Available online at www.sciencedirect.com
ScienceDirect
Computer Standards & Interfaces 29 (2007) 546–560
COMPUTER STANDARDS & INTERFACES
www.elsevier.com/locate/csi

An architecture for flexible scheduling in Profibus networks

J. Silvestre *, V. Sempere [1]

Technical University of Valencia, Ferrandiz y Carbonell 1, Alcoy 03801, Spain

Received 10 March 2006; received in revised form 31 October 2006; accepted 11 November 2006
Available online 20 December 2006

Abstract

Recently, much attention has been given to the need to endow industrial communication networks used in real-time systems with flexible scheduling. This allows control systems to adapt to the variations in the requirements of traffic generated by modifications in the environment of the system, or changes in its structure. Another active area is multimedia transmission in industrial environments. In this paper, a flexible scheduling system for Profibus networks is presented. There are two objectives. Firstly to allow the characteristics of real-time traffic to deal in run time in a fieldbus and secondly to enable the scheduling of video traffic for industrial monitoring purposes. The system proposed allows, with regard to traditional Profibus MAC/scheduling, rapid dynamic adaptation to the new requirements, minimizing the bandwidth necessary for its management and maximizing the use of the available bandwidth. As a result there is an improvement in quality of video sources as well as the number of video sources, which can co-exist with the control traffic, without affecting its QoS.

Keywords: Fieldbus; Profibus; Scheduling; Multimedia

1. Introduction

Nowadays, the need to incorporate flexible scheduling mechanisms into fieldbuses, which operate in computer control systems in real time is becoming more and more evident. Many real-time systems are highly dynamic [1] and need to provide adaptive behavior, modifying the assignment of network resources adequately and in such a way that the system can continue adapting to changes in communication needs. Also modern applications need to work with several types of tasks and requirements [2], enabling these to change dynamically with time. Another reason for this necessity is the growing complexity in which the field buses operate [3], as the number of nodes in the network is always increasing, it is more difficult to be able to predict the behavior of the system in advance. Lastly, the reduction in set-up and maintenance costs [4] as well as the concept of flexible manufacturing (FM) also establish this requirement, in that both the initial configuration and the changes generated in production lines and manufacturing cells to produce new products can be carried out in the minimum period of time.

These flexibility requirements affect the choices of protocols and paradigms in fieldbuses. Thus, fieldbuses traditionally favour flexibility or timeliness [4,5]. They also presents differences concerning efficiency in the management of asynchronous (Event-trigger paradigm) or synchronous (time-trigger paradigm) traffic [6–9]. Event-Triggered communication systems (CAN) are flexible by nature. However, they do not support composability [10], and have high requirements for worst-case analysis. Time-trigger paradigm (TTP/C or TT-CAN) work better for periodic communication from the point of view of efficient bandwidth. However it is inefficient to manage sporadic traffic with fast response requirements. The advantages and disadvantages of these paradigms suggest a combination of both paradigms, where a temporal isolation between the two types of traffic must be assured, the FTT paradigm.

On the other hand, multimedia transmission is an application that is increasingly requested in industrial environments [11]. Multimedia applications could be considered to be the third generation of computer applications [12–14], with applications

* Corresponding author. Tel.: +34 9966528483.
E-mail address: jsilves@disca.upv.es (J. Silvestre).
[1] Tel.: +34 9966528483.

0920-5489/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2006.11.006

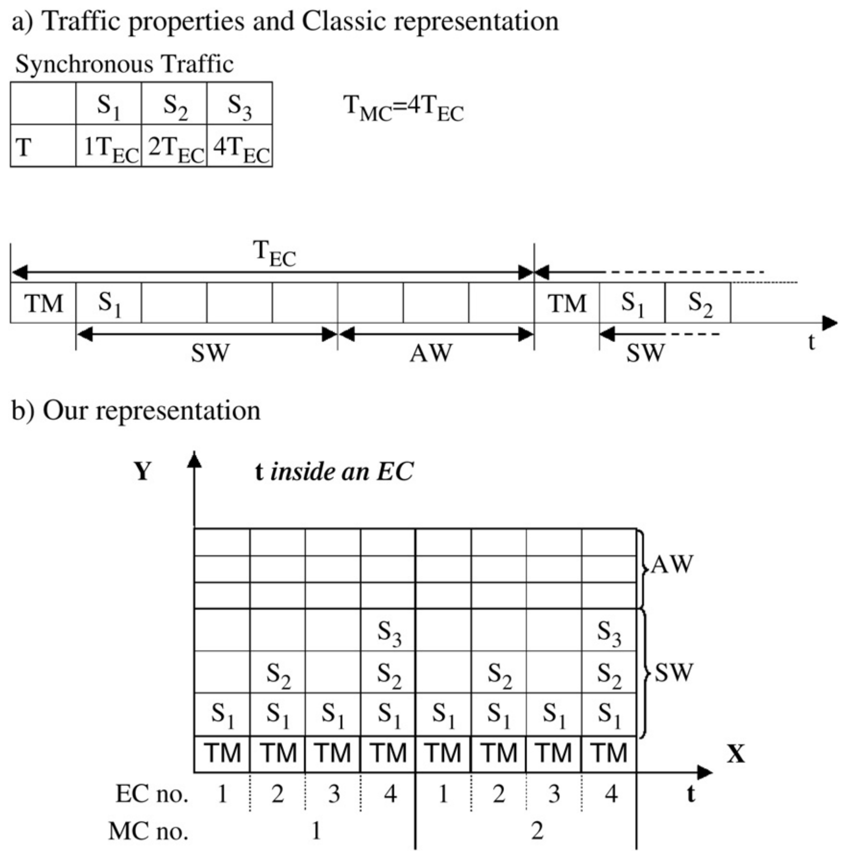

Fig. 1. EC and MC in the FTT paradigm.

such as monitoring, control, quality control, automation and industrial communications etc., which require changes to the architecture on requesting new protocols and adaptive systems. From the industrial point of view, control through image processing [15–17] and process monitoring [18,19] are the two main applications that can be benefit from this integration. However, the dynamic nature of multimedia traffic, the high bandwidth requirements and the need to not interfere in the real-time control traffic mean that integration of this system in industrial networks is a complex task.

In the area of fieldbuses Timed Token networks (like Profibus) are a well known protocol for real-time applications. The Profibus MAC is a simplified version of the timed token protocol [20] where the capacity to assign a synchronous bandwidth (H_i) to the nodes does not exist [21]. Its scheduling mechanism is DBE[2] [22,9], and therefore although the traffic requirements can change dynamically, there isn't admission control or timeliness guarantee. As a token protocol, it creates increased jitter generated by the irregularity of token arrivals, which in turn affects the stability of the control algorithms [23]. Another disadvantage of this protocol is that the bandwidth is equally distributed among all masters, independently of the requirements of each one. This means that in the complex, dynamic scenarios considered, it is difficult to ensure timeliness in the control traffic, which means the development of the FTT paradigm over this type of network is necessary.

[2] Dynamic Best Effort. Some examples Profibus, CAN, WorldFIP asynchronous.

Therefore, given the flexibility requirements of traffic control, and the dynamic nature of multimedia traffic, in this paper a dynamic scheduling to solve both problems simultaneously is analyzed. For this, the FTT paradigm is briefly introduced in Section 2 and in Section 3 the characteristics of multimedia transmission are detailed. In Section 4 FTT-Profibus is described, and in Section 5 the initialization and management protocol is defined. Before the conclusions obtained, two examples of the FTT paradigm in use are presented.

2. Flexible Time Triggered

Recently a new paradigm called FTT (Flexible Time Triggered) has been developed, aimed at giving flexibility to scheduling in fieldbuses [24]. Using this paradigm, originally developed in the Electronic Systems Laboratory in the University of Aveiro, the intention is to give mechanisms that permit the addition, elimination or adaptation of the characteristics of on-line network traffic, guaranteeing timeliness as well.

2.1. Architecture

This centralized architecture is characterized by the use of a master node, denominated Scheduler Master (SM), charged with the development and coordination of communication activities. This central node receives the communication requirements and defines both the scheduling policies and the control of admission.

In FTT the bus time is slotted in consecutive fixed duration [9,24–26] Elementary Cycles (EC) of duration T_{EC}, as in other protocols as Ethernet Powerlink [27–29], PROFInet-IRT [28], WorldFIP or some other proposals in Ethernet such as [30–32], or Profibus FDL [33]. The EC is designated to a bus time window used to interchange the traffic associated with the EC. In a table-based scheduling such as is used in some FTTs approaches [34] or some fieldbuses (WorldFIP, Fieldbus Foundation), the basic methodology to build the schedule table is the Least Common Multiple/High Common Factor (LCM/HCF) [35,36]. Then the T_{EC} (or microcycle) is typically set to the HCF of the periods of synchronous traffic flows; and the macrocycle (MC) duration T_{MC}, defines the period of time in which the pattern of synchronous requests is repeated and is calculated as the LCM of the periods of traffic flow. The size of the schedule table is determined by T_{EC} and T_{MC}. If periods of task are different and relative primes, T_{MC} will be very long, as will the size of the schedule table. To reduce this size there are some approaches, such as the use of "plan" [34] or the backtracking algorithm presented in [37]. However we will consider that all the periods of the synchronous set, denominated Φ, are harmonic with respect to T_{EC}, although a reduction in the scan periodicity has to be made [34,38,39].

$$T_i = 2^{k_i} * T_{EC}, \forall i \in \Phi, k_i = 0..\frac{T_{MAX}}{T_{EC}} \quad (1)$$

being T_{MAX} the greatest period that one synchronous stream can have, and therefore all the periods are expressed as multiples of T_{EC}. One important characteristic of this centralized architecture is the utilization of a special control message for nodes synchronization, named trigger message (TM). This creates a master/multislave transmission system since it can generate the transmission of messages in several slaves. One of the objectives of TM is that of offering the minimum overload of communication.

