Katholieke Hogeschool St.-Lieven
Departement INDUSTRIEEL INGENIEUR
Gebroeders Desmetstraat 1, 9000 Gent

Electronics Division
Information and Communication Technology

Timing performance of a hybrid wired/wireless PROFIBUS-based network

A dissertation submitted in partial fulfilment of the requirements
for the degree of Industrial Engineer

by

Katrijn Van Nieuwenhuyse
Steve Behaeghel

Supervisor: Dr. Ir. Mário Alves

Polytechnic Institute of Porto (ISEP/IPP)-Portugal
IPP-HURRAY! Research Group

June 2003

TIMING PERFORMANCE OF A HYBRID WIRED/WIRELESS PROFIBUS- BASED NETWORK.

Abstract

The integration of wireless communication and multimedia into PROFIBUS is reflected in the IST (Information Society Technology) RFieldbus project (High Performance Wireless Fieldbus in Industrial Multimedia-Related Environment). Our project addresses the analysis and measurement of the timing behaviour of such a hybrid network, in a manufacturing automation test bed. This field trial was implemented to validate the RFieldbus approach for integrated wireless and wired communication in the factory floor, and to support multimedia streams and mobile nodes. A System Planning Application (SPA) is used to compute all the necessary timing parameters (both PROFIBUS and RFieldbus-specific) for the real-time operation of the network, depending on the network topology, on the characteristics of the message streams and on the types of physical media (wired/wireless). The worst-case values obtained (duration of message transactions,...) are compared with the real traffic performance, by means of a PROFIBUS Network Analyser. This analysis was then used to draw some conclusions about the pessimism introduced by the SPA (worst-case values) and about the real timing response for different networks(topology, nodes, streams, ...). In RFieldbus, the intermediate systems used to relay the network traffic between wired and wireless PROFIBUS operate at the Physical Layer level. In spite of their repeating operation, there are several important parameters that must be monitored. For this purpose, another objective of the project was to extend an already existing local monitoring application to remote multiple IS monitoring, via TCP/IP.

Keywords: PROFIBUS - RFieldbus – worst-case analysis timing performance - traffic measurement - IS monitoring

TIJDSPERFORMANTIE VAN EEN HYBRIDE BEDRAAD/DRAADLOOS NETWERK

Abstract

De integratie van draadloze communicatie en multimedia in PROFIBUS is weergegeven in het IST (Information Society Technology) RFieldbus project (High Performance Wireless Fieldbus in Industrial Multimedia-Related Environment). Ons project richt zich tot de analyse en meting van een dergelijk hybride netwerk in een geautomatiseerd fabriekstestopstelling. Deze field trial was geïmplementeerd om de RFieldbus benadering te valideren voor geïntegreerde bedraade en draadloze communicatie op de fabrieksvloer en om multimedia streams en mobiele knooppunten te ondersteunen. Een System Planning Application (SPA) wordt gebruikt om al de noodzakelijke timing parameters te berekenen (zowel PROFIBUS als Rfieldbus-specificiek) voor de real-time werking van het netwerk afhankelijk van de netwerktopologie, de karakteristieken van de message streams en de soorten fysieke media (bedraad/draadloos). De bekomen worst-case waarden (duur van transacties van berichten,...) worden vergeleken met de werkelijke performantie van de trafiek, met behulp van een PROFIBUS netwerk analysator. Deze analyse kan worden gebruikt om enkele conclusies te trekken over het pessimisme geïntroduceerd door de SPA (worst-case waarden) en over de werkelijke tijdsresponsie voor verschillende netwerken (topologie, nodes, streams, ...). In RFieldbus werken de intermedia systemen, gebruikt om de netwerktrafiek tussen bedrade en draadloze PROFIBUS door te geven, in de Fysische Laag. Ondanks hun werking als repeaters, zijn er meerdere belangrijke parameters die gemonitord moeten worden. Daarvoor is een ander doel van het project het uitbreiden van een reeds bestaande monitoring applicatie tot het remote en multiple IS monitoren via TCP/IP.

Keywords: PROFIBUS - RFieldbus – worst-case tijdsanalyse – draadloos – trafiek meting

Acknowledgements

First, we would like to thank all the people in the IPP-HURRAY! research group, especially our supervisor Mário Alves, for the time he invested in us and in our thesis.
We also want to express our gratitude for, on one side, the Polytechnic Institute of Porto (IPP) and its School of Engineering (ISEP) in Porto (Portugal) and on the other side, the department of Industrial Engineering of Kaho St. Lieven in Gent (Belgium), who made this amazing educational Erasmus experience possible.
The effort made by our teachers in Belgium during our four years of study, provided us with lots of knowledge, and that is invaluable.
Thanks to all our friends and family, who always supported us during this unforgettable period of our lifetime.
Finally, we are most grateful to our parents, who believed us and made a great effort to invest in such an adventure.

Porto, 13th of June, 2003

Katrijn Van Nieuwenhuyse
Steve Behaeghel

Table of Contents

1 Introduction ... 2
1.1 Context ... 2
1.2 Structure of this thesis ... 2
2 RFieldbus architecture ... 3
2.1 Federating Communication system ... 3
2.2 PROFIBUS basics ... 3
2.3 The RFieldbus architecture ... 6
3 Setting up the RFieldbus Network ... 10
3.1 Important timing parameters in the network ... 10
3.2 Guaranteeing real-time communications in RFieldbus networks ... 13
3.3 The System Planning Application (SPA) ... 17
4 RFieldbus Manufacturing Automation Field Trial ... 18
4.1 System layout and functionality ... 18
4.2 The network ... 19
4.3 Message streams ... 20
5 Matching analytical vs. experimental results ... 26
5.1 Analytical results (worst-case scenarios) ... 26
5.2 The PROFIBUS network analyser ... 29
5.3 Tackling with additional implementation delays ... 30
5.4 Analytical vs. experimental results ... 37
6 Monitoring of the IS devices ... 45
6.1 Introduction ... 45
6.2 The Monitoring Application ... 46
6.3 Implementation aspects ... 48
7 Conclusion ... 51

ANNEX
A – Parameters monitored by the Local IS Monitoring application
B – Open Day of the MAF of the RFieldbus project
C – Paper published at RTLIA ‘03

1 Introduction

1.1 Context

Nowadays, the communication infrastructure of Distributed Computer-Controlled Systems (DCCS) is usually based on Fieldbus networks. They are appropriate for this purpose since they provide adequate levels of performance, dependability, timeliness, maintainability and cost. Nevertheless, there is the need for extending the capabilities of Fieldbusses to cover functionalities not previously considered in such type of networks: industrial wireless and mobile devices (e.g. handheld computers, transportation equipment) and ability to support industrial multimedia traffic (e.g. video, audio, file transfer, WWW). It is obvious that, for this kind of hybrid wired/wireless architecture, cabling starts to be an obstacle. Nevertheless, the wireless communications must cope with real-time and dependability features at least similar to the ones available in traditional (wired) Fieldbus networks.

PROFIBUS, a popular standard Fieldbus protocol, gathers a set of properties that are relevant for the targeted architecture. The integration of wireless communication and multimedia into PROFIBUS is reflected in the IST (Information Society Technology) RFieldbus project [8] with participation of the IPP HURRAY! Research group [9], where we realise our project.

Our thesis mainly addresses the analysis and measurement of the real-timing behaviour of such a hybrid network, in a manufacturing automation test bed [6]. This field trial was especially implemented to validate the RFieldbus approach.

1.2 Structure of this thesis

Chapter 2 describes a model for a hybrid wired/wireless communication architecture based on PROFIBUS, without changing this protocol.

Chapter 3 overviews the most important parameters that must be set in the network to realize the targeted architecture. It also introduces the System Planning Application tool (SPA), which computes these parameters.

In Chapter 4, the Manufacturing Automation Field Trial (MAF) used to do the experimental part of this thesis, is described.

The worst-case results output by the SPA are compared against measurements carried out using a network analyser, in Chapter 5. This chapter also presents the conclusions about the pessimism introduced by the SPA.

Another objective of the project was to extend an already existing local monitoring application to remote and multiple IS monitoring via TCP/IP. This is described in chapter 6.

Annex A presents all parameters that can be monitored in the Intermediate Systems. Finally Annex B and C reflect some results of our work that go beyond the scope of this thesis. Namely, our participation in the "Open Day of the RFieldbus project", where the most important technological add-ons of this project were presented to some representative Portuguese academic and industrials institutions (Annex B). Additionally, the joint work developed in the scope of this thesis also resulted in the publication of a paper in the 2nd Int. Workshop on Real-Time LANs in the Internet Age – RTLIA'03 [10], held by the Polytechnic Institute of Porto, in conjunction with the 15th Euromicro Conference on Real-Time Systems (Annex C).

2 RFieldbus architecture

2.1 Federating Communication system

PROFIBUS has been adapted for the extension of a traditional (wired) Fieldbus network to support wireless end mobile nodes. This is the world's leading Fieldbus standard for manufacturing automation and process control (over 20% market share). Since it is standardized under the Fieldbus Standards EN 50170 and IEC 61158, stability and openness for users and vendors are guaranteed. More specifically, the PROFIBUS-DP communication profile was adopted. This protocol has a number of advantages, since it gathers a set of features that are relevant for the targeted architecture. These features are briefly described in Table 2.1 [2].

Available features	Relevance for targeted architecture
Fastest *transmission speed* in a Fieldbus system (12 Mbit/s).	Support multimedia bandwidth-consuming applications
Supports *high-priority and low-priority messages* + PROFIBUS *MAC protocol*	Well-defined timing behavior for the transferred messages => support real-time traffic, with bounded response times [11]
Ring maintenance mechanisms	Appropriate to handle wireless/mobile stations
Error detection and *correction mechanisms* of the PROFIBUS DLL	Assures an acceptable level of reliability to the application layer

Table 2.1: Adequateness of PROFIBUS-DP to support the RFieldbus architecture

The extension from Fieldbus to RFieldbus should not influence the performance, high dependability and real-time behaviour of the Fieldbus system. Therefore different radio technologies had to be assessed for the extension of the wired PROFIBUS-DP protocol. The only problem was that there were no such technologies designed for industrial applications. The search for integrating one of them that fulfil the requirements and associated technology of RS-485 (in wired domain) leaded to a physical layer based on IEEE 802.11b for the radio-wireless domain.

The remainder of this chapter presents the most relevant characteristics of PROFIBUS (2.2) and of the RFieldbus architecture (2.3).

2.2 PROFIBUS basics

Based on [4]

2.2.1 Application-specific versions

PROFIBUS (PROcess FIeld BUS) is a successful, open, industrial Fieldbus standard can be used in a broad application spectrum. PROFIBUS enables the exchange of data between devices from different manufacturers, without special interface adjustments.
PROFIBUS comprises the following three application-specific versions (Table 2.2):

PROFIBUS-FMS (FMS = Fieldbus Message Specification)
This is the general-purpose solution for communications tasks at the field and cell levels of the industrial communications hierarchy. This version of PROFIBUS is getting obsolete and in the future the trend is only to use PROFIBUS-DP and PROFIBUS-PA, depending on the environment.

PROFIBUS-DP (DP = Decentralized Peripherals**)**
Optimised for high speed, this PROFIBUS version has been especially tailored for communication between automation systems and local peripherals, enabling plug-and-play for field devices.

PROFIBUS-PA (PA = Process Automation)
PROFIBUS-PA is the PROFIBUS version for process automation applications. PROFIBUS-PA uses the intrinsically safe transmission technology defined in IEC 1158-2 and enables the remote supply of stations through the bus.

