Pre-Run-Time Schedulability Analysis in Fieldbus Networks[†]

Francisco Vasques[‡], Guy Juanole

LAAS du CNRS
7, Av. du Colonel Roche, 31077 Toulouse Cedex, France
phone: (+33).61.33.62.63; fax: (+33).61.33.64.11, e-mail: {vasques; juanole}@laas.fr

Abstract: The goal of this paper is to provide a methodology for the pre-run-time schedulability analysis of Fieldbus networks, based on the evaluation of the worst-case response time. A comparison of the well-known Fieldbusses Profibus and FIP is made.

1: INTRODUCTION

In the technological context of today, we have the growing development of distributed time-critical applications, in particular in the Production / Automation systems area. These applications have timing constraints which, if not satisfied, can induce catastrophic situations. Then, implementing such kind of applications requires the use of distributed real-time and fault-tolerant computer systems, i.e., *responsive systems*.

Responsive systems have several requirements [Kop, 93], among which the *schedulability* is of the first importance. Two scheduling strategies can be distinguished: static (or pre-run-time) and dynamic scheduling. The scheduling strategy must guarantee the timeliness property. This property, concerning the pre-run-time scheduling strategy, can be evaluated through the evaluation of the *worst-case response time*.

Responsive systems in a distributed context are based on communication networks which have then to provide responsive communication services.

In the Production/Automation systems context, an important distributed time-critical application is the physical process control, which is usually based on a Fieldbus [Dec, 93] (low level industrial local network which connects sensors, actuators and automatic controllers).

The goal of this paper is precisely, to present a study which concerns the pre-run-time schedulability analysis in well known Fieldbus networks (Profibus [Pro, 91], FIP [Fip, 91]). This paper includes four sections. A first section is concerned by a general presentation of a distributed responsive system by focusing its main constraints (coming from a time-critical application). A second section presents the main mechanisms of the Fieldbuses: Profibus and FIP. A third section is concerned by the main formulas establishment for the analysis of the pre-run-time schedulability of both Fieldbuses. A final section describes a comparative example.

2: DISTRIBUTED RESPONSIVE SYSTEMS

Consider a physical process controlled by a real-time computer system based on a Fieldbus network. The global architecture is presented on the figure 1. We briefly explain it:

- The *controlled system* (real-world) consists of physical elements (sensors and actuators);
- The *controlling system* consists of application processes, here called *Real Time Entities* (RTEs); The RTEs are abstract views both of physical elements (1) and algorithms (2) which make computations and elaborate control laws; RTEs (1) are called

external RTEs (noted ext. RTE; an ext. RTE is an information producer, if associated to a sensor or an information consumer if associated to an actuator); RTE (2) are called internal RTEs (noted int. RTE; an int. RTE is an information consumer and producer). The real-time attribute of the controlling system results from the interactions with the controlled system (which imposes the timing). The RTEs context is time dependent which induces, in particular, the following problems: temporal validity of a produced information; consistency of the information coming from one producer and consumed by several consumers;

- The communication system high-level (called application layer) provides the service of transferring the time dependent information produced by a producer (this information is called a Real-Time Object: RT-O) to the different consumers (the copies in the consumer nodes are called RT-O images: RT-O_i). In order to solve the consistency problem, we need updating protocols which must have both the *atomicity* and *timeliness* properties. Depending on the triggering mechanisms for the start of the RT-O_i updating, two approaches can be distinguished [Kop, 91]: the Event-Triggered (ET) approach where the RT-O_i are updated whenever a state meaningful change in an RT-O occurs; the Time-Triggered (TT) approach, where the RT-O_i updating are periodically initiated at pre-determined points in time. The second approach leads to an updating traffic load which is higher than in the first one, but which is also completely predictable (the first one induces problems when simultaneous multiple updating are required).
- The communication system lower level, and more precisely the link layer has to manage the sharing of the transmission medium for the frame exchanges. The frame exchanges schedulability is an hard point, which must satisfy time constraints (resulting from the time constraints in the higher layers). This approach can also be ET or TT.

