QoS of IP Services in a Fieldbus Network: on the Limitations and Possible Improvements

L. Ferreira, E. Tovar

IPP-HURRAY Research Group
Polytechnic Institute of Porto (ISEP-IPP), Portugal
{llf, emt}@dei.isep.ipp.pt

Abstract

This paper focuses on the problem of providing efficient scheduling mechanisms for IP packets encapsulated in the frames of a real-time fieldbus network - the PROFIBUS. The approach described consists on a dual-stack approach encompassing both the control-related traffic ("native" fieldbus traffic) and the IP-related traffic. The overall goal is to maintain the hard real-time guarantees of the control-related traffic, while at the same time providing the desired quality of service (QoS) to the coexistent IP applications. We start to describe the work which have been up to now carried out in the framework of the European project RFieldbus (High Performance Wireless Fieldbus in Industrial Multimedia-Related Environments - IST-1999-11316). Then we identify its limitations and point out solutions that are now being addressed out of the framework of the above-mentioned European project.

1. Introduction

Recent technological developments are pulling fieldbus networks to support a new wide class of applications, such as industrial multimedia applications. Examples of such applications for the industrial environment include video, audio, file transfer, http, etc. These kinds of applications can be supported by the TCP/IP protocol, which is widely used, vendor independent, standardised, and interoperable with almost every operating system [1].

A typical fieldbus network is based on a three-layered structure - physical layer, data link layer and application layer. To enable its use in industrial multimedia applications, the TCP/IP suite of protocols can be integrated with the fieldbus stack, leading to a dual-stack approach.

However, there are some relevant aspects that such integration must take into account. In fact what is inferred from Fig. 1 is that the IP packets are to be encapsulated within fieldbus data frames. Typically this requires that the IP packets are fragmented/de-fragmented. This functionality must be supported by an *IP Adapter Sub-layer* (IPAS), which must be placed between the IP and the Fieldbus DLL. The IPAS must also support mechanisms able to provide the tunnelling of IP traffic between fieldbus nodes that do not have communication initiative (slave nodes).

Another important requirement that must be fulfilled by the approach outlined in Fig. 1 is that the hard real-time guarantees provided to the control-related traffic ("native" fieldbus traffic) are kept. At the same time the proposed approach must also provide the desired quality of service (QoS) to IP applications. To achieve this dual goal it is most probably required to have a sub-layer, to which we call Traffic Manager Sub-layer (TMS), between the Fieldbus DLL and the upper layers (Fieldbus Application Layer and IPAS - not directly the IP).

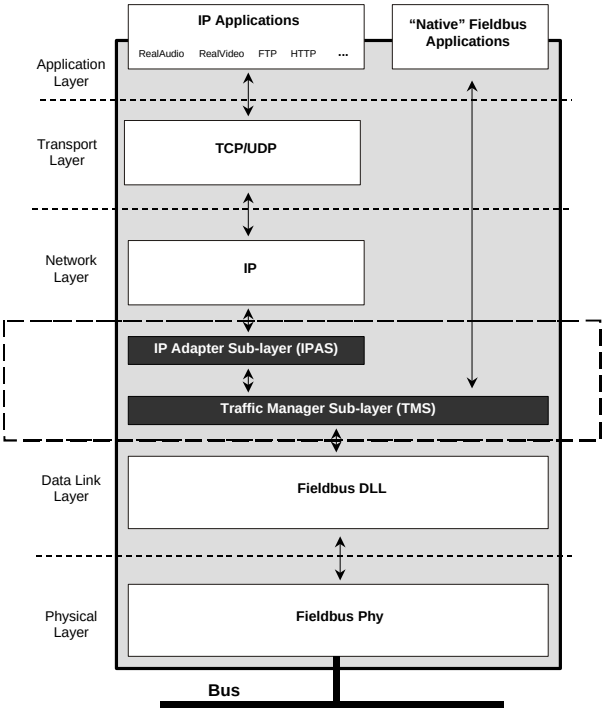

Fig. 1 Integration of TCP/IP into a generic fieldbus stack

2. Some Details on the TMS

The rationale for the TMS was detailed for the particular case of PROFIBUS fieldbus networks in [2]. Due to the characteristics of the PROFIBUS protocol, and in order to support both hard real-time traffic and IP traffic with QoS requirements, we proposed a detailed methodology which is summarised in Fig.2.