DEVICE APPLICATION PROFILES	EN 50170		
	General Purpose Automation	Factory Automation	Process Automation
	PROFIBUS – FMS	PROFIBUS – DP	PROFIBUS - PA
	Universal	Fast	Application Oriented
	-Getting obsolete -Being replaced by PROFIBUS-DP/PA	-plug and play -efficient and cost effective	-powering over the bus -intrinsic safety

Table 2.2: Application Specific Versions of PROFIBUS

2.2.2 PROFIBUS Architecture

The protocol architecture is oriented into the OSI (Open System Interconnection) reference model in accordance with the international standard ISO 7498. In this model, every transmission layer handles precisely defined tasks. In the PROFIBUS architecture, only 3 layers are used: layer1, 2 and 7. Layers 3 to 7 are not defined. Layer 1 (physical layer) defines the physical transmission characteristics. Layer 2 (data link layer) defines the bus access protocol. Layer 7 (application layer) defines the application functions. The architecture of the PROFIBUS protocol is shown in Figure 2.1.

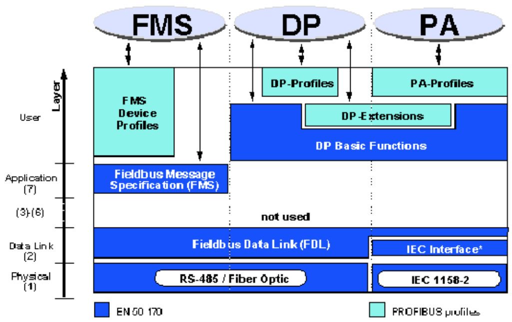


Figure 2.1: PROFIBUS protocol architecture

In the context of the RFieldbus project, namely in the Manufacturing Automation Field Trial (see Chapter 4), PROFIBUS-DP is used. PROFIBUS-DP uses layer 1 and 2, and the user interface. This streamlined architecture ensures fast and efficient data transmission. The Direct Data Link Mapper (DDLM) provides the user interface easy access to layer 2. The application functions which are available to the user as well as the system and device behaviour of the various PROFIBUS-DP device types are specified in the user interface. RS-485 or fibre optics transmission technology are available.

2.2.3 Main characteristics of the PROFIBUS Data Link Layer

Data transfer services

The PROFIBUS FDL offers three a-cyclic and one cyclic data transfer services:

- The Send Data with Acknowledge (SDA) service allows an user to send data to a single remote station. If an error occurs, the data transfer is repeated.

- The Send Data with No acknowledge (SDN) service allows an user to transfer data to a single remote station, to many remote stations (Multicast), or to all remote stations (Broadcast) at the same time, without any confirmation.
- The send and Request Data (SRD) service allows an user to transfer data to a single remote station and at the same time to request data from the remote station. If an error occurs, the data transfer is repeated.
- The Cyclic Send and Request Data (CSRD) service allows an user to poll remote stations (using SRD data transfers). The list of the devices to be polled is called the Poll List.

Message cycle

An important PROFIBUS concept is the Message Cycle, which comprises the request PDU sent by the initiator (always a master) and the associated acknowledgement or response PDU from the responder (usually a slave, but can also be a master). The acknowledgement or response must arrive before the expiration of the Slot Time, otherwise the initiator repeats the request. However, before issuing a new request, the initiator must wait a time interval defined by the Idle Time parameter. This creates the inter-frame synchronizing period of idle bits each Action Frame should be preceded by.

Token passing mechanism

All three PROFIBUS versions (DP, FMS and PA) use a uniform DLL and therefore a common MAC (Medium Access Protocol). In PROFIBUS, layer 2 is called Fieldbus Data Link (FDL). The MAC specifies the procedure for a station to have permission to transmit data. The MAC must ensure that only one station has the right to transmit data at a given moment. This mechanism is needed because PROFIBUS is a multi-master network. Therefore, PROFIBUS defines 'active' and 'passive' participants. Passive participants are always slaves and active participants are always masters who form a "token-ring". The token is continuously being passed between all active participants (masters). This token-bus network structure is depicted in Figure 2.2.

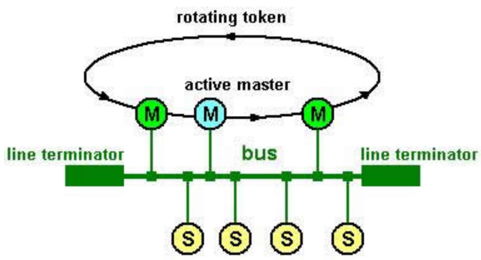


Figure 2.2: Token-bus

An important advantage of the use of a token-bus, is the possibility to guarantee real-time communications prior to run-time [11].

PDU formats
Based on [3]

In the asynchronous (RS-485) version of the PROFIBUS PhL (v1), each PDU is coded in UART characters (Figure 2.3). Each UART character comprises eleven bits: one start bit (binary 0), eight data bits (octet), one (even) parity bit and one stop bit (binary 1).

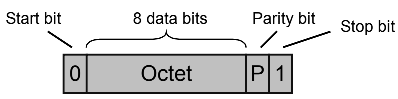


Figure 2.3: UART character

Each Action PDU, the first PDU transmitted in all transactions, must be preceded by a synchronization period of at least 33 idle bit periods (T_{SYN}). Every PDU starts with a start delimiter (SD) that characterizes its type (Table 2.3), defined by the PROFIBUS standard, except for the short acknowledgement PDU. This one comprises a single character (SC = E5) and is used for positive acknowledgements of SDA requests and negative acknowledgements of SRD requests.

Start Delimiter	Hexadecimal Value	PDU Type
SD1	10	Fixed length PDU with no data field
SD2	68	PDU with variable data field length
SD3	A2	PDU with fixed data field length
SD4	DC	Token PDU

Table 2.3: Start delimiters

A fixed-length PDU (request or acknowledgement) with no data field has a starting delimiter SD1, followed by the Information Field, which comprises the destination address (*DA*), the source address (*SA*) and the PDU control (*FC*). The two last fields of the PDU are the PDU check sequence (*FCS*) and the end delimiter (*ED*), which is always the hexadecimal value 16.
PDUs with fixed-length data field have starting delimiter SD3, followed by the Information Field, which also comprises a data field (*DATA*) with a fixed length of eight octets. PDUs with variable data field length have start delimiter SD2 and include the length field, which is duplicated for reliability reasons (*LE*=*LEr*, between 4 and 249). The Information Field comprises the *DATA* field, with a length varying from 1 octet minimum to 246 octets maximum. The token PDU is composed of the start delimiter (SD4) and the source and destination address fields.

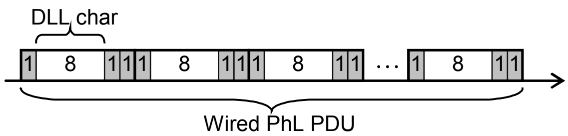


Figure 2.4: Wired PhL PDU

When relaying a PhL PDU (Figure 2.4) from a Wired Domain to a Wireless Domain, the Intermediate System removes every additional 3 bits and encapsulates the entire data octets in the data part of the wireless PhL PDU. The wireless PhL PDU also includes a preamble, start PDU delimiter and header, that are usual in wireless communication physical layers, such as in IEEE 802.11b. The wireless PhL PDU (Figure 2.5) has an overhead of 200 bits, corresponding to preamble, start PDU delimiter and header information.

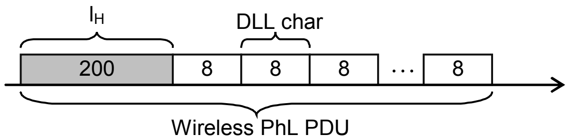


Figure 2.5 Wireless PhL PDU

2.3 The RFieldbus architecture.

Based on [1],[2] and [3]

2.3.1 General aspects of the system architecture

For many years the traditional PROFIBUS protocol provided the required functions on the factory-floor. Moreover, it fulfils the real time requirements of these systems. More recently, there had been a trend to extend the PROFIBUS protocol to encompass wireless and multimedia capabilities. Importantly, the support of these emerging functionalities should still guarantee compatibility with legacy PROFIBUS nodes.

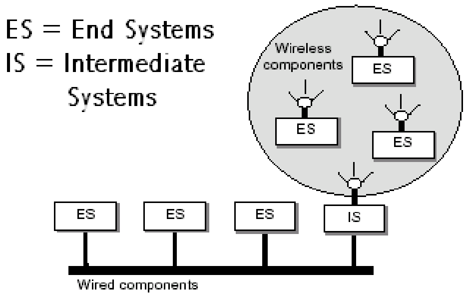


Figure 2.6: Basic components of a hybrid wired/wireless system

First of all, to integrate wireless and mobile nodes in the system, it is necessary to interconnect the traditional wired bus structure with several wireless End Systems (ES). This is carried out using interconnecting devices (Intermediate Systems – ISs) (Figure 2.6). Secondly, due to recent technological developments, the support of industrial multimedia applications by PROFIBUS is also required on the factory floor (e.g. video, audio, www).
This integration of wireless/mobile nodes and multimedia is, for example, realized in the IST (Information Society Technology) project RFieldbus [8]. The purpose of R(adio)Fieldbus was to improve the existing Fieldbus system (with PROFIBUS) by adding radio-based wireless technologies and multimedia capabilities.

To build a hybrid wired/wireless PROFIBUS-based network, there is the need of different objects:

- Physical Media
- Communication Domains
- Intermediate Systems (ISs)
- End Systems (ESs)

The major components for a hybrid wired/wireless PROFIBUS-based network (also referred as 'Communication Network'), are Wired and Wireless End Systems (Figure 2.6). These are End Systems having a wired or wireless interface. Wired network master (M) and slave (S) nodes (Figure 2.7) communicate with wireless/mobile nodes through Intermediate Systems acting as repeaters (at the Physical Layer level): Link Stations (LS), Base Stations (BS) and Link Base Stations (LBS). This leads to a broadcast network where all stations receive every transmitted PDU.

- o A RFieldbus Link Station (LS) enables wireless nodes to be connected to a wired PROFIBUS segment.
- o A RFieldbus Base Station (BS) offers a maximum radio coverage and improves reliability.
- o A RFieldbus Link Base Station (LBS) combines the characteristics of a LS and a BS, as a prerequisite for inter-cell mobility.

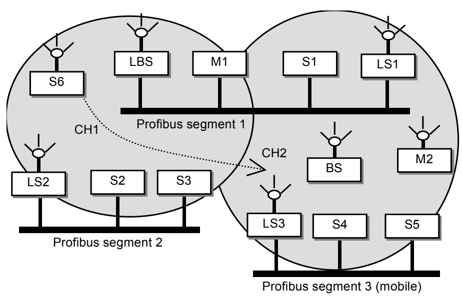


Figure 2.7: A hybrid wired/wireless system supporting inter-cell mobility

A Wired Communication Domain or Wired Domain (WRD) can be defined as a set of End Systems (M and S) and Intermediate Systems (LBS and LS) communicating directly via a wired physical

medium. Wireless ESs have a wireless network interface enabling the communication while moving within a pre-defined three dimensional area. Depending on the dimension and layout of this radio coverage region, there may be the need of splitting the radio coverage area into a number of smaller regions (radio cells). A Radio Cell or Radio Channel (CH) is therefore defined as a common radio coverage area of a group of WLESs and ISs. The set of WLESs/ISs that defines a Radio Cell is called Wireless Communication Domain or Wireless Domain (WLD). Notice that there are no pure WLESs implemented in the RFieldbus Manufacturing Automation Field Trial (see Chapter 4). The Wireless and Mobile nodes are always attached to a PROFIBUS segment.
Radio Cells have to be structured when inter-cell mobility must be supported, i.e. when PROFIBUS nodes/segments must be able to communicate while moving from one Radio Cell to another (CH1 to CH2 or vice-versa in Figure 2.7). The mechanism that supports inter-cell mobility is called handoff. Only Structuring Intermediate Systems (BSs/LBSs) can realise a Structured Radio Cell. A Structured Wireless Domain (SWLD) is defined as the set of End Systems and Intermediate Systems that are associated to a Structured Radio Cell. All communications between ESs belonging to a Structured Wireless Domain must be relayed by the Structuring Intermediate System. LBS and BS operate in different radio channels (CH1 and CH2, respectively) in order to have a structured wireless network, supporting inter-cell mobility.