Figure 1: Distributed System global architecture

[‡] Mr Vasques is also affiliated to FEUP, Universidade do Porto, Portugal and can as well be contacted as vasques@fe.up.pt

[†] This work was partially supported by a JNICT grant (n° BD 1278/91)

The presentation of the global architecture has emphasised the Producer / Consumer aspect and the consequent updating problem. However we have also other causes of information exchanges between the nodes: alarm occurrences; configuration definition messages (which do not necessarily have time constraints), Then, by considering the distributed responsive system at its link layer service interface, we can summarise the different types of traffic in the following way: we have traffic with timing constraints, i.e., real-time traffic (the real-time traffic can still be subdivided into periodic and sporadic traffics), and traffic without traffic constraints (we call it background traffic).

3: THE DATA LINK LAYER OF THE FIELDBUSES PROFIBUS AND FIP

A complete presentation of Profibus and Fip data link layers is out of the scope of this paper. However, some relevant characteristics which enables the support of a responsive system will be summarised in this section.

3.1 - The Profibus standard [Pro, 91]

A Profibus network has a token passing medium access control method, where the communication can be initiated only by the master station which has the token. The token passing mechanism is an usual solution to solve the medium access conflict in ET systems, since it guarantees that each node orderly receives the rights for transmission on the bus.

The idea behind a timed token protocol is to control the token rotation time. At the network set-up, a protocol parameter called *Target Rotation Time* (T_{TR}) must be evaluated in order to keep within the system reaction time required by the controlled system. This T_{TR} parameter is user evaluated and must be:

- High enough to provide the sufficient time window for all the previously requested messages transfers;
- Small enough to match the controlled system dynamics;

In systems based on the timed token protocol, the timeliness test is usually based on simulated pattern loads. The remaining question is how valid and truthful are such loads even in burst circumstances. For instance, the Profibus standard suggests an average load based value for the T_{TR} set-up. This is clearly an unacceptable approach for a timeliness system. In this paper we propose a T_{TR} set-up based on the knowledge of the worst-case traffic conditions, which completely guarantees the system timeliness behaviour.

When the token arrives at a node station, the higher priority messages will be transferred first. Thus, we assign the higher priority to the real-time message set, i.e., to the periodic and sporadic messages.

3.2 - The FIP standard [Fip, 91]

The FIP network is based on a centralised TDMA control solution, where a central entity, the Bus Arbiter (B.A.), controls all the bus accesses enabling data transfer exchanges between communicating entities. The B.A. manages all messages exchanges according to a pre-defined time-line: the scanning table. This table is organised as a sequence of micro-cycles, the Macro-cycle, which is indefinitely repeated.

The micro-cycle T_{mc} is the smallest transmission window, where each periodic message stream may be able to transmit at most one single message:

$$T_{mc} = \left\lceil \frac{HCF(TP_i)}{L} \right\rceil \text{ for } 1 \leq i \leq p$$

where HCF states for Highest Common Factor

The Macro-cycle T_{MC} regroups wanted micro-cycle sequence, in such a way that periodic streams are able to timely transmit desired messages.

$$T_{MC} = \left\lceil LCM(TP_i) \right\rceil \text{ for } 1 \leq i \leq p \text{ (note 2)}$$

where LCM states for Least Common Multiple.

The scanning table generation difficulty increases with the least common multiple of the periodic streams (combinatorial explosion).

Message stream transfer is executed in the following way:

- During the micro-cycle first phase, the B.A. induces a *pre-defined* sequence of messages exchange (periodic message stream transfer). During this phase, distant node stations, if addressed, may signalise to the B.A. message pending aperiodic requests (implicit pooling). Explicit pooling is also possible but we will not consider this option.
- In the following unused micro-cycle time (if any), the B.A. induces exchange of pending aperiodic message requests which have been previously made.

In the following analysis, periodic stream messages transfer is made using FIP periodic services, while sporadic and background message transfer is supported respectively by high and low priority aperiodic transfer services.

4: PRE-RUN-TIME SCHEDULABILITY ANALYSIS

We now develop the main formulas which allows us to analyse the pre-run-time schedulability of the link layer of the following Fieldbusses: Profibus and FIP. In a first sub-section we define a message traffic model relative to the link layer.