The network parameters are set in a way that each PROFIBUS master *i* is able to hold the token for T^i_{SA} time (station allocation). Each type of real-time traffic, either NHP (native high priority fieldbus control-related traffic) or IPH (IP traffic with QoS requirements mapped onto the low priority PROFIBUS FDL services) has a portion of T^i_{SA} (see [3] for some details on how to set these partial allocations). This guarantees, for the case of PROFIBUS networks, that the worst-case token cycle time is bounded to T_{WCCT}. This will be an important notion throughout the

rest of the paper, and will be denoted for scheduling of IP traffic as T_{IPCY}.

We denote the allocation for the IPH traffic as T_{IPH}. It will be used in each token arrival to serve the IPH traffic (PROFIBUS frames containing IP fragments).

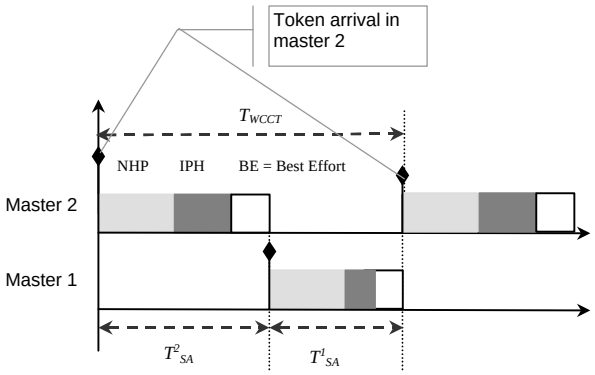

Fig. 2 Traffic allocation example for a 2-master network

Typically there will be a number of IP flows between a particular station and the other stations. These IP flows can have different QoS requirements, namely bandwidth and allowed jitter. Therefore, the TMS must schedule properly the different IP fragments related to the different IP flows.

The crucial guideline for the Scheduler is that it will have to schedule the appropriate T_{IPH} amount of IP traffic to be transferred each T_{IPCY}.

3. A Basic Scheduling Approach

Take as an example the stream set example presented in Table 1.

Table 1

	Periodicity (T_{IPCY})	Transaction Duration (ms)
IPH1	1	1.3
IPH2	2	1.1
IPH3	4	0.3
IPH4	6	0.1
IPH5	6	1.3

An IPH stream is a temporal sequence of message cycles conveying IP fragments. The notion of message cycle results from the underlying fieldbus data link layer. In fieldbus networks, message requests have typically immediate replies. Therefore, a transaction (message cycle) will have a time length corresponding to the time to send the request frame up to completely receive the response frame.

Inherent to Table 1 is the notion of multi-cycle operation [4, 5]. In our case, the primary cycle will be the cycle at which the scheduler will operate (T_{IPCY}), which in turn corresponds to the worst-case token cycle time. All other periods, called secondary cycles, are defined as integer multiples of the primary cycle. If for instance the value of T_{IPCY} = 10 ms and T_{IPH} = 5 (both parameters obtained by a pre-run-time analysis), the scheduler could produce a schedule as illustrated in Fig. 3, assuming that in each cycle the IPH queues have at least one pending fragment, so the actual dispatching corresponds to the schedule produced in each cycle.

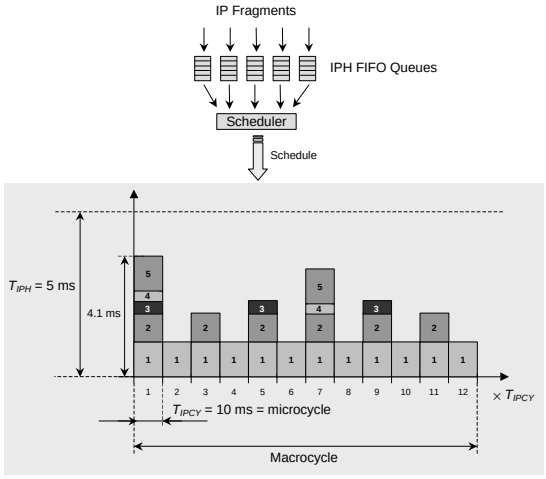

Fig. 3 Example of scheduling of the IP stream set of Table 1

Two important parameters are associated with this type of scheduling: the microcycle (elementary cycle) and the macrocycle. The microcycle imposes the maximum rate at which a fragment from a stream can be dispatched. Usually, the microcycle is set equal to the highest common factor (HCF) of the required stream periodicities. It is easy to depict, for the case of Table 2, that the sequence of microcycles repeats each 12 microcycles. This sequence of microcycles is said to be the macrocycle, and its length is given by the lowest common multiple (LCM) of the scan periodicities.

