A Modified Token Passing Scheme for Enhanced Scheduling in Profibus Networks

L. Lo Bello, P. Bucolo, O. Mirabella
Istituto di Informatica e Telecomunicazioni
Università di Catania
llobello@iit.unict.it, pbucolo@mail.omnitel.it, omirabel@iit.unict.it

Abstract

The Profibus is a token-based communication protocol widely used in distributed process control systems to support time-critical communication between field devices (sensors, actuators, controllers). At the field level the transmission dynamics of the stations may differ quite significantly according to the dynamics of the processes supported, but the Profibus protocol, even if it distinguishes between high-priority and low-priority traffic, cannot take the transmission dynamics of the stations into account. Thus, if the stations featuring high-priority traffic have greatly differing transmission dynamics, the token rotation speed is set according to the station with the fastest dynamic, and all the stations receive the token with the maximum frequency. Thus, at each rotation the token will be passed to a station even if it does not have high-priority traffic to transmit, and this causes drawbacks. To avoid them, the token should be passed to a station when it really has data to transmit, and more frequently to stations with faster dynamics.

To this aim, this paper presents a strategy to modify the token passing policy, called the Rotating Ring, and compares its performance with that of the Profibus standard.

1. Introduction

In distributed process control systems, suitable bus networks called FieldBuses, using ad hoc communication protocols, have been developed [1][2] to interconnect field devices (e.g. sensors, actuators, controllers). One of the most widely used Fieldbuses is the Profibus [3] (included in the European Standard EN 50170 [4]), which uses a token-based Medium Access Control mechanism that is a simplified version of the Timed Token Protocol [5]. In the Profibus protocol, a *token* circulates between the stations of a *logical ring*, which is implemented on a physical bus. In the logical ring system *Master stations* manage the token and control communications with the *Slave stations*, which are devices connected to the bus that cannot transmit unless authorised by a master. Messages are transmitted in a *message cycle*, which comprises an *action frame* (*request* or *send/request frame*), a *reply frame* (*acknowledgement* or *response frame*) and possible *retries*.

Differently from the Timed Token Protocol [5], the Profibus does not provide for a synchronous bandwidth allocation specifying the length of time a station has at its disposal to transmit[1], but each station can use the token for a time interval, called the *Token Holding Time* (T_{TH}). The T_{TH} is determined by the station on each token visit as $T_{TH} = T_{TR} - T_{RR}$, where T_{TR} is the *Target Token Rotation Time* (a design parameter specifying the time that the token should theoretically take to make a complete round of the ring) and the T_{RR} is the *Real Token Rotation Time* (i.e., the time measured between two consecutive arrivals of the token in the station).

At the field level the transmission dynamics of the stations may differ quite significantly according to the dynamics of the processes supported. An example of a scenario in which this can occur is plants for welding and assembling body parts in an automobile industry. In this environment, a Profibus system can be used to connect various PLCs (masters), each of which controls welding or assembly robots that move at different speeds, and so require different control dynamics. In this case the *periodic* traffic on the Profibus network usually consists of a number of variables which can have very different time constraints.

The Profibus distinguishes between high-priority and low-priority traffic, but it is not, however, able to provide

[1] Due to the lack of synchronous bandwidth allocation, the traditional Timed Token real-time analysis, widely used to study the real-time behaviour of networks based on the Timed Token protocol [8][9], cannot be applied to the Profibus, and other approaches, as that proposed in [6], are needed.

different management, within the high-priority traffic class, of the transmission requirements of stations with fast dynamics, and thus more tight deadlines, and those of stations with slower dynamics. That is, if the stations on the bus have greatly differing dynamics, the token rotation speed is that of the variable with the fastest dynamic. However, a number of stations needing to transmit with very different time constraints cannot co-exist well on a Profibus system.

Let us assume, for example, that we have both fast stations and slow stations on the same logical ring. At each round the token will be sent to all the stations, even if the slower ones do not yet have high priority traffic to transmit. These stations can immediately pass the token without transmitting, thus involving a bandwidth waste, or transmit low priority traffic as well. Some of them may use all the T_{TH} at their disposal to process low-priority traffic and end up by delaying rotation of the token. This delay, caused by transmission activities stretching beyond the T_{TH}, is due to the fact that once a transmission has started it is not interrupted even if the T_{TH} runs out. This situation is called a *T_{TH} overrun*. As a result, the next master will only be able to transmit one high-priority message, possibly giving rise to a further T_{TH} overrun (this time due to high-priority traffic transmission), and then it will have to pass the token immediately. In these conditions, the probability of a T_{TH} overrun occurrence by the fastest masters increases.