2.3.2 Inter-cell mobility

As defined in 2.3.1, there is the possibility for Wired Domains to move from one Radio Cell to another. For this reason there is the need for a mechanism where these WLDs can change from Radio Cells without losing connectivity with the rest of the network. The Mobility Management mechanism provides a seamless handoff for all kinds of mobile ESs (master/slave) and mobile ISs (associated to the MWRD) [12]. Due to the broadcast nature of the Communication Network, the proposed mobility management mechanism just encompasses a procedure for Radio Channel assessment and switching. Importantly, the proposed mobility management mechanism guarantees that there is no PDU loss (considering no faults) and permits to fulfil stringent real-time requirements.
The Mobility Master (MobM) is a (dedicated) master that is responsible for triggering the mobility management procedure.
As soon as this master receives the token, the mobility management procedure is carried out. This procedure includes the following steps:

- The MobM broadcasts a *beacon trigger* to the entire network. This is a special PDU, more specifically a PROFIBUS SDN (unacknowledged request) PDU with a PROFIBUS header and a one byte data field.
- All the LBSs receive this beacon trigger and send a predefined amount of beacons in their own radio channel during the *beacon period* (Figure 2.8 - the mobility management procedure for S6, from Figure 2.7, moving from CH1 to CH2). This period is well defined so that even the most remote mobile stations receive the beacons within this period.

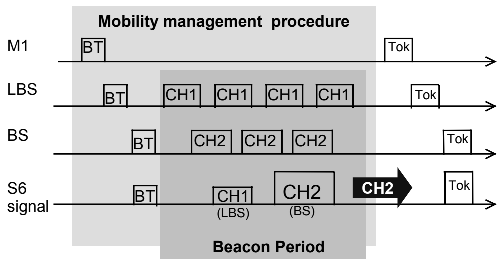


Figure 2.8 Mobility management procedure

- Mobile ESs/LBSs listen to this PDUs and assess the signal quality of all radio channels and switch to the Radio Cell with the best signal quality.

After this procedure, the token is passed to the next master in the network. It is important that the different Radio Cells overlap and that adjacent Radio Cells operate in different Radio Channel Sets.

2.3.3 Integration of Multimedia-applications into RFieldbus architecture

To integrate multimedia into the RFieldbus (Figure 2.9), tunnelling the TCP/IP telegrams into PROFIBUS telegrams is necessary. This is done in a way that the Quality of service (QoS) for the TCP/IP applications and the timing requirements for the PROFIBUS traffic are not affected.

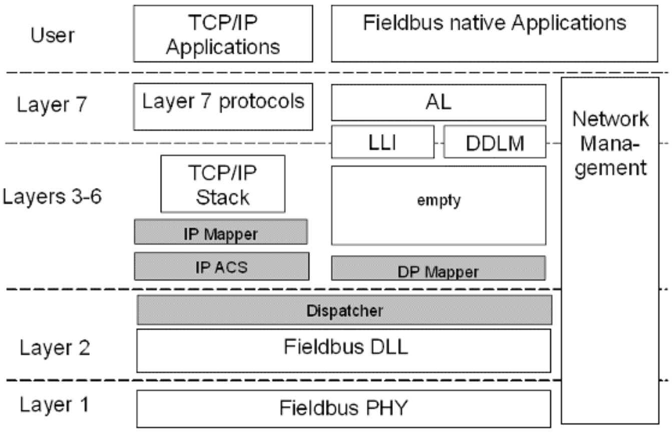


Figure 2.9: Integration multimedia into RFieldbus

This method adapts the master/slave model of PROFIBUS to the symmetric nature of a IP network. It also solves the problem due to the small maximum transmission unit size of PROFIBUS in comparison to the TCP/IP environment. This all happens in a transparent way, from the application point of view. Therefore a dual-stack architecture with extra sub-layers in the standard TCP/IP and PROFIBUS stacks is used.:

- The *IP Mapper* maps the TCP/IP services into PROFIBUS DLL services. It also takes care of the identification, fragmentation and reassembly of the PDUs. An other function of the IP-mapper is the integration of the client/server model of the IP protocol into the Fieldbus communication model.
- The *IP ACS* (Admission Control and Scheduling) manages the control and limitation of network resources usage by TCP/IP applications. It also implements scheduling policies to provide the required QoS (Quality of service) for the multimedia applications.
- The *Dispatcher* sub-layer provides queues, implementing the priority of service requests. It transfers the requests from the queues, in order of priority, limited by master allocation time.

3 Setting up the RFieldbus Network

Based on [3]

To guarantee the real-time behaviour of a distributed computer-controlled system (DCCS), it is important to check before run-time if the worst-case execution of its tasks is smaller then the admissible response time. In a hybrid wired/wireless Fieldbus network, where the transmission delay through a series of different mediums can be much longer then the length of the message itself, the message duration is dependent on the duration of the request and response PDUs and on the number and type of physical mediums that the PDUs must cross between initiator and responder. Such a message duration includes both the duration of the message itself and the duration of its transmission time. It is also dependent on the extra idle time that must be inserted between consecutive PDUs in the network.

Note that throughout the thesis, parameters denoted as ‘T’ are expressed in bit times while parameters denoted as ‘t’ are expressed in seconds.

3.1 Important timing parameters in the network

3.1.1 Message turnaround times and Slot Time parameter

In PROFIBUS, an acknowledged message transaction involves a request PDU from the initiator followed by a correspondent response/acknowledgement PDU from the responder (master or slave). It is possible that the request and response PDUs have to be relayed by one or more ISs (LSs or LBSs) before reaching its destination, e.g. initiator and responder in different Communication Domains (e.g. D^i is Wired Domain and D^j is Wireless domain), in our case a hybrid wired/wireless RFieldbus network.

The **Responder Turnaround time (*t_{rt}*)** for a message transaction can be defined as the time elapsed since a responder ends receiving a request PDU, until it starts transmitting the correspondent response PDU. It can also be referred as the time interval between the end of the request transmission and the beginning of the response reception when both responder and initiator are in the same Communication Domain. As referred to Figure 3.1 we consider a message transaction between Initiator (I) and Responder (R1) both in the same Communication Domain D^i. The Request PDU has not to be relayed to another Domain to reach its responder. The responder’s turnaround time is indicated as t_{rt}.

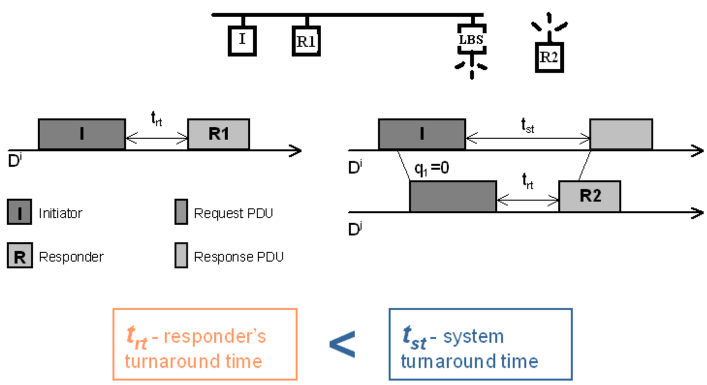


Figure 3.1: Responder's and system turnaround time

So the **System Turnaround time (*t_{st}*)** can be defined as the time interval between the end of the request transmission and the beginning of the response reception, considered in the whole network. As referred to Figure 3.1 we consider a message transaction between Initiator (I) - in

Communication Domain D^i - and Responder (R2) - in Communication Domain D^j – that has to be relayed through an IS with cut-through behaviour.
In order for the initiator to know if there is a problem with the request/response PDU, PROFIBUS provides an important parameter called **Slot Time (*T_{SL}*)**. When the response/acknowledge to a request does not arrive to the initiator in the slot time, the initiator knows that there was a problem (a timeout occurred) and retries this for a predefined number of times or aborts the message transaction. An example is shown in Figure 3.2 in case of an acknowledged request.

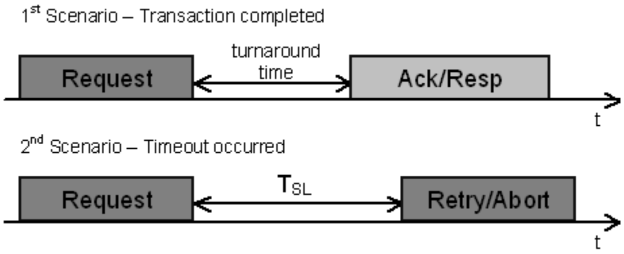


Figure 3.2: Slot time parameter

To set the Slot Time parameter, it is necessary to compute two different components namely T_{SL1} and T_{SL2}.

- T_{SL1} is the maximum time the initiator waits for the complete reception of the first character of the acknowledgement/response PDU after transmitting the last bit of the request PDU. (Figure 3.3)

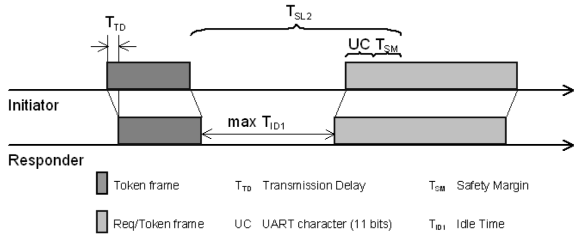


Figure 3.3: Illustration of *T_{SL1}*

- T_{SL2} is the maximum time the initiator waits after having transmitted the last bit of the token PDU until it detects the first bit of a PDU. (Figure 3.4)

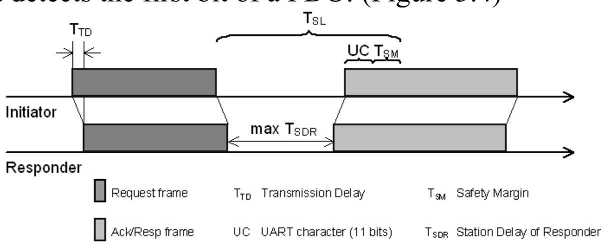


Figure 3.4: Illustration of *T_{SL2}*

Contrarily to the Idle Time parameters, all master stations in the network must set the Slot Time parameter to the same value, which is the maximum between T_{SL1} and T_{SL2}, i.c.:

$$T_{SL} = \max (T_{SL1} , T_{SL2})$$

The computation of the Slot Time parameter depends on the characteristics of every message stream in the Communication Network. Therefore, there is the need to determine the worst-case system turnaround time for all message streams. Message streams fulfil the needs of communicating tasks in master ESs. Additionally, to carry out a worst-case analysis, there is also the need to compute the worst-case duration of all message transactions (request duration, communication latencies, response duration and inactivity time), as briefly described in 3.2.3.