Table 2

Cycle	1	2	3	4	5	6	7	8	9	10	11	12
IPH1	1	1	1	1	1	1	1	1	1	1	1	1
IPH2	1		1		1		1		1		1	
IPH3	1				1				1			
IPH4	1						1					
IPH5	1						1					
Load (ms)	4.1	1.3	2.4	1.3	2.7	1.3	3.8	1.3	2.7	1.3	2.4	1.3

Other scheduling approaches were presented in [2] such as one based on a variation of the rate monotonic, called deferred release with the objective of reducing the required T_{IPH}.

As it was not possible to guarantee a periodic pattern of IP fragments to the queues (e.g., due to the bursty arrival pattern resulting from the timing behaviour of the applications, of the operation system and IPAS) online scheduling algorithms were also developed to compensate for "missed releases". These algorithms perform as illustrated in Table 3 (compensation of IPH3 in the "current" macrocycle) Table 4 (compensation of IPH1 in the follwing macrocycle). Note that the delayed fragments are considered to arrive just before microcycle 9.

Table 3

Cycle	1	2	3	4	5	6	7	8	9*	10	11	12
IPH1	0	1	1	1	1	1	1	1	1	1	1	1
IPH2		1	1		1			1	1		1	
IPH3				0		1				1		1
IPH4				1						1		
IPH5	1						1					
Load (ms)	1.3	2.4	2.4	1.4	2.4	1.6	2.6	2.4	2.4	1.7	2.4	1.6

Table 4

Cycle	1	2	3	4	5	6	7	8	9	10	11	12
IPH1	1	1	1	1	1	1	1	1	1	2	1	1
IPH2		1	1		1			1	1		1	
IPH3				1		1						1
IPH4				1						1		
IPH5	1						1					
Load (ms)	2,6	2,4	2.4	1.7	2.4	1.6	2.6	2.4	2.4	2.7	2.4	1.6

4. Limitations of the Approach

The solution described in Section 3, which is to be used in real prototypes within the framework of the RFieldbus European Project [6], uses a combination of static and dynamic scheduling. The static part is composed of a plan that is generated prior or during run-time and is executed by the scheduler on the TMS layer. The dynamic part is made available by the compensation algorithm that comes into action when it detects a reduction on the stream bit rate.

This solution can be easily implemented with a reduced run-time overhead and without using too many resources. However it presents an important set of limitations:

1. The approach is inefficient for scheduling variable bit rate (VBR) IP traffic;
2. The algorithms are not able to efficiently encompass neither runtime changing traffic characteristics nor runtime modification of the number of IPH streams;
3. The delay guarantees given by the algorithms are coupled with the bandwidth guarantees;
4. The algorithms over waste bandwidth.

These limitations will be individually analysed in the next subsections.

4.1 Scheduling VBR traffic

A VBR stream can be characterised, in a simplified manner, by the maximum and average data rate values. In the solution described in Section 3, VBR traffic is only possible to support if considering that the VBR streams always demand the worst-case bandwidth. This leads to a significant waste of bandwidth. Just to give an example, in a transmission of MPEG video it is possible to have a bit rate peak 10 times greater than the average bit rate.

Additionally, the online compensation mechanism may introduce undesirable effects on the resulting scheduling, since it tries to compensate non-arriving fragments.

4.2 Adaptability

The approach described implies that all the system parameters, namely those described in Section 2, are set, prior to run time, having the knowledge of all the streams' characteristics.

This may not always be possible. Especially when there is not a complete knowledge about the multimedia applications, or the applications change their traffic characteristics upon user commands or environment changes.