4.1 :Traffic model

4.1.1: Link layer service

We consider three sets of message streams:

- A set P of p real-time periodic message streams: $P = \{P_1, P_2, \dots, P_i, \dots, P_p\}$; TP_i and dP_i are respectively the stream P_i message period and the deadline (latest ending time); CP_i is the length of a message of the stream P_i : $CP_i < dP_i$ TP_i .
- A set S of s real-time sporadic message streams: $S = \{S_1, S_2, \dots, S_j, \dots, S_s\}$; TS_j and dS_j are respectively the minimum inter-arrival time between successive messages of the stream S_j and the deadline (latest ending time); CS_j is the length of a message of the stream S_j : $CS_j < dS_j$ TS_j .
- A set B of n background message streams: $B = \{B_1, B_2, \dots, B_k, \dots, B_n\}$; CB_k is the length of a message of the stream B_k .
- Furthermore we suppose: the message streams are independent; nothing is assumed about the real-time message stream distribution among the nodes of the Fieldbus; there is only one background message stream per node.

In the figure 2 we summarise the traffic model, representing the different streams and the different Data Link Entities (DLE), which implements the frame exchange (MAC and LLC functions) and provide the link layer service.

Figure 2: Message model

4.1.2: Link Layer protocol

The different stream messages are encapsulated into PDUs (they represent the data field of these PDUs, which therefore are called *data PDUs*) for their transmission, which can also require the

exchange of *control PDUs*. We characterise the resulting overhead by the following parameters:

- ϕ : overhead in a data PDU, i.e., the duration of the addressing field, the checksum field, ...;
- δ : overhead in terms of control PDUs i.e., the duration of these PDUs and the duration of their exchanges. We further subdivide, when necessary, this overhead into: acknowledge overhead (δ_A), token passing overhead (δ_T) and pooling overhead (δ_P). δ_A is the minimum overhead in terms of control PDUs needed for transferring data PDUs.

Considering these two parameters, we can define the minimum message length of the different streams (from the point of the view of the link layer) in the following way:

$$CP_i^* = (CP_i + \phi_i + \delta_{Ai})$$

$$CS_j^* = (CS_j + \phi_j + \delta_{Aj})$$

$$CB_k^* = (CB_k + \phi_k + \delta_{Ak})$$

Another type of overhead exists in the case where there is a DLE, in a node, which is the global scheduler: this overhead s is called the *dead interval*. It consists of the elapsed time between the message request time in the DLE of a node, and the instant where the global scheduler is informed of this request.

4.2: Evaluation of the worst-case response time

4.2.1: Profibus analysis

Considering a token cycle of duration t_c , we must evaluate the bus load during t_c . We have two kinds of loads:

- load 1: the load resulting directly from the link layer service;
- load 2: the load resulting from the token passing mechanism;

Consider the load 1; it has three components:

- Periodic traffic: $\sum_{i=1}^p CP_i^* \left\lceil \frac{t_c}{TP_i} \right\rceil$;
- Sporadic traffic: $\sum_{j=1}^s CS_j^* \left\lceil \frac{t_c}{TS_j} \right\rceil$;
- Background traffic: $\sum_{k=1}^n CB_k^* \cdot n_k$

(by supposing that each node need to send, during each cycle, at most n_k background messages);

As a protocol based on the token passing mechanisms allows each node to use the bus only once per token cycle, it is then necessary to avoid, in each node, the accumulation of periodic and aperiodic traffic which will induce the no-satisfaction of the deadlines. Then we must have: $TP_i \geq t_c$ and $TS_j \geq t_c$.

Consider the load 2: it is given by:

$$\sum_{k=1}^n \tau_{Tk}$$

We can now express the maximum bus load H during a token cycle (by expressing the three components of the load 1, with the constraints on the periodic and sporadic traffic, and the expression of load 2):

$$H \leq \sum_{i=1}^p CP_i^* + \sum_{j=1}^s CS_j^* + \sum_{k=1}^n CB_k^* \cdot n_k + \sum_{k=1}^n d_k$$

H also expresses the maximum time interval between two consecutive token arrivals at the same node.