This delay in rotation of the token can affect the system's behaviour towards the faster traffic (both high - and low priority). Consequently, some high-priority messages from the faster stations may well remain in the queue, because the portion of bandwidth these stations have at their disposal is not enough to serve all the incoming high-priority traffic. The consequence of this is that when the workload increases, some fast-dynamic variables may not be transmitted by their deadline. The larger the number of stations on the ring, the more critical this situation will become. This is very undesirable, as in distributed process control systems fast-dynamic exchanges, especially periodic ones, are typically the most critical for the correct behaviour of the system and therefore the most important.

Apparently the most straightforward solution to the problem would be to avoid the co-existence of processes with very different dynamics on the same Profibus. However, there are several practical reasons why this co-existence is necessary or convenient. Besides the saving on cabling, which facilitates installation and maintenance and reduces costs, there is the optimisation of variables, as some devices and their variables may be shared by several control loops. Another aspect is supervision of the functioning of a plant. The use of a single bus allows the supervision system to collect information from a number of control sub-systems.

As a result, in order to maintain a single bus connecting both slow and fast stations while avoiding the previously described drawbacks, token passing should be handled in such a way that the token is passed more frequently to faster stations. It should therefore be passed more frequently to stations with faster dynamics.

To this aim, this paper proposes a strategy, called the Rotating Ring, to modify the policy used for token passing so as to meet the different time constraints of the control traffic and at the same time to make bandwidth exploitation as efficient as possible.

2. The Rotating Ring

The approach proposed in this paper is implementing two *virtually separate* logical rings on the same physical channel - a fast one (which will henceforward be called Ring A, comprising the stations with faster dynamics) and a slow one (called Ring B, comprising all the other stations) - using the same token.

As shown in Fig.1, token passing will be handled in such a way that the token will be made to rotate prevalently on Ring A so as to meet the time constraints of the stations on this ring; at a certain point it will be passed to Ring B, which will give it back after it has used it.

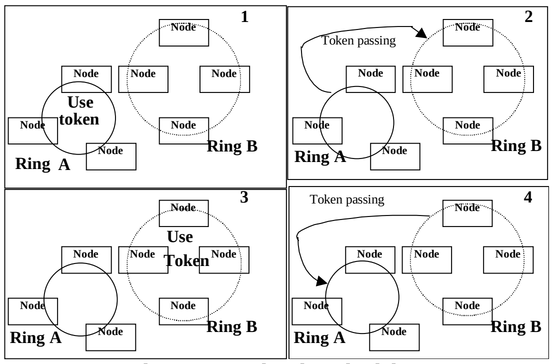

Figure 1. The basic idea

In particular, in the Rotating Ring (RR) the token is passed to Ring B whenever a complete round has been made of Ring A, even if $T_{TH\ A} < 0$ (this is in accordance with the philosophy of the Profibus, whereby one high-priority message cycle is transmitted even if $T_{TH} < 0$). This means that, in practice, it is as if an extra station had been added to Ring A, but not always the same one, as it changes each time the token completes a round of ring A.

It is the n-th master of Ring A (i.e. the last one, as we assume that n is the number of masters on Ring A), indicated as M_{nA}, which passes the token to the NS of the logical ring B, which we indicate as M_{kB}, with k=1..m, where m is the number of masters on Ring B.

Whereas the token, after having been used by a Ring B station, is always returned to the same ring A station (the first one of the logical ring), every time it is passed to ring B, it is sent to a different station, according to the logical ring B, so that all the ring B stations receive it cyclically. For the mechanism to work properly, it is sufficient for the Ring A stations to monitor the Ring B Live List (as the Live List contains all the functioning stations of the logical ring).

On each complete round of the fast ring, the token is passed to a single ring B station. In this way it is possible to estimate the worst case latency of the token on ring B.

In Sect.3 the performance of the Rotating Ring will be compared with that of the Profibus standard by discussing the results obtained via simulation. As centralized systems have the drawback of having a single point of failure, in the conclusions we will discuss how to deal with fault-tolerance in the centralized approach proposed.

3. Performance Evaluation

To compare the performance of the Rotating Ring with that of the Profibus standard (which will henceforward be called Single Ring – SR) a suitable simulator was developed. We map periodic traffic (real-time) onto Profibus high-priority traffic and aperiodic traffic (non time-critical) onto the low-priority one (we assume that devices which can generate alarms are periodically checked through polling cycles, so that they are managed as high-priority traffic). Performance was evaluated with 20 master stations connected to the bus in a configuration with 10 fast stations and 10 slow stations.