Additionally, the presented approach is not able to handle new streams coming into the system during runtime. The possibility of changing the set of streams being scheduled would make the system more flexible and able to work in a more dynamic environment.

4.3 Delay Guarantees

The presented approach is only able to guarantee a predefined bit rate for each stream. Thus the maximum delay for an IP packet, which is conveyed by IPH stream with parameter *p* and *s* (where *p* is the period - multiple of T_{IPCY} - and *s* is the maximum packet size) is equal to *p* x *n*, where *n* is the number of fragments for the IP packet.

So, consider the again that on the example of Table 2, if IPH5 maintains the same data rate but requires a maximum delay of 10ms and assuming a T_{IPCY} parameter of 5ms, then that scheduling does not guarantee this requirement. To realise that, IPH5 would have to be scheduled with a period of 5ms – this situation would lead to a waste of 66.6% of the bandwidth available to IPH5 and the minimum T_{IPH} parameter rise to 3.7ms.

Delay guarantees are especially important for interactive voice and video traffic, which usually requires a delay inferior to 300ms.

4.4 Bandwidth Use

The approach proposed is not able to fully use the available slot time (T_{IPH}) as it can be seen on the example of Table 2, which has an utilisation factor of 52.6% (considering T_{IPH} equal to 4.1ms). Nevertheless, the unused slot time is not wasted and it can be used for the scheduling of Best-effort traffic.

But, the current algorithms are not sufficiently flexible. For example, if a release is delayed by one microcycle in relation to its planned dispatching opportunity, it can only be scheduled by the compensation mechanism. The gap left over by missed releases can only be used by the scheduler to schedule other streams, if there are streams to compensate even when the other stream queues are not empty. Another problem arises when several available in the queues, but are not dispatched because of the static nature of the scheduler. If several best effort fragments arrive just before the slot time, where the guaranteed traffic is to be dispatched, then the best effort traffic must wait until the end of the transmission of the guaranteed traffic. Thus, a scheduler which is able to schedule each release whenever there is a gap available and at the same guaranteeing each stream data rate, would lead to a better use of the reserved IPH bandwidth.

5. Ongoing Work

A solution for these problems is now being thoroughly investigated. The outline is provided in Fig. 4, which reflects the following functionalities:

- Receive a packet from the IP layer and tag it according to the stream and scheduler parameters;
- Store each stream's fragments in a different queue;
- Generate a schedule, either periodically or before the arrival of the token to the master station;
- Execute, during the token holding time, the schedule previously obtained.

This approach is very similar to the Planning Scheduler proposed by Almeida in [7], where a scheduler generates a plan to be executed by the system dispatcher during the next cycle(s) (token holding time).

In this section we will briefly outline how the limitations described in Section 4 can be overcome.

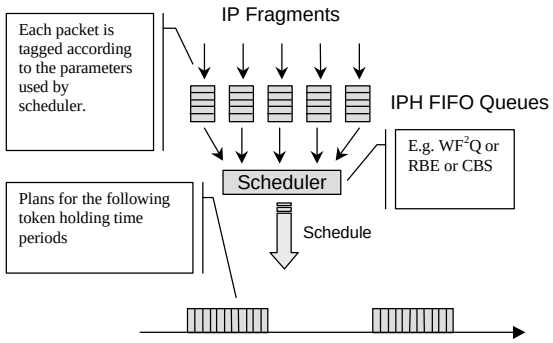

Fig. 4 – Draft of the General Solution

To solve these shortcomings, several rate-base scheduling strategies have been proposed in the literature, which can potentially be used for the system described in this paper.

In rate-based scheduling, each stream is guaranteed to be processed according to a pre-defined rate, or a stream is guaranteed a certain fraction of the available bandwidth.

Rate-base scheduling were classified by Jeffay [8] in three classes: *fluid-flow or proportional share allocation*, *server-based allocation* and *generalised Liu and Layland style allocation*.

Proportional share (PS) algorithms are able to guarantee specific QoS parameters for soft real-time applications, like bandwidth and delay. Examples of PS algorithms are weighted fair queueing (WFQ) and Worst-case weight fair queueing (WF^2Q).