Worst case response time (wcrt)

In order to determine the worst case response time let us analyse the worst case situation, in one node, for a service link request: if it

is made just after the token departure, it has to wait until the next token arrival for being considered. The worst case situation occurs when we have the maximum time interval between two token arrivals. Thus:

$$wcrt = H$$

Schedulability condition

The schedulability condition imposes that $wcrt$ must be always smaller than the message deadlines. Then the schedulability condition is:

$$(dP_i > H) \quad (dS_j > H), \forall i, j$$

Target Rotation Time (T_{TR}) evaluation

T_{TR} must be greater than the maximum elapsed time between two node k token arrivals, plus the maximum amount of real-time traffic that can be found whichever k . Thus T_{TR} may be evaluated as:

$$T_{TR} \geq H + \max \left[\left(\sum_i CP_i^* + \sum_j CS_j^* \right) : (i \wedge j) \in node_k \right]$$

4.2.2: FIP analysis

In FIP we must distinguish the schedulability test of the periodic traffic and the sporadic traffic.

As the periodic traffic execution is completely pre-defined at the scanning table definition, we have its schedulability previously guaranteed.

On the other hand, we need to evaluate the schedulability of the aperiodic traffic. As, in FIP, we have a global scheduler (B.A.), the worst case response time ($wcrt$) and the schedulability conditions depends on the dead interval σ (sub-section 4.1) evaluation.

Dead interval evaluation

We define σ_k , the *dead interval* of the node k , as the maximum interval between two polling operations of node k . This dead interval, is given by the minimum period of the periodic traffic of this node:

$$s_k = \{ \min(TP_i) \}, \forall P_i \in node_k \text{ for } k = 1 \dots n$$

During this interval, the B.A. remains uninformed of pending aperiodic requests in the node k .

Worst case response time ($wcrt$)

Consider a sporadic request relative to a sporadic stream which appears in a node k at time t . This request will reach the B.A., at most, at the time $t + s_k$. We suppose the hardest condition for the arrival of this request in the queue of the B.A. which considers the sporadic requests (this queue is noted RQ): this request arrives whereas requests relative to all the other sporadic streams are also waiting for service.

Two cases have to be considered: all the sporadic requests in the queue RQ can be satisfied before the instant $t + T_{MC}$ (case 1) or not (case 2).

In order to express the conditions of these two cases, we define two durations $\Delta t(rP)$ and $\Delta t(S)$:

- $\Delta t(rP)$ is the duration of the remaining Periodic traffic in the interval $T_{MC} - \sigma_k$ (during the time σ_k , there has been periodic traffic). Such duration may be evaluated as:

$$Dt(rP) = \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC} - s_k}{TP_i} \right\rceil$$

- $\Delta t(S)$ is the duration of the sporadic traffic which has been generated during the T_{MC} interval (after the instant t). Such duration may be evaluated as:

$$Dt(S) = \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil$$

Therefore we have the following two cases for the worst case response time evaluation:

- Case 1: $s + Dt(rp) + Dt(S) \leq T_{MC}$, then the worst case response time for any message of the sporadic stream S_x , (node k) is:

$$wcr|_{S_x} \leq \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC} - s_k}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$$

- Case 2: Else, the worst case response time for any message of the sporadic stream S_x , (node k) is:

$$wcr|_{S_x} \leq \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC}}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$$

Schedulability condition

The schedulability condition imposes that wcr must be always smaller than the message deadlines. Then the schedulability condition, for each *worst case response time evaluation* is:

- $dS_x \leq \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC} - s_k}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$, in the case 1;
- $dS_x \leq \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC}}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$, in the case 2;

4.2.3 - Analysis conclusion

The Profibus standard, as a classical token bus network, imposes some severe restrictions to the real-time traffic scheduling problem, such as:

- Imposing an *equal* lower bound (H) for *all* the periods TP_i of the periodic streams and all the minimum inter-arrival times of the sporadic streams may lead to a poor network utilisation factor, unless the values of TP_i and TS_i are all close enough to the H bounding value. In other words, a wide range of values for the parameters TP_i and TS_j of the real-time traffic, leads to a poor network utilisation factor.
- As the background traffic influences the system real-time behaviour (H evaluation), we must bound it (n_k messages per node) in order to guarantee the system real-time behaviour.