3.1.2 Idle Time Parameters

Every master has to insert a predefined idle time T_{ID} between consecutive message cycles due to the physical layer (PhL) requirements (namely for synchronisation). This period of physical medium inactivity has a duration of a predefined number of bits (Figure 3.5)

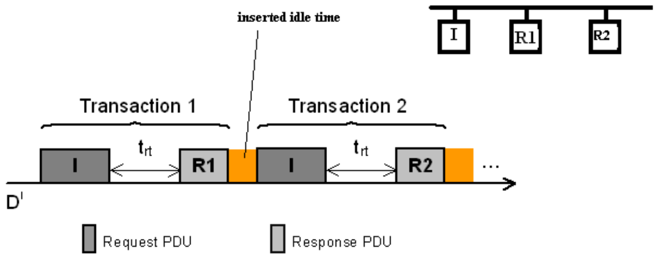


Figure 3.5: Idle time parameter

PROFIBUS defines two different DLL idle time parameters defined, namely T_{ID1} and T_{ID2}, related to acknowledged and unacknowledged requests, respectively.

- After an acknowledged, response or token PDU, a master has to insert an idle time given by (Figure 3.6):

  T_{ID1} = max (T_{SYN} + T_{SM}, min T_{SDR}, T_{SDI})

  with:
  - T_{SYN} is the synchronisation time, the minimum time interval during which each station must receive idle state from the physical medium (33 bits);
  - T_{SM} is a safety margin;
  - min T_{SDR} is the minimum station delay of the responders;
  -T_{SDI} is the station delay of the initiator also referred as responder's turnaround time T_{RT} (see above)

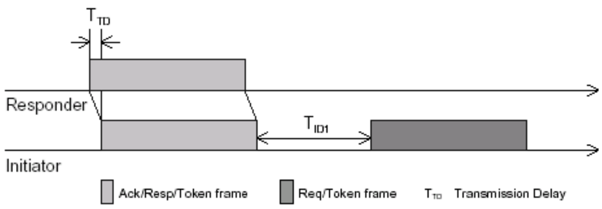


Figure 3.6: Illustration of T_{ID1}

- After an unacknowledged request PDU a master has to insert an idle time given by (Figure 3.7):

T_{ID2} = max (T_{SYN} + T_{SM}, max T_{SDR})

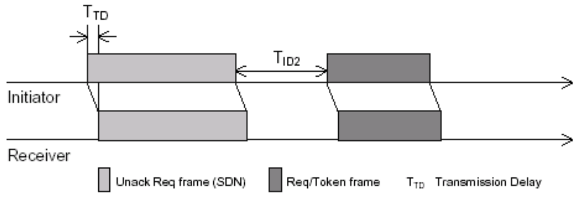


Figure 3.7: Illustration of T_{ID2}

The idle time parameters must be set as a pair (T_{ID1} ,T_{ID2}) for each master and for the different masters this pair can be set differently.

3.2 Guaranteeing real-time communications in RFieldbus networks

In a heterogeneous network, such as the RFieldbus, there are different bit rates and PhL PDU formats (Chapter 2). The relaying of the PhL PDUs between wired and wireless domains is carried out by Intermediate Systems (ISs), acting at the PhL level. This results in a broadcast network, where every ES receives every transmitted PDU. If in a certain time-interval a lot of PDUs must be relayed by the IS, increasing queuing delays leading to unpredictable turnaround times may occur, as it will be outlined next.

3.2.1 Increasing queuing delay

The PROFIBUS MAC mechanism guarantees that only one ES is allowed to transmit at a given moment in time. Nevertheless, the fact that there are different PhL PDU formats and bit rates in the different Physical Media, this may lead to cumulative pending messages in the IS, i.e. traffic congestion.
In Figure 3.8 there is a sequence of message transactions illustrated, between an initiator (I) and a responder (R1) – both in the same Communication Domain D^i, and the resulting PhL PDUs in Communication Domain D^j .

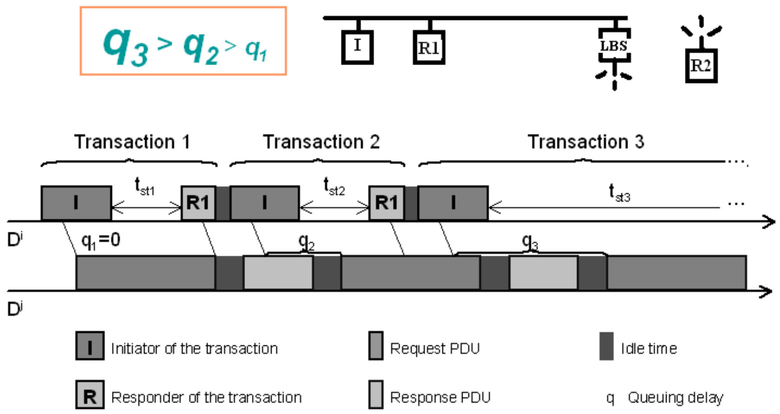


Figure 3.8 Increasing queuing delay

In Figure 3.8 we suppose an IS with cut-through-behaviour and assume (for simplification) the PhL PDU-duration in D^j twice the duration in D^i. In the figure we notice an increasing queuing delay (q3>q2>q1), caused by the different Physical media. This does not impact message transactions in one Communication Domain, but for message transactions that has to pass one ore more Communication Domains – pass between one or more ISs – problems occur. The system turnaround time t_{st} can get unbounded without any adaptation of the traffic. For instance - in Figure 3.8 - the request from transaction 3 is addressed to a responder in Communication Domain D^j, the system turnaround time t_{st3} will be affected by the cumulative queuing delay (q_3) in the IS. Notice that even a sequence of short length PDUs may lead to high queuing delays (with unbounded worst-case message response times)

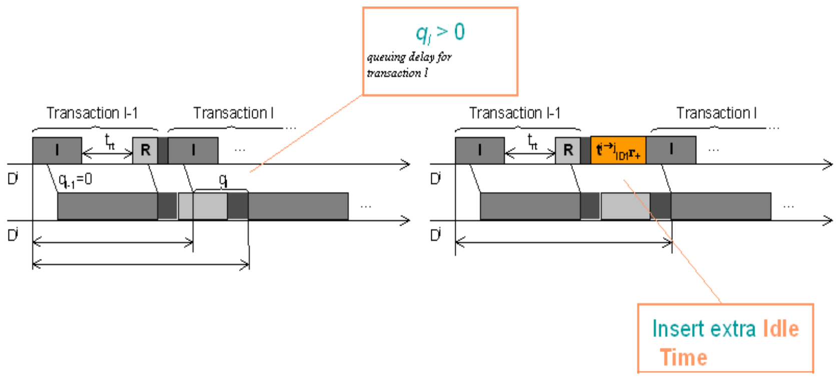
3.2.2 Inserting idle time to adapt heterogeneous media

The problem of the queuing delay depends, among other factors, on the number and duration of consecutive transactions where both initiator and responder belong to the same Communication Domain D^i.

A solution for this problem is to insert an additional idle time before issuing the next message transaction [3,12,13]. The inserted additional idle time must guarantee that there is no increasing queuing in the ISs. On the other side, it can not avoid queuing delays in some ISs between initiator and responder of a transaction or between a master ES and its successor when passing the token.

As explained above, every master ES in PROFIBUS holds a pair of Idle Time parameters (T_{ID1},T_{ID2}). For a traditional wired network, with only one Communication Domain, all master ESs may set their idle time parameters to the minimum default value (T_{ID1m} ,T_{ID2m})[1], which is usually adequate to cope with bit synchronisation requirements. The traffic adaptation to avoid increasing queuing delay in the interconnection of heterogeneous physical media is based on the addition of extra idle times, represented by t_{ID1+} and t_{ID2+}.

As depicted in Figure 3.9(a) there is no additional idle time inserted by the master ES. Transaction l experiences a queuing delay in the IS. By inserting an appropriate inactivity (or idle) time after receiving a response/acknowledgement, it is possible to guarantee that the next PhL PDU will experience no queuing delay in the first IS - Figure 3.9(b) -. The same reasoning can be followed for the additional inserted idle time after receiving the token.



(a) **(b)**

Figure 3.9: Inserting idle time to adapt heterogeneous media

[1] *m* stands for 'minimum'

- Inserted idle time after receiving a response/acknowledgement or token in a hybrid wired/wireless network:

  Taking into account that the PROFIBUS protocol supports only one register for T^i_{ID1}, there is the need to aggregate both the 'minimum' idle time T_{ID1m} with the inserted idle time T^i_{ID1+} in one variable, for each master ES:

  $$T^i_{ID1} = T_{ID1m} + (t^i_{ID1+} \text{ x } r^i)$$

  with: Physical Media i and r the baudrate

- Inserted idle time after sending an unacknowledged request in a hybrid wired/wireless network:

  For the same reason as explained for T^i_{ID1}, there is for each master ES:

  $$T^i_{ID2} = T_{ID2m} + (t^i_{ID2+} \times r^i)$$

  with: Physical Media i and r the baud rate

This adaptation will obviously reduce the number of transactions per time unit when the responder is not in the same Communication Domain as the initiator, but the advantage of avoiding traffic congestion is enormous.

3.2.3 Slot Time Parameter and Duration of message transactions

Worst- case system turnaround time

All master ESs must have the Slot Time Parameter (T_{SL}) set before starting system's operation. To set the T_{SL1} parameter, it is necessary to compute the worst-case system turnaround time of all message transactions (t_{st}), taken in consideration that in the case of hybrid wired/wireless networks a PDU sent by an initiator to a responder might have to be relayed through several ISs. The same reasoning, applied to the case where a master ES passes the token and waits for the next master station to transmit, permits to compute T_{SL2}. On one head, T_{SL} must be set large enough to cope with the extra delays introduced by the ISs. On the other hand, T_{SL} must be set small enough so that the system can handle failures as early as possible.
The inserted idle time guarantees that there is no **increasing** queuing delay in the ISs. However queuing delays may occur in some ISs (except in the first) between initiator and responder of a transaction (or even between a master ES and its successor, when passing the token). But the worst-case queuing delay Q – affecting any request PDU – can be computed.
As an illustration example, consider a Communication Network with two ESs (I and R), where every message transaction must be relayed by three ISs. The timing diagram for a message transaction in this network is depicted in Figure 3.10. C_{ack} is the duration of an acknowledged message transaction.

$$C_{ack} = C^1_{Lreq} + t_{st} + C^1_{Lresp} + t^1_{ID1m} + t^1_{ID1+}$$

With

- C^1_{Lreq} - duration of the request PDU
- t_{st} - system turnaround time
- C^1_{Lresp} - duration of the response PDU
- t^1_{ID1m} - minimum default idle time
- t^1_{ID1+} - inserted additional idle time

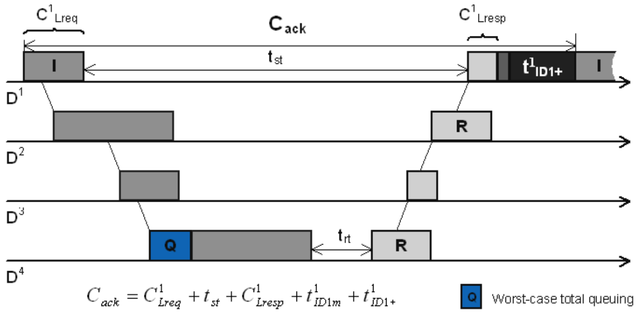


Figure 3.10: Duration of a request/response transaction (C_{ack})

The worst-case total queuing *Q* for this transaction is also shown. This worst-case queuing delay *Q* is a component of the worst-case system turnaround time t_{st} for any message transaction.
$t_{st} = Q + t_{stn}$ with t_{stn} is the system turnaround time assuming no queuing delay. This is shown in figure 3.10 with $t^{i \to j}$ the time to relay the PDU from one Communication domain to another.