For the transmission of synchronous and asynchronous messages, specific windows are defined (SW: Synchronous Window, equivalent to H_i in TT networks. AW: Asynchronous Window) always guaranteeing their temporary isolation, although in some implementations other special windows exist.

In Fig. 1 we can see an example of these concepts, Fig. 1a being the classic representation of traffic distribution, in only one dimension. As in our case we need to represent the distribution traffic for a lot of streams, we use a bidimensional representation, as can be see in Fig. 1b, where X represents the time and Y the time inside one EC. Three synchronous control streams S_i with periods T defined as T_{EC} multiples are shown. The MC is calculated, and the traffic pattern is repeated continuously until some change is needed. Inside each EC, there are windows for synchronous and asynchronous traffic, and also there is time for the TM that starts the beginning of each EC.

2.2. Classification

Within this flexible scheduling scheme, we propose a classification according to the variability of work conditions (change frequencies) and response time (see Table 1). Therefore, in the worst case, we could request frequent change in traffic every few milliseconds, which would require a rapid adaptability of the system. This could be requested by Highly Dynamic Real Time Systems [1]. On another scale, Real Time Complex Systems could be considered, characterized by an elevated number of nodes [3]. In these cases a greater capacity for frequency changes might be necessary, although the system response time could have a larger margin depending on the number of nodes in the system.

As for the tasks of set-up and monitoring [4], given that it is an operation carried out by a human operator, it is not thought that the speed of changes in traffic will be introduced with a high frequency. Furthermore, in the case of monitoring, and given that this information is processed by the operator, the speed of the system to adapt to the new situation is not considered a priority. In an FM system, the changes in the production line are in an order of magnitude far greater than those commented on above, and furthermore it is not necessary for the system to be adjusted immediately to the changes introduced.

The design of an FTT schedule is strongly related to the type of requirements which need to be covered. The FTT-Profibus protocol proposed is designed to provide multimedia support and flexibility in RTCS/DRTS monitoring and FM environments, meaning that neither changes in request nor response time requirements less than 10 ms (typical time in Factory Automation) are considered.

2.3. Previous works

In [9,24] the authors analyze in detail the necessity of flexible scheduling mechanisms, presenting several examples. Particularly, in relation to the incidence in communication systems, they analyzed for example flexibility-timeliness dichotomies and Event or Time Triggered communication of fieldbuses (also analyzed in [4]), analyzing briefly the properties of some fieldbuses to deal with these. To tackle this problem in the CAN fieldbus, they presented the protocol designated FTT-CAN (Flexible Time-Triggered).

In [34] Almeida et al. present a response-time analysis for synchronous and asynchronous traffic on WorldFIP; a static table-based scheduling, with the characteristic of using a new scheduler based on dynamic tables, and therefore with the capacity to adapt to changes in traffic requirements, and providing a mechanism to improve response times of the asynchronous messages.

Pedrieras and Almeida present in [25] the protocol designated FTT-Ethernet, whose objective is basically to achieve deterministic, efficient but also flexible behavior on Ethernet. Since this technique goes to highlights that several parts of the protocol, both in the dispatcher and in the nodes, presents significant temporal

Table 1
Dynamic scheduler classification

		Response	Time
Change Frequency	VH < 10 ms	H 10–100 ms	L 0.1–1 s
VH < 10 mc	HDRTS	–	–
H 10–100 ms		DRTS–RTCS	–
L 0.1–60 s	–	Set-up	Monit.
VL > 60 s	–	–	FM

VH:Very High. H: High. L:Low. VL:Very Low. HDRTS: Highly Dynamic Real Time Systems. DRTS: Dynamic Real Time Systems. RTCS: Real Time Complex Systems. FM: Flexible Manufacturing.

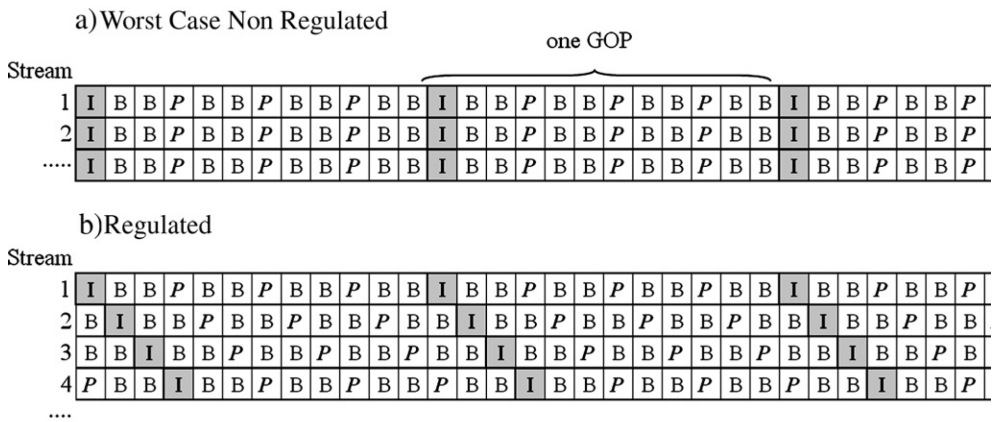

Fig. 2. GOP distribution.

restrictions, which require the use of a real-time nucleus in each node.

The FTT paradigm has been well analyzed in CAN, WorldFIP and Ethernet. In our opinion, the protocols based on tokens play a very important role in factory communications. The dynamic adjustment in token-based networks has been analyzed in [40] using IEEE 802.4 as a network. However, this approach is based on the use of Fuzzy Logic, Genetics Algorithms and Neural Networks to adjust dynamically the different timers and queues of this MAC, instead of adapting through scheduling the actual traffic. However, in current token industrial networks, like Profibus, these parameters have to be fixed and equal in all nodes.

3. Multimedia integration

3.1. Video compression

Video compressors exploit the spacial and temporal redundancy of the frames in order to achieve greater compression rates than the compression techniques of still images. The compressors, such as the MPEG2 or MPEG4 [13] use three types of frame to codify the sequences: frames type I are coded as independent frames, without using any information from the preceding frames; frames type P are constructed from the previous I or P frames; and the frames type B are constructed from the two nearest I or P frames, one before and the other after.

In this context the IPB pattern is denominated GOP, and is repeated continually. For example, in PAL[3] MPEG2/MPEG4 streams, a usual pattern is IBBPBBPBBPBB. One parameter that influences the requirements of the bandwidth is the acquisition-transmission period of frames (T_{AP}). Usually, in conventional video, 25 (PAL, T_{AP}=40 ms) or 29,97 (NTSC[4], T_{AP}=33.3 ms) frames per second are used [41]. The distance between two type I frames is the Keyframe interval (ki) measured in number of frames (ki=12 in the PAL). As each frame is acquired each T_{AP}, then the ki duration will be $T_{ki}=ki*T_{AP}$. This parameter must not be excessively high since the loss of a fragment of an I frame will involve a significant deterioration in the whole GOP up to the following I frame.

Finally, the codec configuration parameters will determine the quality, the size of the I, P and B frames, the I:P:B ratio as well as their distribution. The present work assumes a constant and even distribution of the frames in the GOP, as well as the size relation within the GOP. This hypothesis, which is incorrect for some environments [42], is valid for the scenario considered. This is basically due to the use of symmetrical compression codecs (asymmetrical codifiers, due to the high computational cost that they use in the codification, can achieve greater compression rates, analysing the scenes, making modifications in the GOP structure in function of changes of scene, etc.) and industrial monitoring scenes which present a moderate and know variability [18].

3.2. Video streaming

The transmission of GOPs produces a fluctuation in the demand for bandwidth and is not appropriate for networks or for token MAC protocols [43,44]. Video coding is generally VBR[5] rather than CBR[6] thus obtaining a better, more constant quality. Resources are allocated based on the peak bit rate of the stream, but such over-provisioning is extremely wasteful and undermines the benefits of a constant-quality encoding. The video sources could compete for the highest bandwidth requirements at the same time, the worst case being when all the sources want to transmit an I frame at the same time (non-regulated [43], see Fig. 2a). This creates the need for effective techniques to deal with burstiness.

3.3. Characteristics of industrial monitoring

The end destination of monitoring streams is a manufacturing operator who uses the information for his work. Therefore we have established the priorities in the following order [18]:

(1) Resolution or size of frames. Many theoretical streaming works use very low resolution images, thus the operator can use this information efficiently.

[3] Phase Alternating Line.
[4] National Television System Committee.
[5] Variable Bit Rate.
[6] Constant Bit Rate.

(2) Transmission rate. It is assumed that the operator is performing his task and using the information punctually to do his work. Therefore, it is not necessary to have a QoS parameter in this area or to provide the sensation of real movement, and so slower transmission rates than in conventional video can be used even going far below the 15 frames/s determined for many studies in this area [45].
(3) Compression quality. The quality of compression will influence the quality of the image and the resources necessary for its transmission.