The Fip standard, based on an TDMA approach, treats the real-time traffic scheduling problem in a more appropriate way. Imposing pre-defined windows (micro-cycles) and a periodic traffic time-line leads to an interesting solution for the periodic traffic scheduling problem. However, some associated drawbacks must be stated:

- The interdependency between the length of the *dead interval* and the periodic traffic is one of the major drawbacks of the FIP Fieldbus: the timeliness guarantee of the sporadic traffic is not dependent on its intrinsic characteristics, but on the characteristics of other traffics.
- Also, at the B.A. level, serving the sporadic messages requests in a FIFO basis, leads to an uncontrolled message priority inversion on the bus, and thus it is difficult to guarantee the timeliness of the sporadic traffic.

It's our opinion that:

- FIP is an interesting solution to support exchanges between application processes, when the generated real-time traffic is essentially periodic;
- Profibus, as it not distinguishes between periodic and sporadic traffic, is an interesting solution for essentially sporadic loads.

Our main conclusion is that each one of this standards has its own application field, which is mainly defined by the chosen triggering strategy for exchanging messages between application processes (e.g., the RT-O_i updating). The FIP standard is specially suited for TT exchanges, while the Profibus standard is a better solution for ET exchanges.

5: COMPARATIVE EXAMPLE

In the following example, we evaluate for each one of the studied Fieldbusses, its ability to support a previously defined real-time message set.

The following real-time message stream set is applied to a target network, where u stands for a 2 byte transfer duration

$$TP_i = \{160; 320; 640; 160; 320; 320\}u$$

$$dP_i = TP_i$$

$$TS_j = \{400; 400; 400; 400; 400; 400\}u$$

$$dS_j = TS_j$$

$$CP_i = CS_j = 4u$$

Also, a background message stream set, with the following characteristics, must be supported by the target network:

$$n_k = 3$$

$$CB_k = 4u$$

Figure 3: Target network

5. 1: Profibus analysis example

Protocol dependent parameters evaluation

Considering the case of an SDA service [Pro, 91], with null number of retransmissions, a fixed length frame (4u) and neglecting the transmission delay, we have:

- δ_T (token passing protocol overhead): 4u
- δ_A (acknowledge protocol overhead): 4u
- ϕ (overhead in a data PDU): 8u

Message length evaluation:

$$CP_i^* = (CP_i + \phi_i + \delta_{Ai}) = 16u$$

$$CS_j^* = (CS_j + \phi_j + \delta_{Aj}) = 16u$$

$$CB_k^* = (CB_k + \phi_k + \delta_{Ak}) = 16u$$

Worst case response time evaluation:

The maximum bus load H during a token cycle, which also expresses the maximum time interval between two token arrivals at the same node, may be evaluated as:

$$H \leq \sum_{i=1}^p CP_i^* + \sum_{j=1}^s CS_j^* + \sum_{k=1}^n CB_k^* \cdot n_k + \sum_{k=1}^n d_k$$

$$H = (6 \times 16) + (6 \times 16) + (3 \times 3 \times 16) + (3 \times 4) = 348u$$

The worst case response time for any message is 348u and occurs when the message request is made just after token release.

Schedulability condition

The schedulability condition imposes that the message request response times must be always smaller than the message deadlines.

- Periodic traffic: we can see that we have a value of H which is greater than five of the deadlines of the periodic message streams. Thus the periodic traffic is not schedulable.
- Sporadic traffic: we can see that it is schedulable.