Notice that Domains D^2 and D^4 have a physical medium, while D^1 and D^3 have another physical medium. The length of the request and response PDUs impact the t_{stn} value. To considerer worst-case situations, the maximum responder's turnaround time t_{rt} value has to be assumed ($t^{max}{}_{rt}$).
To set the Slot Time Parameter, it is necessary to determine the worst-case system turnaround time for all message transactions in the whole network. To achieve this, the worst-case system turnaround time for each master ES (taking all possible message streams for that ES into account) has to be determined and then choose the worst-case system turnaround time in the network.

Setting the Slot Time Parameter

All master ESs in the Communication Network must set the Slot Time parameter to the same value, which is the maximum between T_{SL1} and T_{SL2}.

$$T_{SL} = \max (T_{SL1} , T_{SL2})$$

After having computed the worst-case system turnaround time for all message transactions (streams) in the Communication Domain, T_{SL1} can be set as follows:

$$T_{SL1} = \max \{T_{st} (S[i])\}$$

With *i* the index of the message stream in the set of *S*

T_{SL2}, the worst-case system turnaround time after transmitting the token PDU (*T* in Figure 3.11) can be set as follows. t_{st_token} is defined as the maximum time the initiator of the token PDU waits until it detects the first bit of a PDU, either a request or the token transmitted by the ES that received the token.

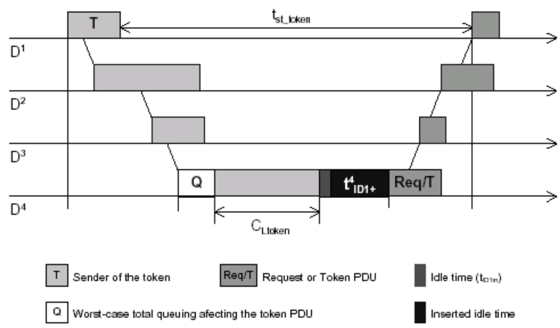


Figure 3.11: Illustration of T_{SL2}

Also here can be proved that there will be no increasing queuing delays but that there can exist a queuing delay in some ISs (except in the first) between initiator and responder of this transaction.

After having computed the worst-case system turnaround times after transmitting the token for all master ESs (M[i]) in the Communication Network.

$$T_{SL2} = \max \{T_{st\_token} (M[i])\}$$

3.2.4 Parameters for the Mobility Management Mechanism

The basics of the Mobility Management mechanism adopted for the addressed hybrid wired/wireless communication network were already presented in Chapter 2.
In order for this mechanism to be compatible with the characteristics of PROFIBUS, after the Mobility Master (MobM) triggers the mobility management mechanism, it must insert an adequate idle time corresponding to the duration of the mobility management procedure, before issuing another transaction or passing the token. Since the Beacon Trigger (BT) PDU is a PROFIBUS SDN (unacknowledged request) PDU, the idle time to be inserted by the MobM after transmitting the BT PDU must be implemented by using $T_{ID2,MobM}$. This value is related to the worst-case duration of the mobility management procedure (which happens periodically). The duration of this mobility management procedure depends on the number of beacons (*nb*) that each LBS must transmit, after having received (and relayed) the BT PDU. The *nb* transmitted by each LBS can be different, since they can receive the BT PDU at different instants (depending on the number of LBSs and on the Physical Media in the path between the MobM and each LBS). Make notice that $T_{ID2,MobM}$.is not the same value as theT_{ID2} set in the usual Masters.

3.3 The System Planning Application (SPA)

A System Planning Application (SPA) was developed based on the algorithms proposed in [*3*]. It computes all the necessary timing parameters (both PROFIBUS and RFieldbus-specific) for the real-time operation of the network, depending on the network topology, on the characteristics of the message streams and on the types of physical media (wired/wireless). The values for all relevant parameters (T_{ID1}, T_{ID2}, T_{st}, and C_{ack} for all message streams, T_{SL1}, T_{SL2}, T_{SL} and the mobility management parameters) are obtained (Figure 3.12). The T_{st} and C_{ack} can be compared with the real traffic performance, by means of a PROFIBUS network analyser, for every message stream. T_{ID1}, T_{ID2}, T_{SL} and mobility management parameters can be used to set up the network.

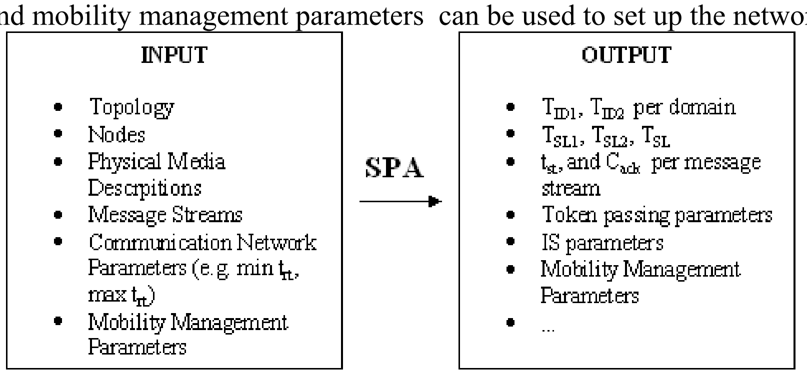


Fig 3.12 Scheme of the System Planning Application

The System Planning Software Application provides the user with a complete but also very simple and intuitive interface. After defining the targeted network topology (wired and wireless nodes – masters or slaves –, wired and wireless physical media and intermediate systems) and after setting some Physical Layer characteristics, the adequate parameters are automatically computed . The System Planning Software Application was used to provide the theoretical results presented in Chapter 5 (Matching Analytical vs. Experimental results).

4 RFieldbus Manufacturing Automation Field Trial

Based on [5] and [6]

To test and assess the RFieldbus system there were developed two field trials, namely process and manufacturing automation. These pilot applications show the possibilities of the RFieldbus system in an industrial environment. In what follows we will take a closer look at the RFieldbus Manufacturing Automation Field Trial (MAF).
The MAF includes a hybrid wired/wireless network with mobile nodes. These mobile nodes are 2 transportation vehicles and handheld terminals for supervision and maintenance. All this is supported by the traditional distributed control systems and multimedia application services. The MAF is a suitable platform to analyse the timing behaviour of both real-time control data and multimedia data.

4.1 System layout and functionality

The layout of the manufacturing application is presented in Figure 4.1. When a new part arrives (is transported to this subsystem), it must be classified according to a certain criteria and must be distributed to storage buffers or to the next stage of the manufacturing process. This next stage could be further processing (e.g. cutting, drilling) or just transporting a storage buffer to a warehouse. Roller belts and different pneumatic equipment are used to transport and distribute parts to output buffers, according to their type. When output buffers are full, they are moved (either by an automatic vehicle, a robot arm, or an operator) to the respective unload station, in order to be emptied. Considering the classification criteria, at this stage each part is distinguished by its colour.

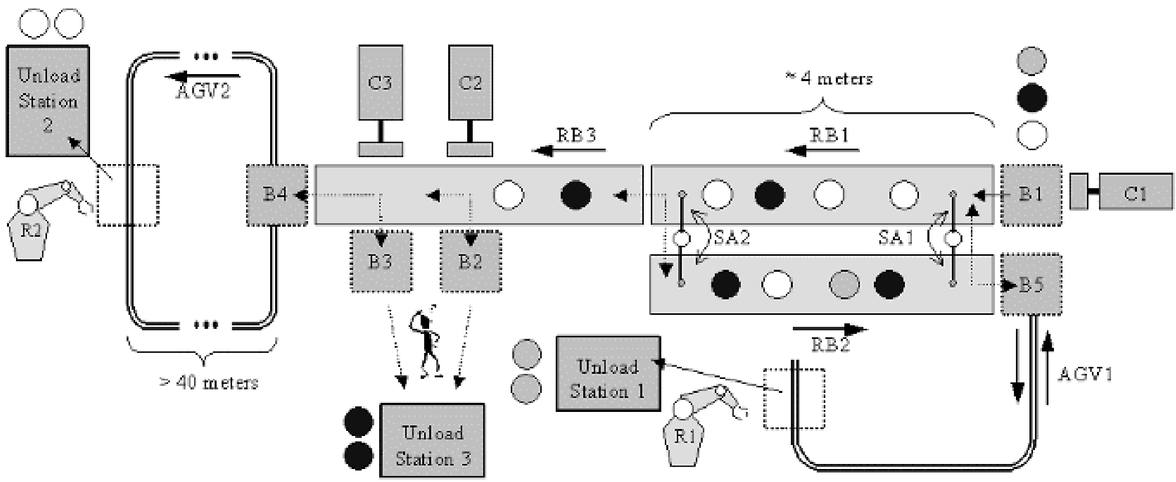


Figure 4.1: Mechanical system layout

The input buffer (B1) stores black, white and grey (defective) parts, which are sequentially pushed into the roller belt (RB1). SA2 (a swivelling double arm with suction cups) pushes grey parts to RB2. Grey parts go into B5. If this buffer is full or in transit grey parts must circulate around RB1-RB2. When B5 is full, AGV1 moves to U1, for unload operation carried out by a robot arm (R1) and an operator, and then returns to the initial position. White and black parts go into RB3, and black parts are pushed into output buffer (B2). When B2 is full, an operator is warned, in order to unload it. Meanwhile B3 must be used to receive black parts. If both B2 and B3 are non-operational (full or in transit), black parts must circulate in RB1-RB2. White parts go into B4, until it is full or if it is in transit. When B4 is full, AGV2 moves to U2, for unload operation carried out by R2. White parts must circulate around RB1-RB2, if B4 is unavailable.

4.2 The network

The MAF is supported by a RFieldbus hybrid network with both wired and wireless communication nodes. RFieldbus mobility requirements impose the use of wireless nodes, such as transportation vehicles. It also involves the use of wired segments, i.e. a hybrid wired/wireless fieldbus network. The interconnection of the different wired and wireless domains is achieved through Inermediate Systems (ISs) acting as Link Stations (LSs) and/or Link Base stations (LBs). Figure 4.2 depicts the structure of this network. In this case, LBS1 and LBS2 interconnect the two wireless domains(WL1 and WL2) and the wired segment (WR). There are two PROFIBUS Masters in the system, namely PC1 and the Mobility Master (in PC2). All the other nodes, PC2- PC6, ET 1-2 and MM1-2, are PROFIBUS slaves. PC2 has two RFieldbus interface cards. One for the slave in PC2 (standard traffic) and one for the Mobility Master functionality. It should be noted that wireless stations were implemented as traditional PROFIBUS wired stations with corresponding LSs.