As a consequence of these priorities we do not consider the use of bandwidth smoothing techniques. The smoothing reduces the burstiness transmitting data with fixed rates, and reconstructs the VBR based on receiver buffer [46]. However, due to the lower transmission rate of an industrial environment, this could introduce an excessive delay from acquisition to decodification for a real-time monitoring system. The same reasoning could be applied to the use of frames type B, since we have to wait for future frames to codify it. Then, we integrate a regulated scheme ([43], see Fig. 2b) of streams with frames type I and P in a FTT paradigm.

3.4. Previous work

One of the first papers on this integration was written by Shin. In his work [47] a new methodology for SP-50 fieldbuses, and their interconnection is proposed. However, it is only evaluated as a multimedia network, without integration with control traffic. Also the type of streams used are not representative for industrial applications [18]. The same criticism can be applied to Ritcher's work [48], although in this case over a monomaster architecture with Profibus. Cavalieri [16] analyzes the impact of substituting a centralised architecture for a control system distributed in a robot fruit picker, studying its hardware and software impact.

Tovar et al. analyze the integration of multimedia traffic through the incorporation of a TCP/IP over the data link layer of fieldbuses. However the configuration must be realised off-line and, as the authors conclude, is inefficient for VBR type traffic [44].

Kee-Yin [43] proposes to regulate the transmission of the different I frames (see Fig. 2b) in Timed Token networks. He analyzes in a theoretical form the improvement that its application would have (from 21 to 59% depending on the type of traffic). However, its application to only a single source node is shown.

The authors analyze in [15] the integration of control and image processing applications over Profibus, although using off-line scheduling. Through the low priority traffic profile, the token overruns are avoided, obtaining the best use of the available bandwidth, although with fixed static window sizes.

4. FTT-Profibus

Profibus is one of the most popular fieldbus standards, being one of the three standards belonging to EN 50170 since the beginning of this standard, and also the same with respect to IEC 61158. It has been on the market since 1995 and has a 20% share of the market, with more than 500,000 plants and 10 Millions of nodes [49]. This work is done using the Profibus (EN 50170-2 and IEC 61158 type1/3/10) FDL (Fieldbus Data Link) services SDA and SDN, and the level architecture shown in Fig. 3. The underlayer multimedia protocol and the use of SDA, SDN and FDL frame formats are defined in paper [50]. The functioning of the FTT-Profibus dispatching level is analyzed in this section.

4.1. Introduction

The proposed architecture of flexible scheduling denominated FTT-Profibus is hybrid. Although a central node that governs the state of the system also exists, contrary to other FTT protocols where it is necessary to explicitly control the instant of transmission of each message or the set of messages that could be transmitted in each EC (by TM transmission), the SM only has to carry out the admission test, the scheduling and the cycle test where all the components of the system are found. This hybrid operation is due to the use of the underlaying token passing mechanism, and so part of the responsibility of the transmission of messages is distributed to the rest of the network nodes, simplifying the function of SM and reducing the bandwidth necessary for the management of the protocol. The scheduling protocol is table driven, and also uses the EC and MC cycles. As FTT approaches, the bus is slotted in consecutive fixed duration Elementary Cycles.

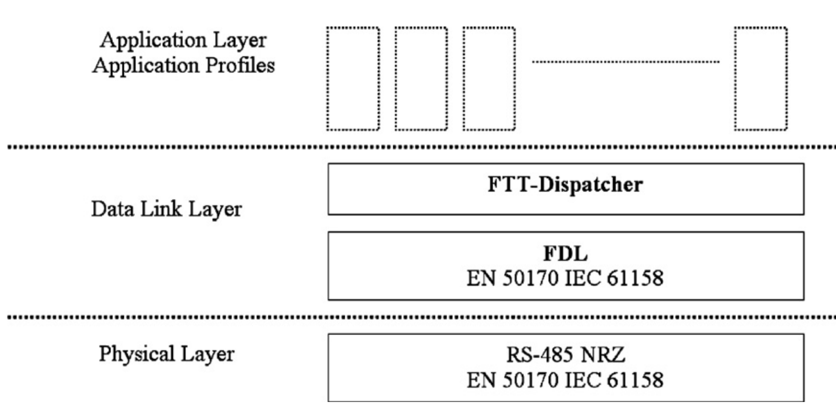

Fig. 3. Standard relationship.

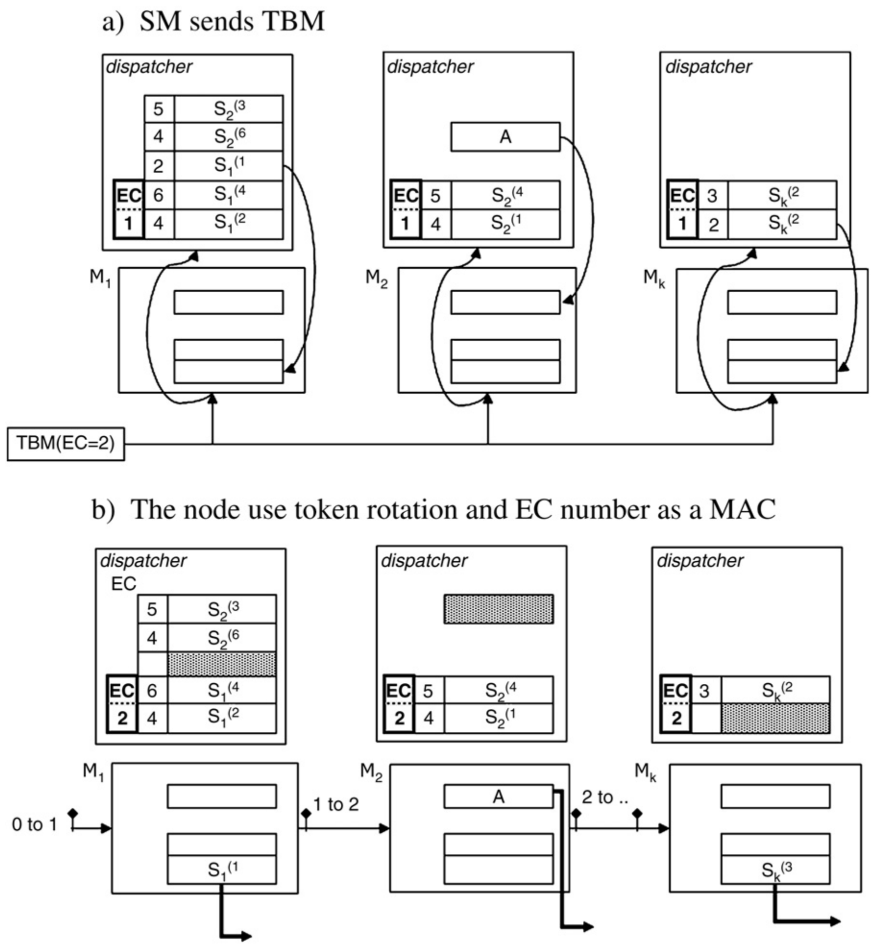

Fig. 4. TBM decodification. a). SM sends TBM. When EC 1 finishes, the SM sends a coded broadcast message that the new EC is EC 2. Then, the nodes have to decode this message, and dispatch the messages associated with this EC. b). The nodes use token rotation and EC number as a MAC. After TBM, the SM release the token. Node 1 sends its EC 2 pending messages (S_1^1) and, as there aren't asynchronous messages, releases the token. Node 2 doesn't have synchronous messages, but there is a pending asynchronous message that will be sent. After that, the token is also released.

4.2. Multimedia integration

The characteristics which define video traffic have been explained in Section 3. Of these, two have special relevance to scheduling video traffic, given that these define the frame transmission period (determined by T_{AP}), and the frequency with which the maximum transmission capacity (determined by ki) is needed.

In order to assign bandwidth to this type of traffic, it is necessary to know beforehand the basic multimedia parameters, such as the period in which the traffic pattern is repeated and the I–P frame distribution. As we analyzed in [18] this is not a problem in classic industrial monitoring. We propose a new cycle, denominated the video cycle period (VC), it is this which determines the number of EC's in which the transmission of a GOP is contained. Its duration is equal to the T_{ki}. We define its duration as a T_{EC} number as:

$$T_{VC} = T_{ki} = ki * T_{AP} = ki * m * T_{EC} \tag{2}$$

We propose to use a T_{VC} multiple of the T_{MC}, so the T_{AP} and T_{ki} must be pre-selected to meet this requirement. This guarantees that the available bandwidth for each frame is the same in each GOP, thus simplifying its scheduling. Also the bandwidth assignment is simplified if T_{AP} is a multiple m of T_{EC}. Thus T_{MC} determines the period in which the periodic traffic is repeated, while the T_{VC} determines the same but for video traffic.

In table driven scheduling, as has been previously mentioned, two different windows are usually distinguished for bandwidth assignment (SW and AW) to each type of traffic. The size of SW in Profibus is defined by the peak of the synchronous traffic in a MC, so that in all the EC's this type of traffic has guaranteed transmission, satisfying the temporal requirements. The rest of the time until completion of the EC is reserved for AW. In the FTT protocols different techniques are used so that the scheduler assigns this asynchronous bandwidth to the nodes through polling until this time is used up.

When the difference between the peak and the trough within a MC is significant, the quantity of bandwidth wasted through this system of fixed windows is considerable [51]. However this problem has not usually been considered since asynchronous real-time traffic has a very small packet length (given that it is usually used for alarm type messages, typically less than 10 bytes). However, to include the video traffic in the transmission, characterized by the need to transmit a large volume of information, this fixed assignment does represent a significant problem.