The following operating conditions were assumed:

- A network with a bit rate of 1Mbit/s was used.
- The average frame size was fixed as 250 bits.
- The workload was chosen in such a way that the slow and fast stations share 50% of the bandwidth.
- Each station generates both high-priority and low-priority traffic, so as to split its available bandwidth into two equal parts.

There are therefore four kinds of traffic, classified as Fast High Priority, Fast Low Priority, Slow High Priority and Slow Low Priority Traffic. We measured the percentage of high-priority messages transmitted within the deadlines and the percentage of low-priority messages transmitted within some reference delay thresholds (to be seen as indexes of quality of service rather than as real constraints). The values chosen for the deadlines and the thresholds are given in Table 1.

Table 1. Deadlines and Thresholds used

Fast Ring High Priority	Fast Ring Low Priority	Slow Ring High Priority	Slow Ring Low Priority
5 ms	10 ms	22 ms	32 ms

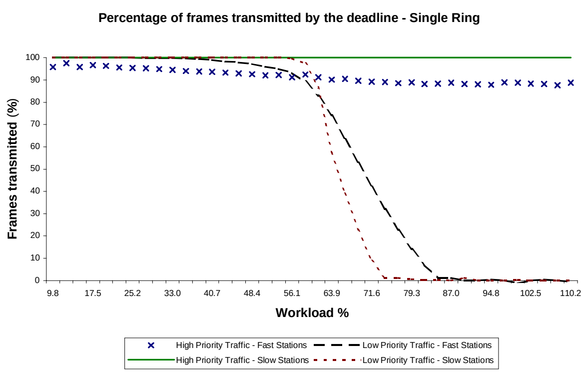

Figure 2(a). Percentage of frames transmitted by the deadline in the SR

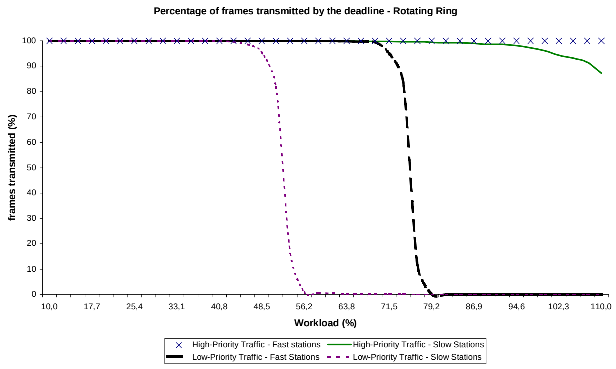

Figure 2(b). Percentage of frames transmitted by the deadline in the RR

Figs. 2(a) and 2(b) show the percentage of frames transmitted by the deadline. Unlike the SR, the RR can meet the deadlines for the fast stations' high-priority traffic until the bandwidth is saturated. Figs. 3(a) and 3(b) show that the throughput for high-priority traffic is the same in both the systems, but in the RR the fast stations' low priority traffic outperforms that of the slow stations.

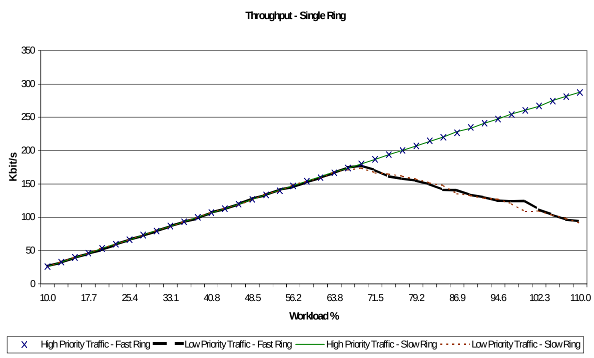

Figure 3(a). Throughput for the SR

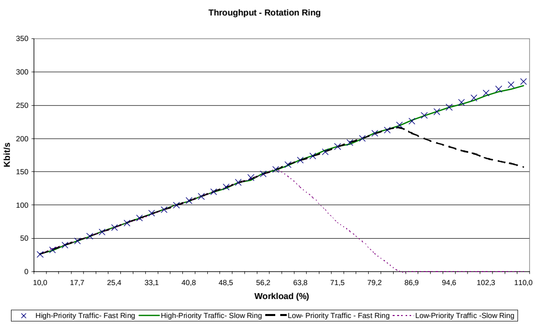

Figure 3(b). Throughput for the Rotating Ring

To highlight the difference in behaviour as far as delay is concerned, statistical analysis was made of the delay values for the single messages. Fig. 4 shows the access delay distribution obtained for fast high-priority traffic with the RR and the SR in workload conditions equivalent to 50% of the available bandwidth.