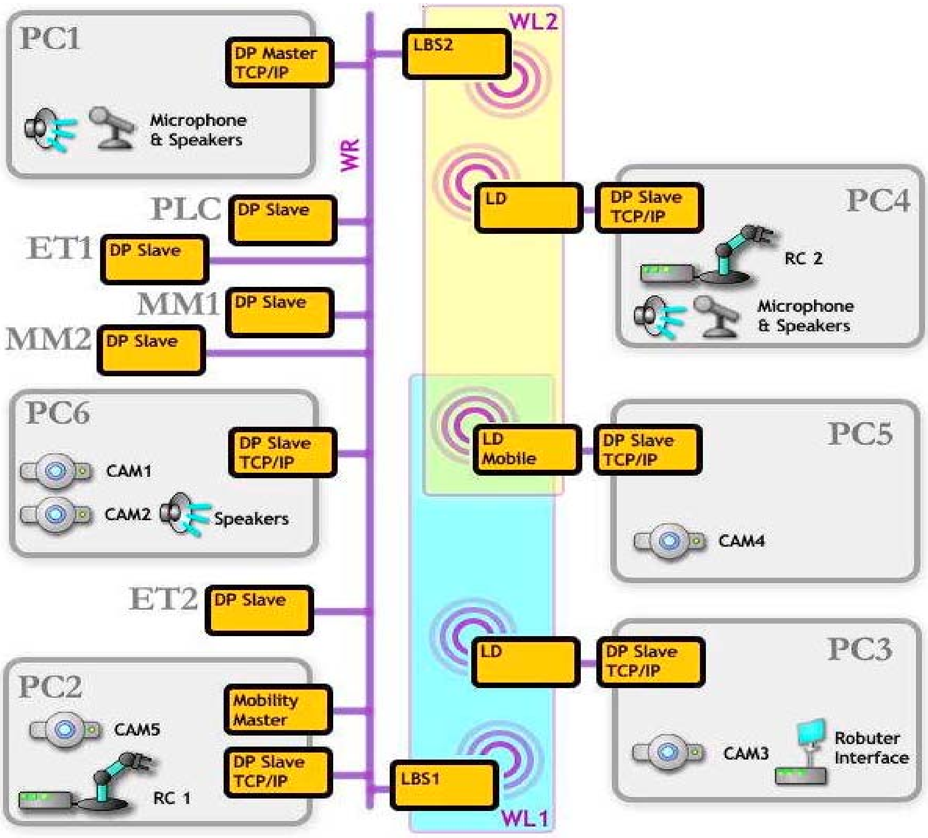


Figure 4.2: MAF network layout

WR	Wired Segment – Standard PROFIBUS cable
WL 1	Wireless Segment (Main Area) –RFieldbus Radio
WL 2	Wireless Segment (Second Area) –RFieldbus Radio
DP Slave	Standard PROFIBUS DP Slave Device
DP Slave TCP/IP	RFieldbus Slave with Multimedia support
Mobility Master	RFieldbus Mobility Master
LBS	Linking Base Station
LD	Linking Device (=Link Station (LS))
LD Mobile	Linking Device with mobility support (handoff) (=mobile Link Station)

Table 4.1 Abbreviations used in Figure 4.2

4.3 Message streams

4.3.1 DP message streams

The different PROFIBUS-DP applications (Fig 4.3) and the DP message streams used in the MAF are briefly described next. Figure 4.3 outlines these DP applications.

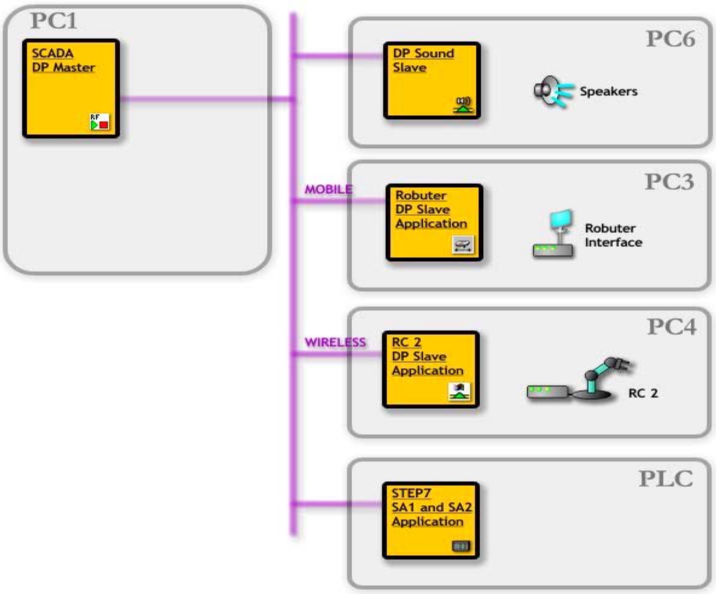


Figure 4.3: DP applications

SCADA

SCADA (supervisory control and data acquisition) is a category of software application program for process control, for gathering of data in real time from remote locations in order to control equipment and conditions. The SCADA application is a fundamental application in the RFieldbus MAF. It is a windows multi-threaded application, responsible for coordinating the entire system. It acts together with the cooperating applications (Disk Colour Detection [Master], Robot Arm Operator [Master]), working in the master PC, in order to effectively orchestrate the system's equipment through both TCP/IP and PROFIBUS communication. Additionally, the SCADA application cooperates with the Human Machine Interface (HMI) in a complementary fashion. The SCADA application serves the HMI with information about the system events (sensors and actuators, control, errors, etc), so that the HMI is aware of the system's state and is able to both show this information in a user-friendly fashion and deliver part of this information to other system applications.

The *SCADA DP Master* is a sub-system of the entire SCADA application, dedicated to the acquisition of data. In the MAF, the hardware responsible for receiving the data is the _isPRO multiboard, provided by IFAK [14]. The data is originated from the several equipments in the system (Slaves), which communicates via PROFIBUS with a central PC (Master), responsible for the coordination of the system. The _isPRO multiboard is a PROFIBUS card, in this case acting as a PROFIBUS Master, that incorporates the defined extensions for the RFieldbus project. At system initialisation time the DP Master is responsible for correctly configure all the Slaves in the system and initiate the appropriate cyclic data transfers in a timely manner.

HMI

This application fulfils two objectives:

1. provide an user-friendly interface to the SCADA system,
2. provide SCADA related information to the Intranet Server.

DP Sound Slave

The DP Sound Slave acts as a normal PROFIBUS Slave, enabling the system to issue warning sounds, using a simple PROFIBUS data exchange, consisting of a master request with a data field length of 2 bytes and an identical response. This DP data exchange is generated every communication cycle. This message stream will be referred to as S1 (Figure 4.4) .

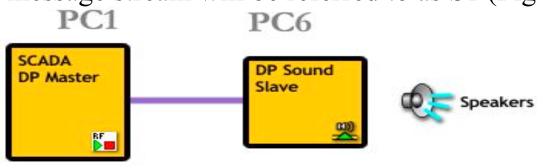


Figure 4.4: DP Sound Slave Application Stream

Robuter DP Slave Application

This DP data exchange is triggered every communication cycle and consists of a request from PC1 with 2 bytes of data used to signal the robuter controller and receive from PC3 a identical sized PDU containing the robuter's location/status. This data exchange uses a mobile RFieldbus link. This application cooperates with the robuter controller in order to receive commands from PC1 and transmit the robuter's localization. The robuter stays in the buffer station until the buffer is full. Then PC3 sends a command to PC1 and the master will send the appropriate commands in order to send the robuter to the unload station. When unload operation ends, PC3 will inform PC1 and receive the appropriate order to return to the original position. This message stream will be referred to as S2 (Figure 4.5).

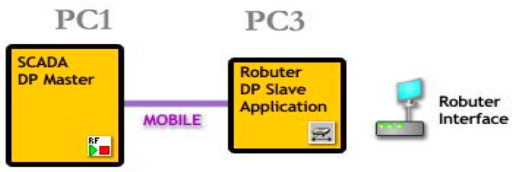


Figure 4.5: Robuter DP Slave Connection

RC2 DP Slave Application

This DP Slave application cooperates with the robotic controller applications, in order to initialise and manipulate the robot. Additionally, the robotic controller also, serves the DP Slave with information about the detection of the AGV arrival. This data is used in order to enable the master to signal if the AGV unload should be enabled and, in the opposite direction, to signal the master whether the AGV is detected in the unload station position of not. This PROFIBUS data exchange between the master and PC4 (via wireless RFieldbus link) is triggered every communication cycle and both request and response have a two byte data field. This message stream will be referred to as S3 (Figure 4.6).

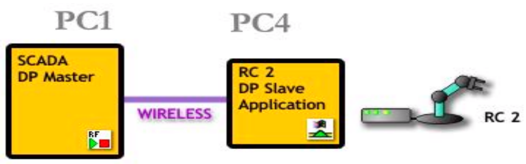


Figure 4.6: RC2 DP Slave Application

MM1/MM2

The two micromasters consist in variable speed drives used in RFieldbus field trial in order to be able to vary the speed of the transportation system. The stream between PC1 and the two micromasters consist in a request from the first with 12 bytes data field and a response from the second with the same size. This stream will be referred to as S4 and S5 (Figure 4.7).

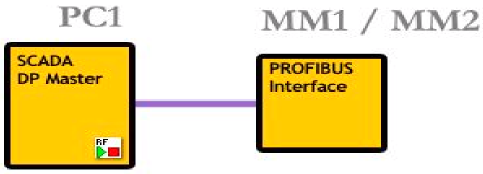


Figure 4.7: MM1/MM2 Slave Connection

ET1/ET2

The DP traffic generated between the overall control system (PC1) and the PROFIBUS Concentrators ET1 and ET2 is generated every communication cycle and consists of a master request with a 3 bytes data field and a consequent response with a 4 byte data field. This traffic is responsible for the interconnection between PC1 and the sensors and actuators of two control subsystems, one for the roller-belts RB1 and RB2 and the other for the RB3. It is also used to command the indication lights for the manual buffers. The request stream consists on the values read from the sensors (infrared sensors for parts and magnetic position detectors for the cylinders) of the referred control subsystems. These streams will be referred as S6 and S7.(Figure 4.8)

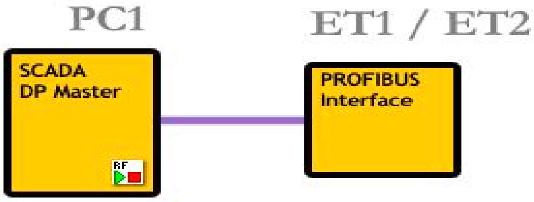


Figure 4. 8: ET1/ET2 Slave Connection

SA1 and SA2 Application

This data stream is generated every communication cycle and consists of a request from PC1 with 16 bytes and a response of the same length from PLC. The control of the 2 Swivel Arms (SA) is done directly by this Siemens PLC, receiving the appropriate controls from PC1. This message stream will be referred to as S8 (Figure 4.9).

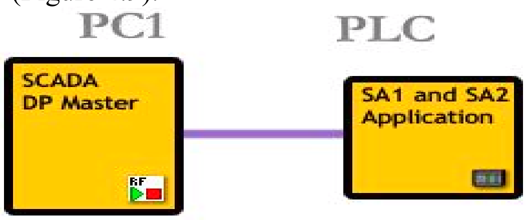


Figure 4.9 : PLC Slave Connection

4.3.2 Multimedia IP Applications

The different multimedia IP applications (Figure 4.11) and the IP message streams used in the RFieldbus MAF are briefly described next. The streams inside PC1 and between PC1 and the Intranet are not important for the study of the timing behaviour in the network.