Therefore, the scheduling algorithm proposed will work with SW of variable size in each EC, permitting generally larger video windows (VW). In this way it will be possible to maximize the number of video streams, their quality and to reduce the number of frames dropped.

4.3. Services/traffic supported

In the model the existence of *ncn*[7] nodes, *p* synchronous real-time control streams in each node and *nmn* [8] nodes with video streams sources is assumed. The total number of nodes is *ns*, being $ns = ncn + nmn$. As it is a FTT system, one node, the $SM(M_0)$, is the one charged with scheduling and synchronization of the rest of the nodes ($M_1 \ldots M_{ns}$). The model is based on the synchronous allocation capacity of Profibus with a constrained low priority profile [15,21]. In the following sections the traffic supported by the model, the management traffic, and the latencies introduced are shown.

4.3.1. Real-Time control synchronous traffic

Synchronous data messages are used to periodically distribute data among nodes. Each node i ($i \le ncs$) has j ($j \le p$) streams of synchronous messages S^j_i characterized by its minimum period $P^j_{m_i}$ and its maximum period $P^j_{M_i}$ ($P^j_{m_i} \le P^j_{M_i}$), both values being harmonic multiples of the T_{EC}. This range of multiples of T_{EC} is the QoS parameter used in the control traffic [52]. It is assumed that each message has a transmission time of C^j_i. For simplicity it is also assumed that each synchronous message has the same length, known as C_S. The Deadline D^i_j will be less than or equal to the P in use.

For each flow $S^j_i = \{P^j_{m_i}, P^j_{M_i}, D^j_i, C_S\}$, it is necessary to reserve the bandwidth needed for a message of length C_S in a determined EC_v of the MC, know as V^j_i. The rank of the valid EC is determined by the period accepted by the scheduler P^j_i and its deadline, being:

$$V^j_i \subset \{(p+0), (p+1), \ldots, (p+P)\} \tag{3}$$

where $p = P^j_i / T_{EC}$, and $P = (P^j_i + D^j_i)/T_{EC} - 1$.

The criteria used to assign the EC is that of minimizing the energy of the periodic traffic pattern. This means using the least busy valid ECs in the MC when the request arrives. Therefore, for each S^j_i it is necessary to calculate in which EC it will be placed:

$$V^j_i = v : \mathrm{SW}^v < \mathrm{SW}^l, \forall l = p, p+1, .., p+P \tag{4}$$

We don't considered priorities or deadlines in the bandwidth assignation. In the classic FTT paradigm, as the scheduling is designed for each EC, and it is codified in the TM at the beginning of each EC, each S^j_i can be transmitted in different positions in each EC, in function of the priorities or deadlines of new synchronous streams. In our approach, we consider lower change frequency situations, and then, we don't codify the scheduling in the TM. This is also more adequate for RTCS, where the number of synchronous streams is higher, and then we can't codify in one TM the scheduling of each EC. This is to reduce the number of responses to be made for each request, since the reallocation in one EC of a previous stream to enable the new stream could have a significant incidence in the next ECs. Thus we sacrifice scheduling flexibility, giving more priority to speed and efficiency.

Therefore, the synchronous bandwidth assigned between the different streams of *i* node in the EC *v* will be:

$$\mathrm{SW}^v_i = \sum_{j=0}^{p} e * C^j_i, \forall v = 1..\mathrm{MC} \tag{5}$$

Where $e=1$, if $V^j_i = ve = 0$, in other casesand then, the synchronous bandwidth assigned between the different node streams in the EC *v* will be :

$$\mathrm{SW}^v = \sum_{i=1}^{ncs} \mathrm{SW}^v_i, \forall v = 1..\mathrm{MC} \tag{6}$$

4.3.2. Video traffic

The video streams M_i are characterised by the frame dimension in X (X_i) and in Y (Y_i), the number of bits per pixel (bpp_i), the acquisition period (T_{AP_i}), the interval between keyframes (T_{ki_i}), and the distribution of frames in the GOP (G_i). For each of the *m* video streams $M_i = \{X_i, Y_i, \mathrm{bpp}_i, T_{\mathrm{AP}_i}, T_{\mathrm{ki}_i}, G_i\}$ it is necessary to reserve the VC video windows of different sizes according to the EC and the type of frame. Therefore, being VW^i_n the space reserved in the EC *i* for the stream *n*, the bandwidth reserved for the stream *n* in VC will be the set:

$$\mathrm{VW}_n = \{\mathrm{VW}^1_n, \mathrm{VW}^2_n, \ldots, \mathrm{VW}^{\mathrm{VC}}_n\} \tag{7}$$

So that all the frames of a GOP can be transmitted before their deadline, which is also T_{AP_i}, the following must be fulfilled:

$$\mathrm{VW}^1_n + \mathrm{VW}^2_n + \ldots + \mathrm{VW}^{\mathrm{VC}}_n \ge \mathrm{CI}_n + np * \mathrm{CP}_n \tag{8}$$

where CI_n and CP_n are the load of the frames I and P of the stream *n*, while *np* is the number of frames of type P determined by the codec and ki.

However, this bandwidth has to be correctly allocated to guarantee the delivery time of each frame. Because each frame can be transmitted in various ECs, in fact between ap ECs, where $\mathrm{ap} = T_{\mathrm{AP}} / T_{\mathrm{EC}}$, the following must be fulfilled:

$$\mathrm{VW}^t_n + \mathrm{VW}^{t+1}_n + \ldots. + \mathrm{VW}^{t+ap}_n \ge C_n \tag{9}$$

being *t* an T_{EC} multiple and C_n the load of the frame I or P in function of the IP distribution of this stream and of the EC in which we find ourselves.

[7] Number of control nodes.

[8] Number of multimedia nodes.

Table 2
Codification of frames

a) Management and request frames

Function	T	C				NB
Initialization	B	0				1
TBM	B	1	EC no.			2
Sync. Req	M	2	N	P_m	P_M	4
Vid. Req.	M	4	N	QR1	QR2	9
Async. BE.	M	8	N	NBW		6
Control	–	16				1
Video	–	32				1

b) RBM

Function	T	C		NB
RBM	B	128	BWA	255
BWA codification				
Traffic type				
Sync.	N	EC		2
Vid.	N	QA	ECI	2
Async. BE	N	EC	NC	2

T: Type. C: Code. NB: number of bytes. B: Broadcast. BWA: Bandwidth assignment. N: number request. ECS: EC no. QR: Quality Requested QA: Quality assigned. ECI: EC number for keyframe, for regulated purposes. NC: Number of C_S assigned in EC.

The window for the video traffic in each EC v will then be the sum of the windows associated with each of the streams:

$$\mathrm{VW}^v = \sum_{i=1}^{nmn} \mathrm{VW}_i^v \tag{10}$$

4.3.3. Real-Time control asynchronous traffic

Real-Time control asynchronous messages are used for alarms and requests for traffic requirement changes from nodes to the SM. As has previously been seen, asynchronous control messages of this type are limited to 10 bytes with a transmission time of C_A. Also, the traffic request messages have a length lower than 10 bytes. [9]

Each one of the *ncs* nodes will be able to generate messages of this type at whatever moment. It is assumed that all asynchronous messages have the same deadline D^{asyn}. With the aim of guaranteeing maximum speed in the delivery of this type of traffic, this deadline will be of the T_{EC}.

In each EC the AW will be enough for the transmission of an asynchronous message in each node. Therefore, in all ECs we have:

$$\mathrm{AW} = \mathrm{AW}^v = \sum_{i=1}^{ncs} C_A, \forall v = 1..\mathrm{MC} \tag{11}$$

4.3.4. Non Real-Time asynchronous traffic

For the non-real-time asynchronous traffic (monitoring and configuration Profibus messages) a Best Effort mechanism will be used, in the free bandwidth of the EC. For these the nodes use asynchronous real-time messages [10] requesting the required bandwidth. The SM will process the messages and in the response message (RBM, see Section 4.3.5) will specify to the requesting node the EC in which it can make use of the channel, and for how long (BW: Best effort Window).

4.3.5. Management traffic

The message related to bandwidth management is transmitted as a broadcast message and is denominated TBM (Trigger Broadcast Message). The SM communicates the EC to all the nodes in the system. It is necessary one C_A in each EC for the management window (MW).

Another aspect related to the scheduling of the traffic is the decoding time of TBM messages in the receptor nodes known as T_{dec} (see Fig. 4). This time includes decoding of the message to know which EC we are in, and also the downloading of the messages from higher layer to MAC buffers level.

On the other hand, given that in the TBM the bandwidth that each node can use is not codified and given that in each EC new traffic requirements can be produced, it is necessary to guarantee a bandwidth to the SM in a way in which it can respond to these requests through a message denominated RBM (Response Broadcast Message, see 2.b), and that each node can recognise if its new request is accepted, and in which ECs it will be able to use bandwidth for this. This message has in the worst case a length of C_S. Therefore:

$$\mathrm{MW} = \mathrm{MW}^v \le C_A + T_{\mathrm{dec}} + C_S, \forall v = 1..\mathrm{VC} \tag{12}$$

4.4. Exclusion window

It is necessary to guarantee the temporal isolation between the windows of an EC, and also between the windows of the next EC's. If the SM releases the token until EC has finished, then it is possible that in one EC token passes are given which were due to the previous EC. This could break the temporal isolation between the ECs causing a loss of messages. To avoid this, the EC will have to retain the token whenever the timer programmed to $T_{\mathrm{EC}} - \Gamma$[11] reaches 0, being $T_{\mathrm{EC}} - \Gamma <$ TSYNI [12] . This doesn't represent a problem given that the SM can recognize at any moment the number of nodes in the ring.

