with 3 segments.

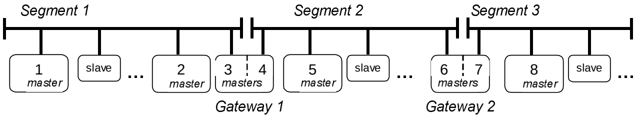

Fig. 9 Numerical Example Network

The 8 masters have the following number of message streams:

TABLE I
MESSAGE STREAMS CHARACTERISATION

Master	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ns(k)	3	4	3	2	1	4	5	6
Max{$C_j^{(k)}$}	200*bp*	200*bp*	200*bp*	200*bp*	200*bp*	200*bp*	200*bp*	200*bp*

Within those message streams, both $\mathbf{S}_1^{(1)}$ and $\mathbf{S}_2^{(8)}$ message streams are to be relayed through the gateways as characterised in table II.

TABLE II
CHARACTERISATION OF MULTIHOPING MESSAGE STREAMS

	$S_1^{(1)}$	$S_2^{(8)}$
λ( )	2	4
gtw (,1)	3	7
gtw (,2)	4	6
gtw (,3)	-	4
gtw (,4)	-	3

These two streams will impose two additional (resulting for messages being relayed through this gateway) message streams in masters 3 and 4 (gateway 1) and one additional message stream (for the same reason) in masters 6 and 7 (gateway 2). Table III reflects the aggregate number of message streams per master station.

TABLE III
AGREGATE NUMBER (INTRINSIC PLUS EXTRINSIC) OF MESSAGE STREAMS PER STATION

Master	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
ns(k)	3	4	5	4	1	5	6	6

We also assume that in both the gateways ϕ can be neglected.

We evaluate the $\boldsymbol{v}\mathbf{tcycle}_{seg\ \xi}$ for each segment (8), with **bp** = 1/76800sec:

$$\boldsymbol{v}\mathbf{tcycle}_{seg\ 1} = 3\times 247\times\mathbf{bp} \approx 9{,}65\,\text{ms}$$

$$\boldsymbol{v}\mathbf{tcycle}_{seg\ 2} = 3\times 247\times\mathbf{bp} \approx 9{,}65\,\text{ms}$$

$$\boldsymbol{v}\mathbf{tcycle}_{seg\ 3} = 2\times 247\times\mathbf{bp} \approx 6{,}43\,\text{ms}$$

We can now determine the deadline constraints for message streams $\mathbf{S}_1^{(5)}$, $\mathbf{S}_1^{(1)}$ ans $\mathbf{S}_2^{(8)}$.

$$D_1^{(5)} \geq \mathbf{ns}^{(5)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 2} \approx 1\times 9{,}65 = 9{,}65\,\text{ms}$$

$$\begin{aligned} D_1^{(1)} &\geq \left(\mathbf{ns}^{(1)} + \mathbf{ns}^{(3)}\right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 1} + \mathbf{ns}^{(4)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 2} = \\ &= (3+5)\times 9{,}65 + 4\times 9{,}65 = 115{,}80\,\text{ms} \end{aligned}$$

$$\begin{aligned} D_2^{(8)} &\geq \left(\mathbf{ns}^{(8)} + \mathbf{ns}^{(7)}\right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 3} + \left(\mathbf{ns}^{(6)} + \mathbf{ns}^{(4)}\right) \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 2} + \\ &+ \mathbf{ns}^{(3)} \times \boldsymbol{v}\mathbf{tcycle}_{seg\ 1} = 12\times 6{,}43 + 9\times 9{,}65 + 5\times 9{,}65 = \\ &= 212{,}26\,\text{ms} \end{aligned}$$

If the network were not segmented, the same three examples would lead to the following values:

$$\boldsymbol{v}\mathbf{tcycle} = 8\times 247\times\mathbf{bp} \approx 25{,}73\,\text{ms}$$

$$D_1^{(5)} \geq 1\times 25{,}73 = 25{,}73\,\text{ms}$$

$$D_1^{(1)} \geq 3\times 25{,}73 = 77{,}19\,\text{ms}$$

$$D_2^{(8)} \geq 6\times 25{,}73 = 154{,}4\,\text{ms}$$

This numerical example clearly illustrates the options a system designer should make in order to comply with the distributed application real-time requirements. It is clear that message streams that are to be relayed through P-NET gateways will have the their response times significantly increased. Conversely, message flows within the same segment will very much see their response times significantly reduced.

V. CONCLUSIONS

In this paper, we provide basic pre-run-time schedulability conditions for supporting real-time communications with P-NET fieldbus industrial communication networks.

As the schedulability condition very much depends on the number of master nodes existent in a P-NET network, we suggest and analyse the advantages of using P-NET multihoping devices for supporting segmentation. As the segmentation allows for a reduction of the maximum bound inter-arrival time of the virtual token, if the network segments group the adequate nodes, tighter message deadlines can then be supported.

VI. REFERENCES

[1] J.-D. Decotignie and P. Pleinevaux, “A Survey on Industrial Communication Networks”, **in Annales des Télécommunications, invited paper**, Vol. 48, No. 9-10, pp. 435-448, 1993.

[2] Profibus Standard DIN 19245 part I and II, translated from German, **Profibus Nutzerorganisation e.V.**, 1992.

[3] EN 50170, “**General Purpose Field Communication System**”, Vol. 1/3 (P-NET), Vol. 2/3 (Profibus), Vol. 3/3 (FIP), CENELEC, December 1996.

[4] Union Technique de l'Electricité. Normes FIP NF C46-601 to NF C46-607, (AFNOR) 1990.

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

[6] The P-NET Standard, International P-NET User Organisation ApS, November 1994.

[7] H. Kopetz and G. Grünsteidl, "TTP -- A Protocol for Fault-Tolerant Real-Time Systems", IEEE Computer, Vol. 27, No.1, pp. 14-23, January 1994.

[8] K. Tindell, H. Hansson and 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.

[9] K. Tindell, A. Burns and A. Wellings, “Calculating Controller Area Network (CAN) Message Response Times”, **in Control Engineering Practice**, Vol. 3, No. 8, pp. 1163-1169, 1995.

[10] 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.

[11] P. Raja, L. Ruiz and J.-D. Decotignie, “On the Necessary Real-Time Conditions for the Producer-Distributer-Consumer Model”, **in Proceedings of 1st IEEE Workshop on Factory Communication Systems (WFCS'95)**, Leysin, Switzerland, October 1995.

[12] 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).

[13] E. Tovar and F. Vasques, “Guaranteeing Real-Time Message Deadlines in Profibus Networks”, **to be presented at the 10th Euromicro Workshop on Real-time Systems**, Berlin, Germany, June 1998.

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

[15] Process Data A/S Homepage, **in http://www.p-net.dk**.

[16] C. Jenkins, “P-NET as a European Fieldbus Standard EN 50170 vol. 1”, **in Institute of Measurement + Control Journal**, 1997.

[17] E. Tovar and F. Vasques, “Scheduling Real-Time Communications with P-NET”, **in IEE Real-Time Systems Colloquium, IEE Digest 98/306**, York, England, April 1998.