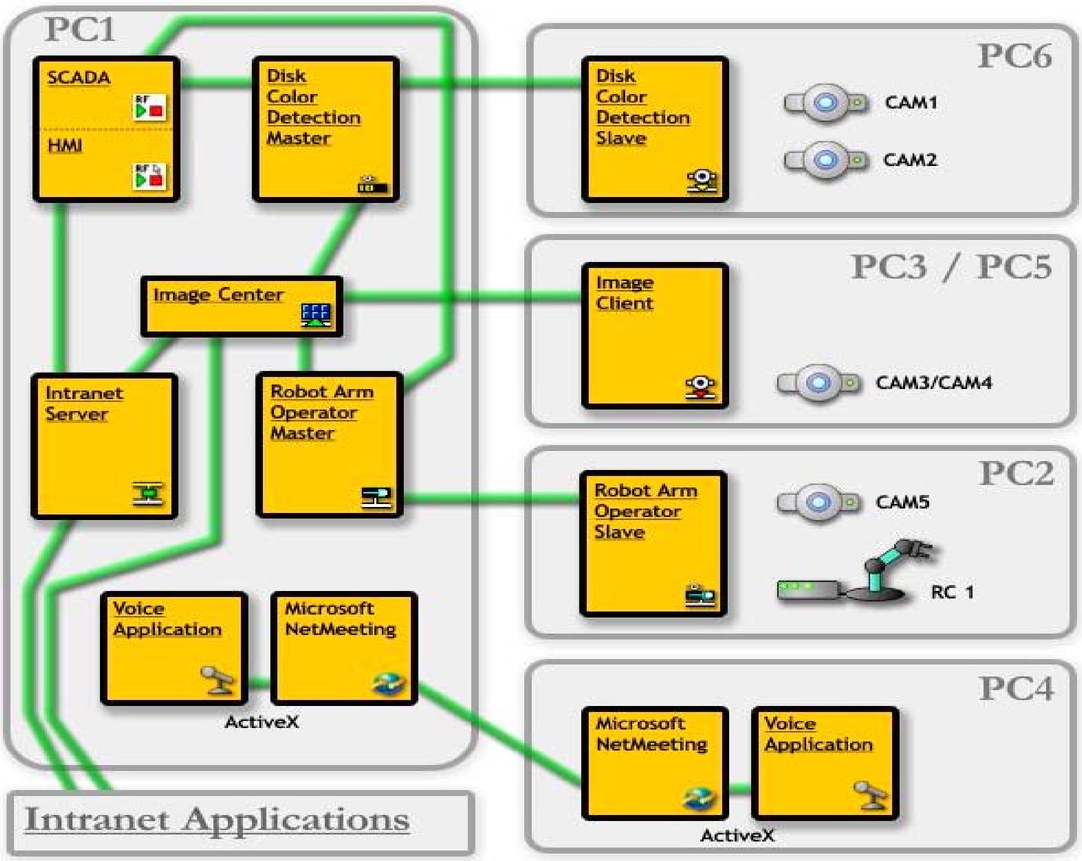


Figure 4.11: IP applications

Disk Colour Detection Master/ Disk Colour Detection Slave

This is a TCP/IP connection between Disk Colour Detection Master (PC1) and Disk Colour Detection Slaves (PC6), where the slave works as a TCP server and the master as a TCP client. The slave application uses a video camera to take snapshots of the roller-belts. When a part is detected a snapshot is sent to the master, which will be used to detect the colour of the part. The images have a size between 2 and 3 Kbytes. When the part is not detected anymore a small 4-byte message is sent to the master. A new part is feed into to the system every 3 seconds, this will be also the maximum periodicity of this stream. This stream will be referred to as S9 (Figure 4.12).

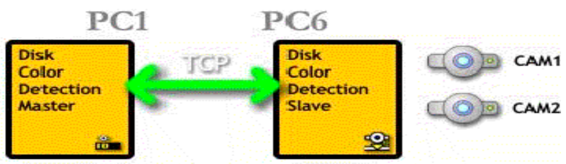


Figure 4.12: Disk Colour Detection Stream

Image Centre/ Image Client

This is a UDP stream between Image Center (PC1) and Image Client (PC3/PC5). The Image Client Application periodically (every second) sends a snapshot from the video camera (CAM3/CAM4) on top of an AGV. The images are compressed in JPEG format and the length is between 2 and 3 Kbytes. The Image Centre will distribute images from all the cameras in the Field Trial to devices in the Intranet. It also provides image display of all the cameras to the operator in PC1.
The images are reset to the Intranet in a given interval. The active image streams are managed by the Intranet Server that sends Windows Messages to the Image Center activating/ deactivating the streams. These streams are referred to as S10 and S11 (Figure 4.13).

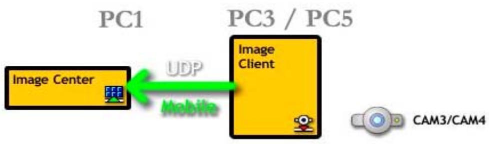


Figure 4.13 : Image Client Image Stream

Robot Arm Operator Master/ Robot Arm Operator Slave.

There is a TCP/IP connection between Robot Arm Operator Master (PC1) and Robot Arm Operator Slave (PC2). The slave application has direct control of both the robotic arm RC1 (RS232) and the video camera that is filming the RC1 pick-up area. The TCP/IP connection is used to pass robot operations and snapshot requests from the master to the slave, and the respective responses (from RC1) and images from the slave to the master. The operations, requests and responses are sent as strings of text. The images are compressed in JPEG format and their size is between 8 and 16 Kbytes and are sent from PC2 to PC1 sporadically, whenever there is the need to initiate an unload operation by the robotic arm. . This stream will be referred to as S12 (Figure 4.14).

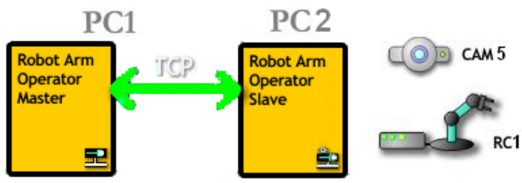


Figure 4.14: Robot Arm Operator Connection

Intranet Server

The Intranet Server runs in PC1 and provides not only information about the system to the devices connected to the Field Trial Intranet but also the possibility to interact with the system. The information is exchanged using UDP packets. The Intranet Server manages the broadcasts of image streams from all the video cameras on the field trial (the broadcast itself is done by the Image Center application).

Voice Application

The voice application uses Microsoft NetMeeting control to send audio grabbed by microphones between PC1 and PC4. PC4 is a station that is far away from PC1 (using a wireless RFieldbus link), and this is the only way for the operators to communicate. The NetMeeting control uses TCP and UDP connections. This application is sporadic and symmetric, that means that could be initiated either by the master or the slave whenever the operators feel the need to communicate. This stream will be referred to as S13 (Figure 4.15).

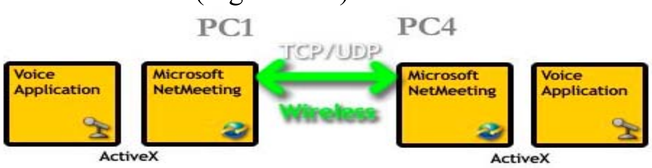


Figure 4.15 : Voice Application

4.3.3 Overview of the different streams

Table 4.2 gives an overview of the different message streams in the MAF, which are described in the previous sections.

Stream	Type	Size of requests (PhL bytes)	Size of answer (PhL bytes)	Description	Initiator ⇔ Responder
S1	DP	11	11	Sound Slave Application	PC1 <=> PC6
S2	DP	11	11	Robuter Slave Connection	PC1 <=> PC3
S3	DP	11	11	RC2 Slave Connection	PC1 <=> PC4
S4	DP	21	21	Speed Roller Belts	PC1 <=> MM1
S5	DP	21	21	Speed Roller Belts	PC1 <=> MM2
S6	DP	12	13	Sensors-Actuators	PC1 <=> ET1
S7	DP	12	13	Sensors-Actuators	PC1 <=> ET2
S8	DP	25	25	Swivel Arms	PC1 <=> PLC
S9	IP	6	255	Disk Color Detection	PC1< => PC6
S10	IP	6	255	Video Monitoring	PC1< => PC3
S11	IP	6	255	Video Monitoring	PC1< => PC5
S12	IP	39	255	Robot Arm Operator	PC1 <=> PC2
S13	IP	255	255	Voice Netmeeting	PC1< => PC4

Table 4.2 Message streams in the network

5 Matching analytical vs. experimental results

5.1 Analytical results (worst-case scenarios)

5.1.1 Input to the SPA

To setup the system (real time operation of the network) the System Planning Application (SPA) calculates all the necessary timing parameters depending on the network topology, on the characteristics of the message streams and on the types of physical media (wired/wireless). (Chapter 3, 3.3)

Topology

Our case study is based on a simplified network topology of the MAF, used as input to the SPA (Figure 5.1). We can recognize on the main PROFIBUS segment the Master(1), Mobility Master(3), 2 LBSs with the correspondent radio cells and a slave(2). This slave represents all the slaves in this segment. There is also a wireless (mobile) segment with a LS and a Slave (4), that represents the two mobile segments with the AGVs and the wireless (not mobile) segment. Between Master(1) and Slave(2) there is a connection where the different streams between those two ES (i.e. the Length of Request (C_{Lreq}) and Length of Response (C_{Lresp}) PDU) must be defined. The same applies for the stream connection between Master(1) and Slave (4). The SPA automatically detects all possible paths for these message streams. This is very important in case of mobile nodes, since they can move from one radio cell to another.

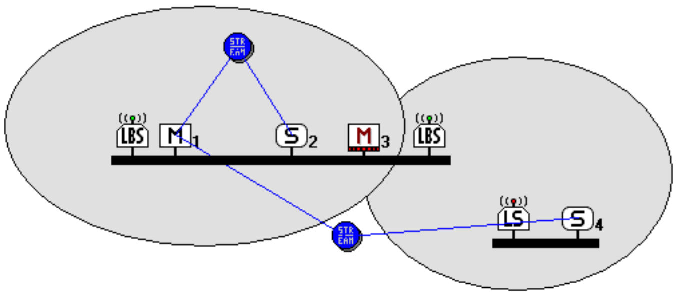


Figure 5.1: Simplified Network Topology of the Manufacturing Automation Field Trial

Physical Media parameters

The specification of the 2 different media used in the MAF is described in Figure 5.2. Medium 1 represents the wired PROFIBUS part (bit rate of 1.5 Mbit/sec) and Media 2 the wireless RFieldbus part (bit rate of 2 Mbit/sec).

	r (MBaud)	IH (bits)	IT (bits)	k (bits)	o (bits)
1	1.5	0	0	3	33
2	2	180	32	0	148

Figure 5.2: Physical Media specification

Mobility Management parameters

Figure 5.3 presents the mobility management input parameters. These values are used by the SPA to compute e.g. the number of beacons (*nb*) that should be issued by the LBSs and the Idle Time that must be inserted by the Mobility Master (Chapter 2, 2.3.4).

Beacon Trigger Length (BTLen)	10	chars
Number of Radio Channels (nch)	3	
Duration of the Beacon (Cbeacon)	100	µs
Time interval between beacons (tbgap)	25	µs
Channel Switching Delay (tsw)	700	µs
Buffering Delay (tbd)	25	µs

Figure 5. 3: Mobility Management Parameters

Network parameters

Figure 5.4 depicts the network-specific parameters set for the MAF. It was assumed that t_{rt_min} = 10µs = 15 bit times and t_{rt_max} = 50µs = 75 bit times are reasonable values, since native PROFIBUS boards have short turnaround times (t_{rt}).

Number of data bits per DLL char (d)	8	bits
Maximum Length of a DLL Request PDU (Lreq_max)	255	chars
Minimum Length of a DLL Request PDU (Lreq_min)	6	chars
Maximum Length of a DLL Response PDU (Lresp_max)	255	chars
Minimum Length of a DLL Response PDU (Lresp_min)	6	chars
Length of the token PDU (Ltoken)	3	chars
Internal Relaying Delay of the IS (trd)	25	µs
Minimum idle time between Phl PDUs (TIDm)	100	bits
Minimum responder's turnaround time (trt_min)	10	µs
Maximum responder's turnaround time (trt_max)	50	µs

Figure 5.4: Communication Network-specific Parameters

Message streams

The lenghts of the request and response PDU must be completed (Figure 5.5) as an input to the SPA.