Therefore:

$$\mathrm{EW} \ge \Gamma \tag{13}$$

In this way, in each EC the following must be fulfilled:

$$\mathrm{EW} + \mathrm{MW} + \mathrm{AW} + \mathrm{SW}^v + \mathrm{VW}^v + \mathrm{BW}^v + \Gamma \le EC \tag{14}$$

5. Protocol

This section is based on the constrained low-priority traffic profile, which guarantees the necessary bandwidth allocation

[9] See Table 2a, codes 2, 4 and 8.

[10] See Table 2a, code 8.

[11] Γ is the token rotation time.

[12] TSYNI: Synchronization Interval Time. Maximum time that the nodes wait to detect and IDLE period in the bus, before considering that an error has been produced.

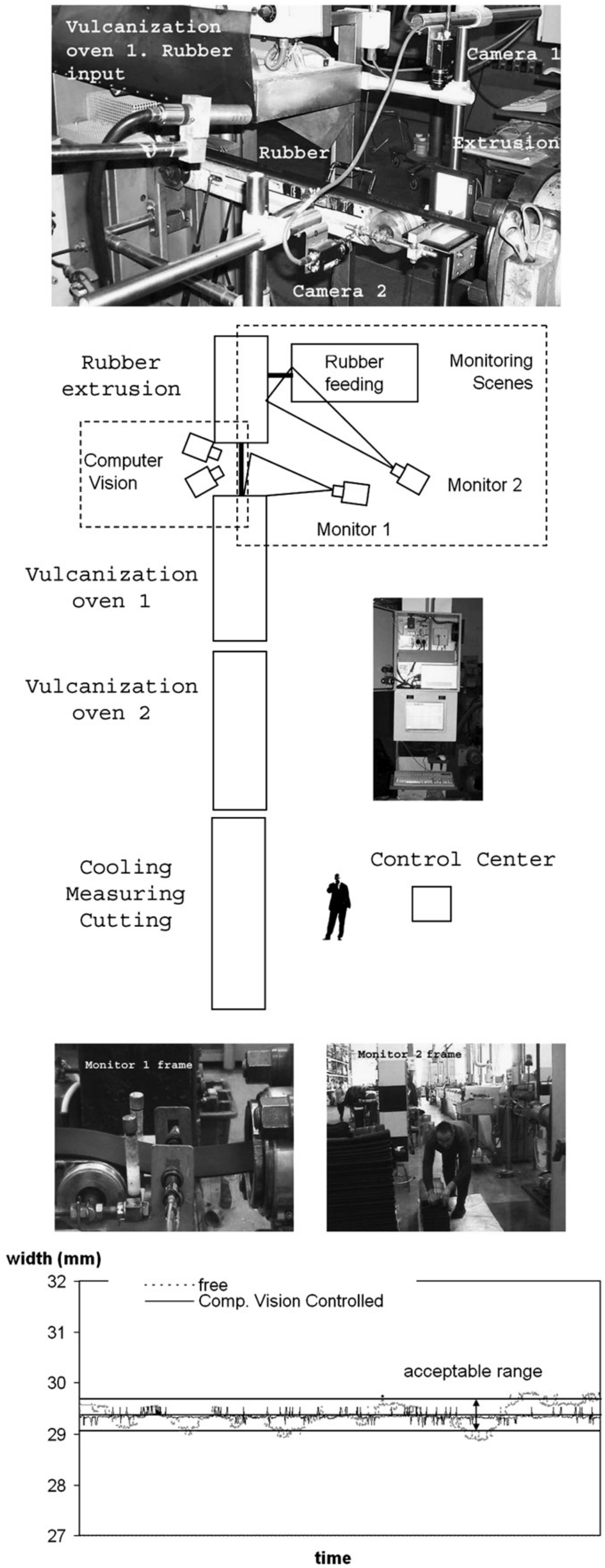

Fig. 5. Real scenario diagram.

Table 3
Multimedia streams properties and requirements

a) Properties

Stream	Quality	Properties
$M_{1,2}$	1	{600,22,8,6,1,0}
	2	{600,16,8,3,1,0}
$M_{3,4}$	Codec Function	{360,288,16,6,6,IPPPPP}

b) Requirements

Stream	Quality	CI	CP
$M_{1,2}$	1	60	
	2	40	
M_3	High	38 (9000 bytes)	3 (675 bytes)
M_3	Low	23 (5382 bytes)	2 (272 bytes)
M_4	High	54 (12871 bytes)	6 (1377 bytes)
M_4	Low	32 (7548 bytes)	3 (666 bytes)

scheme over Profibus. This has been previously presented by Tovar [21] and has been adapted and implemented by the authors for image transmission [51], being also the basis of other analyzes of real applications such as those analyzed by the authors [15] (for a detailed description of the Profibus FDL services used, see [50]).

One of the priorities of the protocol has to be to allow the desired flexibility in the scheduling, minimizing the bandwidth required and giving the shortest possible response times. The protocol also differentiates between the services with QoS guaranteed (the synchronous and asynchronous control messages) and the differentiated services used for multimedia.

5.1. Initialization

In the set-up phase of the system, the SM transmits a broadcast message code 0 (see Table 2a) indicating the initialization state to the rest of the nodes and releases the token. As this reaches the other nodes, these make their request to the SM using the transmission of request messages (code 2, Table 2a) .

As the SM executes the scheduling algorithm, each request S_i^j is assigned, according to the arrival order, so it is a First Come First Served (FCFS) system. One C_S is assigned in the EC determined by the $P^j_{m_i}$, $P^j_{M_i}$ and the bandwidth available in the ECs involved. The SM uses RBM messages to communicate to the request source the information about the EC in which it has assigned bandwidth.

5.2. Run-time

When the initialization phase is finished, and bandwidth assignment for real-time control synchronous traffic has been done according to the request, the SM enables transmission through the broadcast of TBM. This message, which functions in a similar way to the trigger message (TM) of the FTT seen earlier, has the advantage of not having to indicate the set that each master can transmit in this cycle. This allows a reduction in the necessary bandwidth, also simplifying the message decoding in the nodes, thus reducing the wasted bandwidth. Nevertheless, reserved bandwidth for another message exists; the response message to the request from the nodes. As both messages fit in a single message, with the object of reducing bandwidth use, they are coded in one message, thus eliminating the intergap time between these messages.

5.3. Video request

Once the initialization has finished, the monitoring node might wish to visualize video flows in the network with a determined quality. To do this, it transmits request messages type 4 to SM, indicating the direction of the video source, the number of the associated requests and the maximum and minimum quality required.

The scheduler will have to analyze if it is possible to deal with the request with the qualities required. If this is not possible, various guidelines of behavior can be established in the SM:

- FCFS: The qualities of the previous streams are maintained, not admitting the new request.
- Quality adjustment reduction. The qualities of the previous streams are reduced to be able to admit the new streams request. It is possible to realize a homogeneous reduction of the quality in all the streams or realize a heterogeneous reduction according to the priorities.

Table 4
Initial messages set

a) Synchronous

Node 1	$S_1^1=(1,1)$	$S_1^2=(2,4)$	$S_1^3=(2,4)$
	$S_1^4=(2,4)$	$S_1^5=(4,8)$	$S_1^6=(8,8)$
Node 2	$S_2^1=(1,1)$	$S_2^2=(4,8)$	$S_2^3=(8,8)$
Node 3	$S_3^1=(2,4)$	$S_3^2=(8,8)$	
Node 4	Node 5	Node 6	
$S_4^1=(2,2)$	$S_5^1=(4,8)$	$S_6^1=(8,8)$	

b) Video stream requirements

Quality	$CI_{(m}$	$CP_{(m}$
High quality	7072 ($30C_S$)	3037 ($13C_S$)
Low quality	4715 ($14C_S$)	2024 ($6C_S$)

The SM will answer in the RBM of the following EC with the information of the accepted quality and the bandwidth assigned for each type of frame. It is assumed that the scheduler has information stored about the history of past sessions and/or by configuration on the index relation of requested quality, and bytes required for the I and P frame types. The transmission protocol of the frames has been explained in [50].