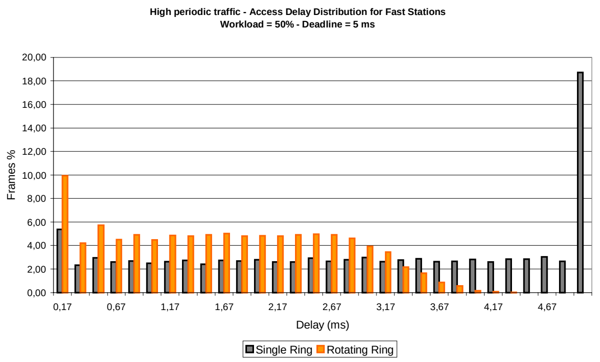

Figure 4. Access delay distribution for fast high-priority traffic

The values obtained with the RR are more centred on lower delay values than those obtained with the SR, which are quite evenly distributed and show a peak around 5 ms, showing that the delay values for the single messages may be very high.

4. Conclusions

The Rotating Ring offers stations with a faster dynamic a better "quality of service", as all their traffic (both high- and low-priority) is served better. This feature is a positive one, as in many distributed process control systems fast-dynamic exchanges, especially periodic ones, are typically the most critical for the correct behaviour of the system and therefore the most important.

As it is known, the Profibus is often used in harsh environments, and so data corruption and possible message resends should be taken into account. In the RR, message loss and retries can be handled as in the standard Profibus. To make the system behave like the standard Profibus when the RR is used, it is necessary that all the masters (on both the fast and slow rings) possess and keep updated a *global* Live List (this is possible as the various masters are all connected to the same bus).

As far as fault tolerance is concerned, in the RR the most critical case happens when a fault occurs in the n-th master on the fast ring (i.e. M_{nA}). If the fault hits the transmitter when M_{nA} has the token and has to pass it to the slow ring, when the fast ring timeout expires without any valid frames being heard, the same procedures of the standard Profibus are activated. If the fault affecting M_{nA} is a permanent one and so it is excluded from the logical ring, another master (the PS of M_{nA}) will pass the token to the slow ring (using the global Live List to make the token circulate correctly). If, on the other hand, the fault is in the receiver, and so the token coming from the slow ring is not received by M_{nA}, the effect is the same as the one occurring in the standard Profibus upon loss of the token, and so the conclusions of the analysis in [10] are still valid.

5. References

[1]J.D. Decotignie, P. Pleinevaux, "Time Critical Communication Networks: Field Buses", *IEEE Network*, vol.2, No.3, May 1988.

[2]J. Decotignie, P. Pleinevaux, "A Survey on Industrial Communication Networks", *Annales des Telecommunications*, 48, n.9-10, pp.435-448, 1993.

[3]Profibus: Process Field Bus, German Standard DIN V 19245 *Teil1:Ubertragunstechnik, Buszugriffs, und Ubertagunsprotokoll, Dienstschnittstelle zur Anwendung-Schnitt, Management*, Jan.1988.

[4]EN 50170 "General Purpose Field Communication System", European Standard, CENELEC, Vol.2/3 Profibus, July 1996.

[5]R.M. Grow, "A Timed Token Protocol for Local Area Networks", in Proceedings of Electro '82, Token Access Protocols, Paper 17/3, May 1982

[6]E.Tovar, F. Vasques, "Real-Time Fieldbus Communications Using Profibus Networks", *IEEE Transactions on Industrial Electronics*, Vol.46, NO.6, Dec 1999.

[7]*Fiber Distributed Data Interfaces (FDDI)-Token Ring Media Access Control (MAC), ANSI X3.139*, American National Standard for Information Systems, 1987.

[8] G. Agrawal, B. Chen, W. Zhao, S. Davari, "Guaranteeing Synchronous Message Deadline with the Timed Token Medium Access Control Protocol, *IEEE Trans. on Computers*, Vol.43, N.3, pp.327-339, March 1994.

[9] Q. Zheng, K.G. Shin, "Synchrounous Bandwidth Allocation in FDDI Networks", *IEEE Trans. on Parallel and Distributed Systems*, Vol.6, N.12, pp.1332-1338, Dec.1995.

[10]L. Lo Bello, O. Mirabella, "A Fault-tolerance analysis of Profibus Systems by means of Generalised Stochastic Petri Nets", IECON'99, 25th IEEE Industrial Electronics Society, Nov. 1999, San Jose, California, USA.