TTR evaluation:

The Target Rotation Time may be evaluated as:

$$T_{TR} \geq H + \max \left[\left(\sum_i CP_i^* + \sum_j CS_j^* \right) : (i \wedge j) \in node_k \right]$$

$T_{TR} = 412u$

5. 2: FIP analysis example

Protocol dependent parameters evaluation

Considering the case of a free explicit transfer service (for aperiodic messages) [Fip, 91], 8 data bytes length frame (4u) and neglecting the transmission delay, we have:

- δp (Polling protocol overhead: id_dat): 4u
- ϕ (overhead in rp_dat): 6u

Message length evaluation:

$$CP_i^* = (CP_i + \phi_i + \delta_{Ai}) = 16u$$

$$CS_j^* = (CS_j + \phi_j + \delta_{Aj}) = 16u$$

T_{MC} and T_{mc} evaluation:

$$T_{MC} = 640u$$

$$T_{mc} = 160u$$

Dead interval evaluation

The *dead interval* (σ_k) of node station k , has been defined as the maximum interval between two polling operations of node k , and may be evaluated in the following way:

$$s_k = \{ \min(TP_i) \}, \forall P_i \in node_k \text{ for } k = 1 \dots n$$

$$\sigma_k = \{160; 160; 320\}$$

Worst case response time evaluation

The worst case response time for any sporadic message depends on the station dead interval and the duration of the remaining traffic in the interval $T_{MC} - \sigma_k$.

Two cases has been defined, so the following equation must be evaluated for each sporadic message stream:

$$\sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC} - s_k}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$$

Therefore, for each node he have the following case for evaluating the worst case response time:

- Node 1 (which includes S1): $512 < T_{MC}$ then we are in the case1
- Node 2 (which includes S2, S3): $512 < T_{MC}$ then we are in the case1

- Node 3 (which includes S3...S6): $672 > T_{MC}$ then we are in the case2

For the nodes 1 and 2, the value which has been obtained represents $wcrt$. On the other hand, for the node 3, we have to compute the new value:

$$wcrt|_{s_k} \leq \sum_{i=1}^p CP_i^* \left\lceil \frac{T_{MC}}{TP_i} \right\rceil + \sum_{j=1}^s CS_j^* \left\lceil \frac{T_{MC}}{TS_j} \right\rceil + s_k$$

Now, we can summarise the values of $wcrt$ for S_j ($j=1 \dots 6$) with the following tuple:

$$wcrt = \{512; 512; 512; 752; 752; 752\}$$

Schedulability condition

Here, by definition of the Fieldbus FIP, we are only concerned by the sporadic traffic schedulability.

We can see from the values got for $wcrt$, with respect to the values of dS_j , that the timeliness property is not guaranteed.

5. 3: Comment

This example shows that:

- Profibus is an interesting solution for essentially sporadic loads.
- FIP is an interesting solution when the generated real-time traffic is essentially periodic;

6: CONCLUSION

A methodology for analysing the timeliness property (based on the worst-case response time) of Fieldbusses has been presented and a comparison of the Fieldbusses Profibus and FIP has been made. Key elements are the traffic model (the link layer service and the link layer protocol) which is a general concept and the considerations for Profibus and FIP of respectively, the maximum time interval between two consecutive token arrivals in a node, and the dead interval in a node.

The comparison between FIP and Profibus shows that FIP is well suited for supporting periodic real-time traffic, whereas Profibus is an interesting solution for essentially sporadic traffic.

References

- [Fip, 91]; FIP Bus for exchange of information between transmitters, actuators and programmable controllers; Norme AFNOR C-46603 and C-46605; Data Link Layer and Management Layer standards; 1991
- [Iec, 93]; Digital Data Communications for Measurement and Control - Field Bus for use in Industrial Control Systems; SC65C 105 Data Link Service definition and SC65C 106 Data Link Protocol specification draft proposals; 1993
- [Kop, 91]; H. Kopetz; Event-Triggered versus Time-Triggered Real-Time Systems; in Operating Systems of the 90s and beyond; Karshmer, Nehmer (eds.), LNCS 563, Springer Verlag 1991
- [Kop, 93]; H. Kopetz; Six Difficult Problems in the Design of Responsive Systems; in Dependable Computing and Fault-Tolerant Systems (vol. 7): Responsible Computer Systems; Kopetz, Kakuda (eds.), Springer Verlag 1993
- [Dec, 93]; J. Decotignie, P. Pleinevaux; A Survey on Industrial Communications Networks; Ann. Télécommun., 48, n° 9-10, 1993
- [Pro, 91]; Profibus standard (translation of German standards); DIN 19245: Process Field Bus; 1991