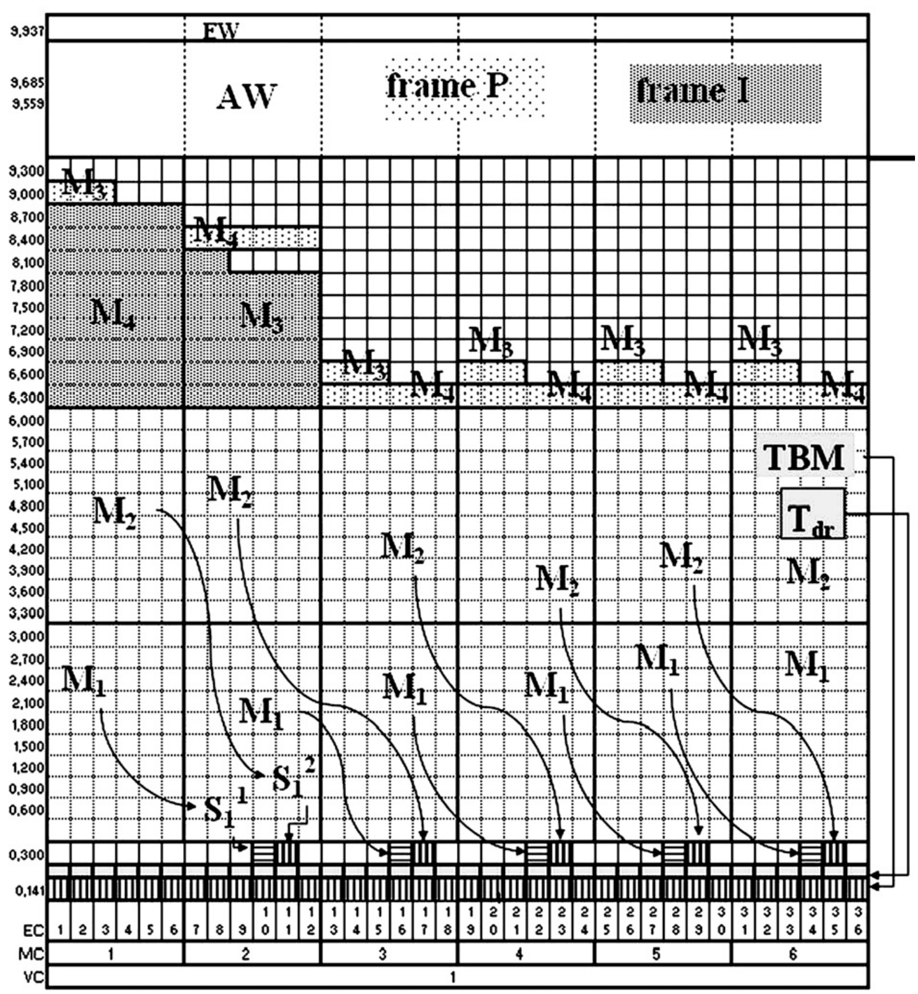

Fig. 6. Traffic distribution in a VC.

5.4. Requests changes

At any moment the nodes can transmit a type 2, 4 or 8 message (see Table 2a) to ask for the transmission of new flows of traffic, change its requirements, or to cancel previous requests.

Faced with these requests, SM has to reschedule the synchronous bandwidths associated with each master and notify the nodes within RBM frame. Given the greater priority of control traffic over video traffic, the control requests have to be attended to with greater speed of reply than the video traffic. This does not create any drawbacks, as each modification in the synchronous traffic involves few changes in the assigned bandwidth owing to the FCFS policy used. In the case of video requests, significant modifications may be involved, and these must be modified in a synchronized form. Furthermore, with control traffic we are on a ms timescale, whereas the operator who monitors the video information is on a seconds scale [45].

6. Analytical numerical examples

6.1. FM real scenario

In this case a real scenario has been chosen where monitoring and computer vision are mixed with control traffic to show the advantages of this integration. The process is the vulcanization section in a rubber factory (see Fig. 5). The process begins with the extrusion of the rubber, after which a conveyor belt carries it into the vulcanization oven. At this point, a vision system

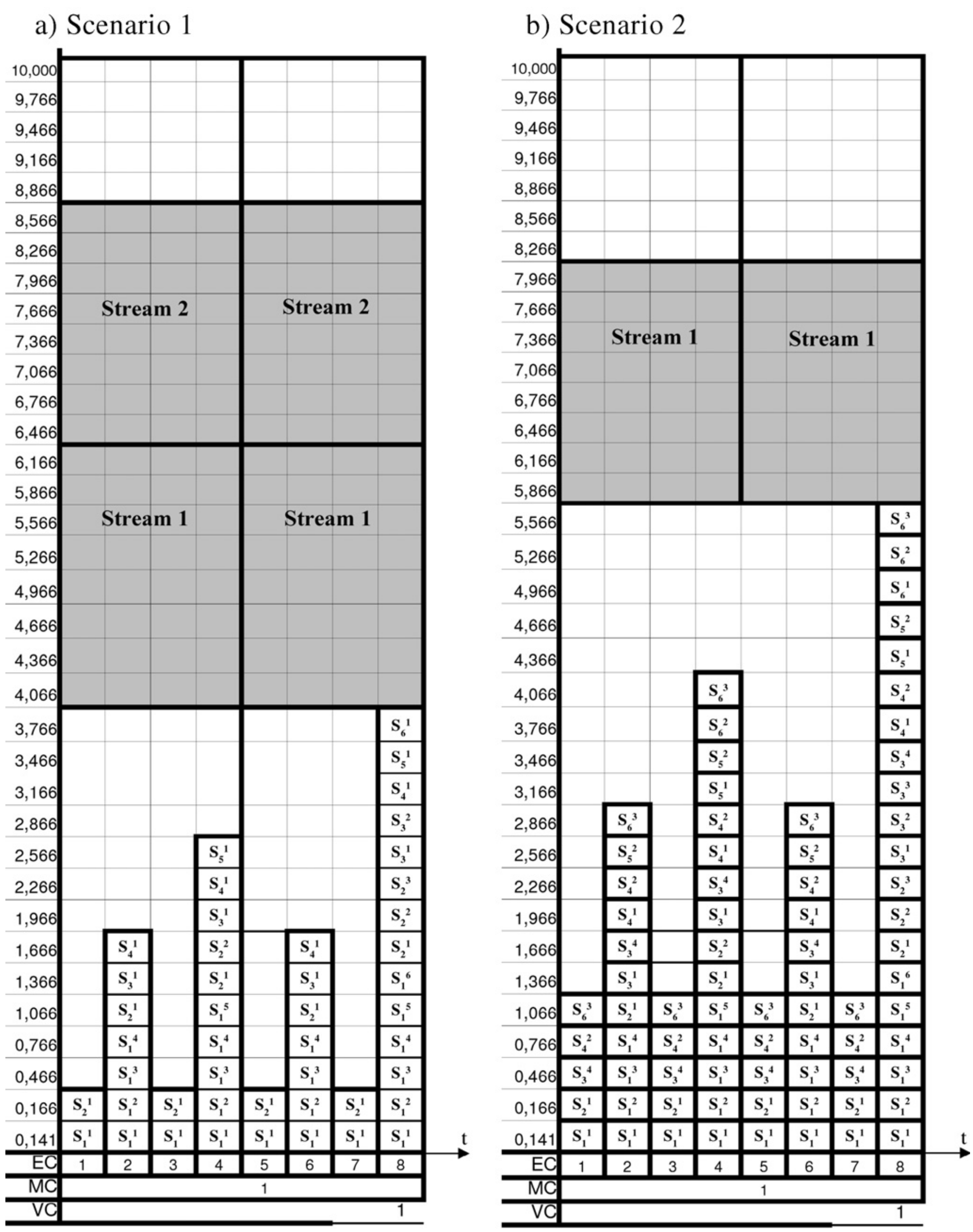

Fig. 7. Profibus distribution.

controls the base and height dimensions of the rubber in order to regulate the extrusion and conveyor belt motors and thus ensure the quality of the product (computer vision output). Depending on the type of piece being produced (something which could change every one or two hours, FM, Table 1), it is possible to work with different parameters of X_i, Y_i (measured in pixels) of gray level (8 bpp) and T_{AP_i} (measured in T_{EC}). As it is a multimedia stream for image processing operating with subpixel resolution, there is no compression; therefore $ki_i=1$ and $G_i=0$. The T_{EC} is set to 10 ms. Therefore, these will be two multimedia streams, $M_{1,2}$ of image processing whose properties and requirements can be seen in Table 3a and b. As a result of the vision process, instructions are sent to the two regulators to control their speed. Thus, as synchronous control traffic we have $S_I^{1,2}=\{3, 6, C_S\}$.

There are also two other areas which would benefit from video monitoring: the exit from the extruder (monitor 1) and the rubber feed into the extruder (monitor 2). Although the operator carries out his work at the end of the process, from the control centre he can monitor all the information gathered by the