In Server-based allocation mechanisms [9], a server is given a specific amount of bandwidth (server capacity), which it can use to schedule aperiodic tasks until its capacity is exhausted. The server capacity replenishment is usually done periodically. Recent developments like the Constant Utilisation Server (CUS) or the Constant Bandwidth Server (CBS) try to address the specific needs of multimedia applications.

Generalised Liu and Layland style allocation schemes address the problem of allowing more flexibility in the way a schedule responds to events that arrive at a uniform average rate but unconstrained instant rate. An example of such algorithms is the rate-based execution model proposed [8].

Many of the referred algorithms were not specifically developed for communication systems, and to our best knowledge none of them was developed specifically for a time division multiplexing system like the one described in Sections 2 and 3. Thus, its use in our system is only possible with some adaptations. These adaptations concern with the need to generate a plan for each token holding time, and to the need of fitting a set of streamed IP packets into one or more T_{IPH} time slots.

Several advantages can be envisaged by the use of these algorithms: improved flexibility on the schedule generation; compensation mechanisms are intrinsic to the algorithms; the streams' scheduling parameters are more easily changed and the transition between functioning modes can be smoother; the algorithms can be more responsive on the transmission of fragmented IP packets.

Nevertheless, its use also brings some drawbacks in relation to the solution described in Section 3. The schedule produced is not optimal and in some circumstances it may not be as good as the original solution. This results true for some algorithms, since it depends on the arrival time of the IP fragments. Additionally, dimensioning T_{IPH} parameter may turn out to be more difficult.

Adding adaptability to the algorithms would allow an efficient scheduling of traffic streams with unknown parameters or with variable parameters [10]. Such kind of mechanisms rely on the identification of the streams' characteristics and on the runtime modification of the stream scheduling parameters, e.g. by changing the weight of the stream (if scheduled under a PS algorithm).

These alternatives, together with admission control mechanisms are now being under thorough investigation in order to obtain performance improvements concerning the IP support in PROFIBUS networks.

6. References

[1] Pacheco, F., Tovar, E., Kalogeras, A., Pereira, N. "Supporting Internet Protocols in Master-Slave Fieldbus Networks", Proc. 4th IFAC Int. Conference on Fieldbus Systems and Their Applications (FET'2001), France, 2001, pp. 260-266.

[2] Ferreira, L., Tovar, E., Machado, S. "Scheduling IP Traffic in Multimedia Enabled PROFIBUS Networks", Proc. 8th IEEE Intl. Conf. on Emerging Technologies and Factory Automation (ETFA'2001), France, 2001, pp. 169-176.

[3] Tovar, E., Vasques, F., Pacheco, F., Ferreira, L. "Industrial Multimedia over Factory-Floor Networks", Proceedings of the 10th IFAC Symposium on Information Control Problems in Manufacturing (INCOM '01), Austria, 2001.

[4] Kumaran, S. and J.-D. Decotignie, "Multicycle Operations in a Field Bus: Application Layer Implications", Proc. of IEEE Annual Conference of Industrial Electronics Society (IEEE) 531-536, 1989.

[5] Raja, P. and Noubir, G., "Static and Dynamic Polling Mechanisms for Fieldbus Networks". In ACM Operating Systems Review, Vol. 27, No. 3, pp. 34-45, 1993.

[6] IST-1999-11316 RFieldbus Project, "High Performance Wireless Fieldbus in Industrial Multimedia-Related Environment, URL: http://www.rfieldbus.de.

[7] Luís Almeida, R. Pasadas, J. A. Fonseca - "Using The Planning Scheduler to Improve Flexibility in Real-Time Fieldbus Networks" IFAC, International Federation of Automatic Control, Control Engineering Practice Vol. 7, Nº 1, pp. 101-108, 1999, Elsevier Science, Ltd.

[8] K. Jeffay, S. Goddard, "A theory of rate-based execution", Proceedings of the 20th Real-Time Systems Symposium, Phoenix, USA, Dec. 1997, pp. 204-314.

[9] G. Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*, Kluwer Academic Publishers, 1997.

[10] L. Abeni, L. Papoli, G. Buttazzo, "On Adaptive Control Techniques in Real-Time Resource Allocation", Proc. of the 12th Euromicro Real-Time Systems Conference, June 2000, pp 129-136.