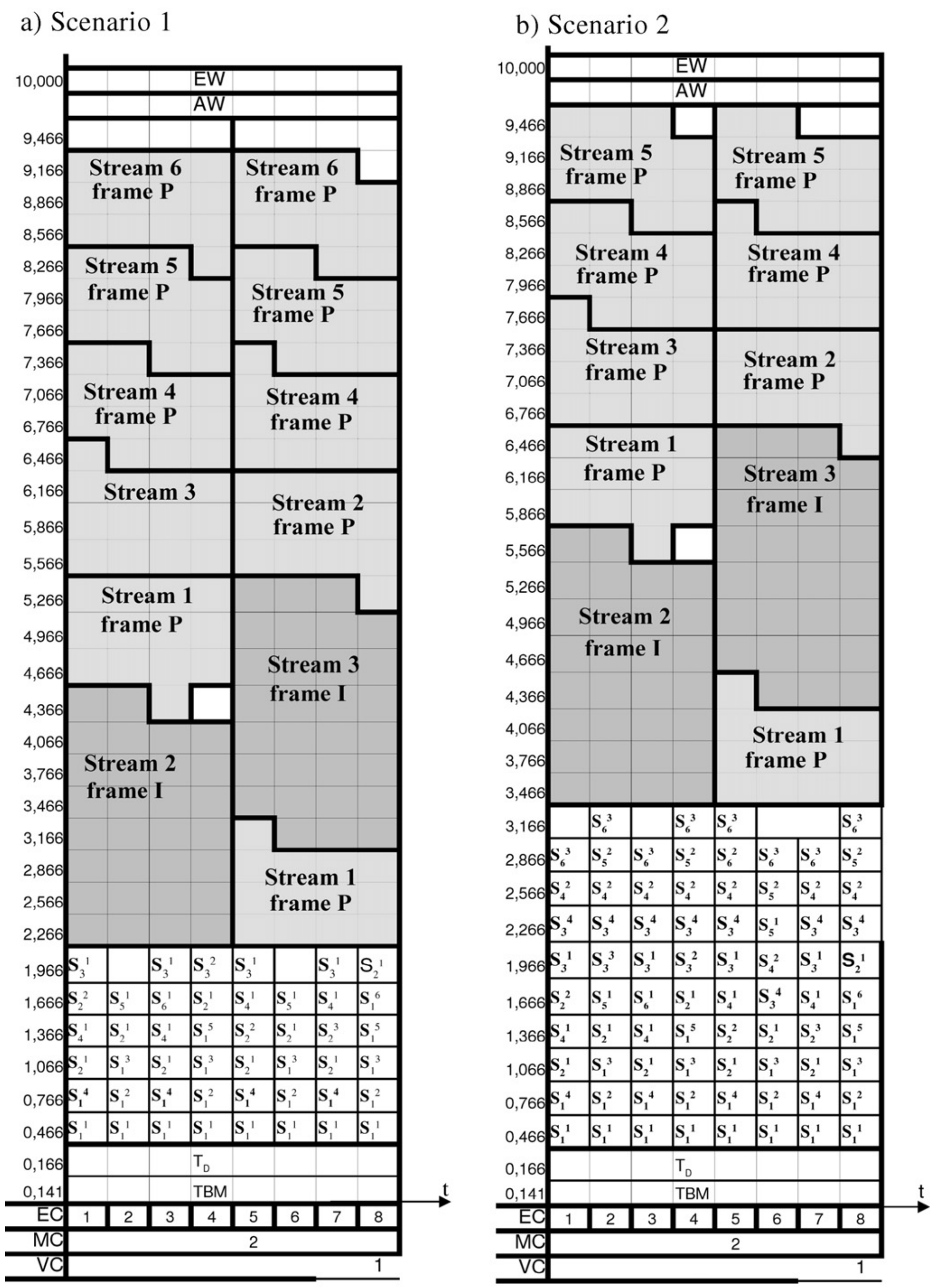

Fig. 8. FTT-Profibus distribution.

multimedia systems. Therefore we have two multimedia streams $M_{3,4}$ with the requirements and properties that can be seen in the same tables for the two qualities used.

In Fig. 6 the traffic distribution in a VC can be seen, which in this case contains 6MCs. In an MC there is a SW and VW traffic capacity of 180 frames. Without regulation a bandwidth for the transmission of 257 packets could be required, exceeding available capacity. Through regulation, the frame I of stream M_4 is sent in MC 1 while the I frame of stream M_3 is done in MC 2. Changing to T_{AP} of 3 T_{EC} would make it necessary to reduce the quality of video streams to low resolution (it would also be possible to reduce the number of lines to 12 and maintain the quality of the video streams). The use of a T_{EC} of 10 ms instead of 30 ms means a small overload in the management traffic, as it requires three times the number of TM messages than in the other way. However, the overload produced is negligible while the advantages are reduced jitter as well as better response time for real-time asynchronous request.

6.2. RTCS simulated scenario

This example begins with a T_{EC} of 10 ms. and the group of control messages in a network with 6 masters as can be seen in Table 4a, where P_m and P_M are expressed in T_{EC} units, that is (P_m, P_M). This means a TM C of 80 ms.

For the video streams, type IV streams [43] are used, which are characterized by an approximate I:P ratio of 10:4, and ki = 12. The average sizes of the frames are shown in bytes and in C_S in the Table 4b.

The transmission of 12 streams (25 frames/s, $T_{\mathrm{AP}} = 40$ ms) in the worst case in which the 12 sources try to transmit the frame type I at the same moment would require a bandwidth of 17 Mbps. Avoiding the overlapping of these I frames, the bandwidth is reduced to 4.2 Mbps. That is to say that the necessary bandwidth is reduced by 75%. In Fig. 7a we can appreciate how in a static assignment scheduler like dynamic Best Effort of Profibus, only 2 of the high quality streams could be transmitted. Given that in Profibus, the sources don't have information from the EC, it is necessary to reserve for each cycle a fixed and constant space, without being able to differentiate between I and P frames. This leads to a significant waste of bandwidth. In Fig. 8a we can see how, with a dynamic and regulated schedule, it is possible to transmit 6 streams with high quality.

If in time the control traffic characteristics are modified in the following way: these flows are added $S_3^{(3} = \{8,9\}$, $S_3^{(4} = \{1,1\}$, $S_4^{(2} = \{1,1\}$, $S_5^{(2} = \{2,4\}$, $S_6^{(2} = \{4,4\}$, $S_6^{(3} = \{1,2\}$, with this periodic traffic scenario, in a static assignment scheduler, only 1 of the high quality streams could be transmitted (Fig. 7b). With the proposed scheduling, in order to satisfy the new requirements and given that in all the ECs the SW size grows, the system will have to reduce the size of the VW in each EC. As a consequence of this, with high quality we can have 5 streams being continuously transmitted without interfering with the control traffic and maintaining the reserves made for asynchronous traffic and the management of the protocol. Reducing the quality of the video in this scenario, 11 streams could be transmitted, or 12 if in turn the control traffic QoS were reduced.

In this way, in the scenarios with a heavy load it is possible to transmit up to 3 times the number of streams over Profibus, while if the level of traffic is extremely high up to 5 times the number of streams are possible.

7. Conclusions

In this paper an analysis has been made of the different situations that may be required of flexible scheduling mechanisms, giving a classification of such situations. After studying previous works and making a description of the particular characteristics introduced by video traffic, the FTT-Profibus and the type of traffic carried is presented. FTT-Profibus is an implementation of the FTT paradigm over industrial timed token networks, which is particularly well suited for RTCS and DRTS. Lastly, the protocol is detailed and finally two examples are presented. In the first FM scenario the control, computer vision and monitoring traffic is integrated in the same infrastructure. This is done guaranteeing their coexistence without interferences. The change in requirements in the image processing stream can be dynamically absorbed by changing the QoS in monitoring traffic (FM). Moreover, the non regulated traffic, with a peak bandwidth requirement of 8.1 Mbps (257 packets each 60 ms), unschedulable in Profibus, can be scheduled through a regulated mechanism with peak traffic requirement of 5.71 Mbps (180 packets each 60 ms) and the low constraint traffic profile. The second is a simulated RTCS scenario, where we can see not only a greater capacity to transmit high quality streams, but also how FTT-Profibus allows better use to be made of the bandwidth, provides a constant jitter, and can adapt to new control traffic request, adapting the quality assigned to multimedia streams.

Acknowledgements

This work was supported by the "Ministerio de Ciencia y Technologia" of Spain under the project TIC2003-08129-C01, which is partially funded by FEDER.

References

[1] J. Stankovic, et al., Strategic directions in Real-Time and embedded systems, ACM Computing Surveys 28 (4) (1996) 751–763.

[2] G. Fohler, G.C. Buttazzo, Introduction to the special issue on flexible scheduling, Real-Time Systems 22 (2002) 5–7.

[3] D. Dietrich, T. Sauter, Evolution potentials for the fieldbus system, Proc. of WFCS'00, IEEE Int. Workshop on Factory Communications Systems, Porto, 2000, pp. 343–351.

[4] J.P. Thomesse, The Fieldbuses, Annual Reviews in Control 22 (1998) 35–45.

[5] J. Ferreira, P. Pedreiras, L. Almeida, J.A. Fonseca, The FTT-CAN protocol for flexibility in safety-critical systems, IEEE Micro, 2002, pp. 42–55.

[6] H. Kopetz, The Time-Triggered (TT) Model of Computation, Proc. of RTSS'98 (19th IEEE Real-Time System Symposium, Madrid), 1998, pp. 168–177.

[7] J.P. Thomesse, Main paradigms as a basis for current fieldbus concepts, Proc. of FeT99, IFAC Fieldbus Systems and their Applications, Magdeburg, 1999.

[8] J.P. Thomesse, Fieldbus technology in industrial automation, Proceedings of the IEEE 93 (6) (2005) 1073–1101.

[9] L. Almeida, P. Pedreiras, J.A. Fonseca, The FTT-CAN Protocol: why and how, IEEE Transactions on Industrial Electronics 49 (6) (2002) 1189–1201.

[10] H. Kopetz, Composability in the Time-Triggered Architecture, Proc. of SAE International Congress and Exhibition (Detroit,2000).

[11] C.H. Wu, J.D. Irwin, F.F. Dai, Enabling multimedia applications for factory automation, IEEE Transactions on Industrial Electronics 48 (5) (2001) 913–919.

[12] J.R. Pimentel, Industrial multimedia systems, IES Industrial Electronics News Letter 45 (1998).

[13] C.H. Wu, J.D. Irwin, Multimedia and multimedia communication: a tutorial, IEEE Transactions on Industrial Electronics 45 (1) (1998) 4–14.

[14] A.C. Weaver, Survey of industrial information technology, Proc. of IES'01 (27th Annual Conf. of the IEEE Industrial Electronics Society, Denver), 2001, pp. 2056–2061.

[15] V. Sempere, J. Silvestre, Multimedia applications in industrial networks: integration of image processing in Profibus, IEEE Transactions on Industrial Electronics 50 (3) (2003) 440–448.

[16] S. Cavalieri, A. Di Stefano, O. Mirabella, Impact of fieldbus on communication in robotic systems, IEEE Transactions on Robotics and Automation 13 (1) (1997) 30–48.

[17] A. Guha, A. Pavan, J.C. Liu, B.A. Roberts, Controlling the process with distributed multimedia, IEEE Multimedia 2 (2) (1995) 20–29.

[18] J. Silvestre, V. Sempere, T. Albero, Industrial video sequences for network performance evaluation, Proc. of WFCS'04 (5th IEEE Workshop on Factory Communication Systems, Vienna), 2004, pp. 343–347.

[19] E. Tovar, L. Pinho, M. Alves, F. Pacheco, Bringing industrial multimedia to the factory-floor: what is at stake with RFieldbus, Proc. of FeT03 (5th IFAC Int. Conf. on Fieldbus Systems and their Applications, Aveiro), 2003, pp. 131–138.

[20] N. Malcom, W. Zhao, The Timed-Token Protocol for Real-Time communications, Computer 27 (1) (1994) 35–41.

[21] E. Tovar, F. Vasques, Real-Time fieldbus communications using Profibus networks, IEEE Transactions on Industrial Electronics 46 (6) (1999) 1241–1251.

[22] K. Ramamritham, J. Stankovic, Scheduling algorithms and operating systems support for Real-Time systems, Proceedings of the IEEE 82 (1) (1994) 55–66.

[23] J.D. Decotignie, P. Pleineveaux, A survey on industrial communication networks, Annales des Telecommunications 48 (9) (1993) 435–444.

[24] L. Almeida, Flexibility, Timeliness and Efficiency in Fieldbus Systems: the DISCO Project, Proc. of ETFA01 (8th IEEE Int. Conf Emerging Technologies and Factory Automation, Sant Joan Les Pins), 2001, pp. 159–167.

[25] P. Pedreiras, L. Almeida, P. Gai, The FTT-ethernet protocol: merging flexibility, timeliness and efficiency, Proc. of ECRTS'02 (14th Euromicro Conference on Real-Time Systems,Dortmund), 2002, pp. 134–142.

[26] P. Pedreiras, L. Almeida, P. Gai, G. Buttazzo, FTT-Ethernet: a platform to implement the Elastic Task Model over message streams, Proc. of WFCS'02 (4th IEEE Workshop on Factory Communication Systems, Vteras), 2002, pp. 225–232.

[27] J.D. Decotignie, Ethernet-Based Real-Time and industrial communications, Proceedings of the IEEE 93 (6) (2005) 1102–1117.

[28] P. Doyle, Introduction to Real-Time Ethernet II, The Extension. A Technocal Supplement to Control Netwot, July–August, 5(4) 2004.

[29] P. Pedreiras, P. Gai, L. Almeida, G. Buttazzo, FTT-Ethernet: A Flexible Rea-Time Communication Protocol That Supports Dynamic QoS Management on Ethernet-Based Systems, IEEE Transactions on Industrial Informatics 1 (3) (2005) 162–172.

[30] A. Bonaccorsi, L. Lo Bello, O. Mirabella, A. Pchmann and P. Neumann, A Distributed Approach to achieve predictable Ethernet access control in industrial, Proc. of FeT'03 (Int. Conf. IFAC Fieldbus Systems and their Applications, Aveiro, 2003).

[31] T. Pop, P. Eles, Z. Peng, Schedulability Analysis for Distributed Heterogeneous Time/Event Triggered Real-Time Systems, Proc. of ECRTS'03 (15th Euromicro Conference on Real-Time Systems, Porto), 2003, pp. 257–266.

[32] H. Kopetz, A. Ademaj, P. Grillinger, K. Steinhammer, The Time-Triggered Ethernet (TTE) Design, Proc. of ISORC'05 (8th IEEE Object-Oriented Real-Time Distributed Computing, Seatle), 2005, pp. 22–33.

[33] S. Ho Hong, Implementation of a bandwidth allocation scheme in a token-passing fieldbus network, IEEE Transactions on Instrumentation and Measurement 51 (2) (2002) 246–251.

[34] L. Almeida, E. Tovar, J.A. Fonseca, F. Vasques, Schedulabitity analysis of Real-Time Traffic in WorldFIP networks: An Integrated Approach, IEEE Transactions on Industrial Electronics 49 (5) (2002) 1165–1174.

[35] Y. Kim, S. Jeong, Pre-Run-Time Scheduling method for Distributed Real-Time Systems in a FIP Environment, Control Engineering Practice 6 (1) (1999) 103–109.

[36] E. Tovar, F. Vasques, Engineering Real-Time Applications with WorldFIP: Analysis and Tools, Control SICICA 2000.

[37] S. Cavalieri, A. Di Stefano, O. Mirabella, Pre-Run-Time scheduling to reduce schedule length in the fieldbus environment, IEEE Transactions on Software Engineering 21 (11) (1995) 865–880.

[38] S. Ho Hong, Bandwidth allocation scheme for cyclic-service fieldbus networks, IEEE Transactions on Mechatronics 6 (2) (2001) 197–204.

[39] L. Ferreira, S. Machado, E. Tovar, Scheduling IP traffic in multimedia-enabled profibus networks, Proc. of the ETFA'2001 (8th IEEE Int. Conf. on Emerging Technologies and Factory Automation, Antibes - Juan les Pins), 2001, pp. 169–176.

[40] S. Lee, S. Ho Lee, K. Chang Lee, M. Hyung Lee, F. Harashima, Intelligent performance management of networks for advanced manufacturing systems, IEEE Transactions on Industrial Electronics 48 (4) (2001) 731–741.

[41] H. Heising, K.U. Barthel, Internet still image and video formats, in: R. Zurawsky (Ed.), The Industrial Information Technology Handbook, CRC Press, 2004.

[42] D. Isovic, G. Fohler, L. Steffens, Timing Constraints of MPEG-2 Decoding for High Quality Video: Misconceptions and Realistic Assumptions, Proc. of the ECRTS03 (15th Euromicro Conference on Real-Time Systems, Porto), 2003, pp. 73–82.

[43] J. Kee-Yin, MPEG Transmission schemes for a Timed Token Medium Access Control Network, ACM SIGCOMM Computer Communication Review 29 (1) (1999) 66–80.

[44] L. Ferreira, E. Tovar, QoS of IP services in a fieldbus network: on the limitations and possible improvements, Proc. of ECRTS02 (14th Euromicro Conf. on Real-Time Systems, Vienna), 2002.

[45] A.A. Lazar, R. Stadler, On reducing the complexity of management and control of broadband networks, Proc. of Workshop on Distributed Systems: Operations and Managements, 1993.

[46] W. Feng, J. Rexford, Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video, IEEE Transactions on Multimedia 1 (3) (1999) 302–313.

[47] Kang G. Shin, Chih-Che Chou, Design and evaluation of Real-Time communication for fieldbus-based manufacturing systems, IEEE Transactions on Robotics and Automation 12 (3) (1996) 357–367.

[48] K.R. Ritcher. Modeling of an image transmission system with integrated Profibus-slave, Project Report. Computer Engineering, Technical University of Braunschweig. 1998.

[49] U. Jecht, W. Stripf, P. Wenzel, Profibus: open solutions for the world of automation, The Industrial Communication Technology Handbook, CRC Press, 2005, [Chapter 10].

[50] V. Sempere, J. Silvestre, Image transport in profibus networks, Proc. of FeT'01 (IFAC Fieldbus Systems and their Applications, Nancy), 2001, pp. 24–31.

[51] J. Silvestre, V. Sempere, M. Montava, Optimization of the capacity of PROFIBUS for the transmission of images and control traffic, Proc. of WFCS'02 (4th IEEE Workshop on Factory Communication Systems, Vteras), 2002, pp. 133–140.

[52] G.C. Buttazzo, G. Lipari, M. Caccamo, L. Abeni, Elastic scheduling for flexible workload management, IEEE Transactions on Computers 51 (3) (2002) 289–302.

Víctor M. Sempere (S'93−M'98) was born in Alcoy, Spain, in 1964. He graduated in Electronics and Computer engineering and received the Ph.D. in Telecommunications engineering from the Technical University of Valencia (UPV), Valencia, Spain in 1987, 1993 and 1998 respectively. Currently, he is Assistant Professor in the Department of Communications (UPV), where he teaches Industrial Communications and Computer Networks. His current research interests are Factory Communications and Real Time Communications in Public Networks. He has served as a program committee member for several conferences in the area of Factory Communications. Since 1996, he has authored or coauthored more than 30 technical papers.

Javier Silvestre received his MSc/PhD. degree in Computer Science from Technical University of Valencia (UPV), Spain, in 1999 and 2004 respectively. Currently, he is Assistant Professor in the Department of Computer Architecture (UPV), where he teaches computer architecture and Network technologies. He has served as a program committee member for several conferences in the area of Industrial Informatics. Since 2001, he has authored or coauthored more than 20 technical papers in journals and international conferences. His current reseach interests include real-time communications systems, scheduling, fieldbuses and image and video transmission over control networks.