Message Stream	Initiator ES	Responder ES	LReq (chars)	LResp (chars)
1	M1	S2	11	11
2	M1	S2	6	255
3	M1	S2	12	13
4	M1	S2	21	21
5	M1	S2	25	25
6	M1	S4	6	255
7	M1	S4	11	11
8	M1	S4	39	255
9	M1	S4	255	255

Figure 5.5: Input concerning the message streams

5.1.2 Output from the SPA

The SPA calculates all the necessary parameters to set up the system and outputs a table with the parameters concerning the timing behaviour C_{ack} and t_s, for all the message streams.

System parameters

The idle time parameters can be set in a per-station basis, i.e. each master station can hold different values for the (T_{ID1}, T_{ID2}). However, the algorithm implemented by the SPA computes the pair of Idle Time parameters in a domain basis. Therefore the Idle Time parameters (T_{ID1},T_{ID2}) are computed for both physical media.
The SPA computes T_{ID1} and T_{ID2} considering the extra idle time that is necessary for the hybrid wired/wireless network. While these parameters can not be set directly into the configuration files of the masters, they can be set via the min T_{SDR} and the max T_{SDR} parameters. We know that according to the PROFIBUS standard, i.e. T_{ID1} and T_{ID2} can be computed as follows:

$$T_{ID1} = \max (T_{SYN}+T_{SM}, \min T_{SDR}, T_{SDI})$$
$$T_{ID2} = \max (T_{SYN}+T_{SM}, \max T_{SDR})$$

This means that, assuming a reasonable value of 50 bit times for $T_{SYN}+T_{SM}$, if we consider min T_{SDR} = 393 bit times (this is the T_{ID1} computed by the SPA), T_{SDI} =250 bit times, then T_{ID1} = max (50, 393 , 250) = 393 bit times. The same procedure is followed to set T_{ID2}. In case of the MAF, only T_{ID1} is important for the Master in PC1 because this master station never issues unacknowledged request PDUs. T_{ID2} is only important for the Mobility Master, because this station issues unacknowledged request PDUs (i.e. the Beacon Trigger). For this reason, the same values are set for min T_{SDR} and max T_{SDR} in both configuration files of the masters.

Another important parameter that has to be set in the configuration files of the masters, is $\boldsymbol{T_{SL}}$=max (T_{SL1}, T_{SL2}). All the master stations in the network must set the Slot Time Parameter to the same value. In this case T_{SL}= max (1850, 262) = 1850 μs (= 2775 bit times). Figure 5.6 summarises all the fundamental parameters (except *nb* and C_{ack}, which are depicted in other tables).

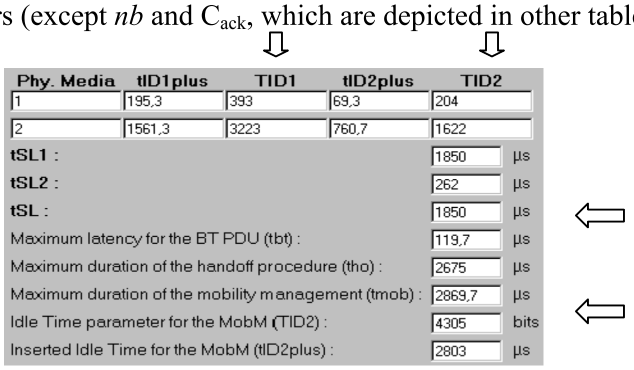


Figure 5.6: System Parameters (μs)

Message Stream Table

The Message Stream Table (Figure 5.7) shows, for every message stream, the possible path (e.g. 1 2 1), t_{st} and C_{ack}. These last two parameters are very important to compare the analytical with the experimental results. Notice that there are worst-case scenarios for only 9 different message streams, while in the MAF 13 are used. 5 message streams are analogous to other ones i.e. same path, L_{req} and L_{resp}, so they are not considered.

Message Stream	Initiator ES	Responder ES	LReq (chars)	LResp (chars)	Path (M,...)	tstn (µs)	Q (µs)	tst (µs)	Cack (µs)
1	M1	S2	11	11	1	50	0	50	473,3
2	M1	S2	6	255	1	50	0	50	2226
3	M1	S2	12	13	1	50	0	50	495,3
4	M1	S2	21	21	1	50	0	50	620
5	M1	S2	39	255	1	50	0	50	2468
6	M1	S2	25	25	1	50	0	50	678,7
7	M1	S4	6	255	1 2 1	1116	664,7	1780,7	3956,7
8	M1	S4	11	11	1 2 1	382	664,7	1046,7	1470
9	M1	S4	255	255	1 2 1	1850	0	1850	5852

Figure 5.7: Message Stream Table

Intermediate Systems Table

The Intermediate Systems Table (Figure 5.8) shows parameters concerning the mobility in the Link Base Stations (LBSs) where the most important is the number of beacons each LBS must transmit (*nb*), upon receiving and relaying the BT PDU.(Chapter 2, 2.3.4)

SIS	Path	tbtn (µs)	Q (µs)	tbt (µs)	t'bp (µs)	nb	tbp (µs)	tmob (µs)
LBS1	1 2	119,7	0	119,7	2675	22	2750	2869,7
LBS2	1 2	119,7	0	119,7	2675	22	2750	2869,7

Figure 5.8: Intermediate Systems table

Token Passing Table

The Token Passing Table (Figure 5.9) shows parameters concerning the passing of the token between the masters e.g. t_{st_token} (system turnaround time after transmitting the token PDU). In this case the passing of the token takes place between the Master in PC1 and the Mobility Master.

Sender ES	Receiver ES	Path	Sum_tsrtoker (µs)	Q (µs)	tID1_ndp (µs)	MaxSum_tsr (µs)	Ctoken_ndp (µs)	Ctoken_1 (µs)	tst_token (µs)
M1	MobM3	1	0	0	262	0	22	22	262
MobM3	M1	1	0	0	262	0	22	22	262

Figure 5.9 Token Passing Table

5.2 The PROFIBUS network analyser

Based on [7]

To carry out the measurements on the PROFIBUS network, a network analyser was used. In addition to monitoring bus activities, the PROFIBUS Analyser provides analytic functions such as statistics and error watching.
When the analyser is started, the schematic window appears (Figure 5.10). This window shows the current operating status. The connections represent the way the data stream passes through the analyser's hardware and software components.

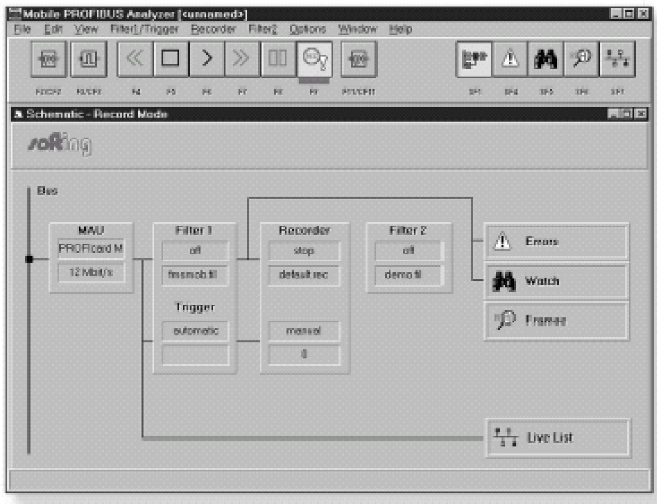


Figure 5.10 PROFIBUS Analyser

The Analyser provides functions for recording and analysing bus traffic in two different modes:

- *On-line mode*
  Here, the analyzer listens to the signals on the bus. On-line analysis primarily consists of representation and interpretation of data. Major features are therefore LIVE LIST to check which devices are available on the bus and various functions for data analysis such as HISTOGRAM, STATISTICS, ERRORS and WATCH. (Table 5.1) These functions are similar to the functions for off-line data analysis. In order to analyze bus traffic over a longer period of time, the user is able to define hardware filter and trigger conditions.
- *Off-line mode*
  In the off-line mode, the PROFIBUS Analyzer performs a detailed examination of the previously recorded data file. The Bus Analyzer decodes the information contained within the PDUs and displays the information depending on the selected layer and display mode. The various display possibilities range from simple hexadecimal through disassembled representation on the FDL (Fieldbus Data Link Layer), the LLI (Lower Layer Interface), the FMS (Fieldbus Message Specification) or DP (Decentralized Periphery) layer. An additional software filter can be defined for extended data analysis.

HISTOGRAM	Offers detailed information about PDU formats
STATISTICS	Shows the general bus load
ERRORS	Displays a statistical list of both, physical and syntactical errors
WATCH	Displays the current values of PROFIBUS PDUs transferred on the bus

Table 5.1: Functions for data analysis

5.3 Tackling with additional implementation delays

The PROFIBUS analyser provides the possibility to measure directly the t_{st}, T_{st}, t_{ID}, T_{ID}, t_{SL}, T_{SL} parameters. C_{ack} and the PDU-duration can be obtained in an indirect way. In this Chapter, we will carry out a detailed analysis of the experimental timing behaviour of all the message streams in the MAF and compare the theoretical (worst-case) values, computed by the SPA(See 5.1.2), with the experimental results.

5.3.1 Experimental results for the original configuration

In the configuration files of the master ESs the following parameters, according to the values computed by the SPA, are set (See 5.1.2).

T_{ID1} = 393 bit times

T_{SL}= 2775 bit times

*Remarks

-The worst-case message duration C_{ack} (See Chapter 3, 3.2.3.1) is defined as:

$$C_{ack} = C_{Lreq} + T_{st} + C_{Lresp} + T_{ID1} \text{ [bit times]}$$

with $T_{ID1} = T_{ID1m} + T_{ID1+}$

-The measured values are indicated with the following symbols: $C^m{}_{ack}$, $C^m{}_{ack\_withoutidletime}$, $T^m{}_{ID1}$ and $T^m{}_{st}$.

-Every PC in the MAF contains a RFieldbus board as network interface(Chapter 4, 4.4.1).

-Measurements were not carried out for S9 and S11 due to implementation constraints

We only discuss in detail a subset of the message streams, since the other have an analogous timing behaviour. This analogy exists because the considered slaves are similar (RFieldbus boards; native PROFIBUS boards) and the path between initiator and responder is the same (e.g Master/Slave in 1 domain). Results of the measurements are presented in Table 5.2.

Stream	C_{ack} (bit times)	C^m_{ack} (bit times)
S1	710	1160
S2	2205	1642
S3	2205	1646
S4	930	1120
S5	930	1122
S6	743	932
S7	743	931
S8	1018	1208
S10	5935	5518
S12	3702	3950
S13	8778	7199

Table 5.2: Original results

As it can be easily noticed, most measured results are higher than the computed worst-case figures, which denotes an abnormal situation. The reason for this is presented next.

Stream 1 (S1)

This Sound Slave Application DP-stream between PC1(M) and PC6(S) occurs in a Wired Communication Domain (Figure 5.11).

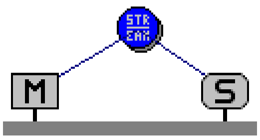


Figure 5.11: PC1-PC6

Theoretic (worst-case)

The worst-case values, calculated by the SPA, are the following:

$$C_{Lreq} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$
$$C_{Lresp} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$
$$T_{st} = 50\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 75 \text{ bit times}$$
$$C_{ack} = 121 + 75 + 121 + 393 = 710 \text{ bit times} = 473{,}3\ \mu s$$

Figure 5.12 depicts the worst case situation between PC and PC in the wired domain.

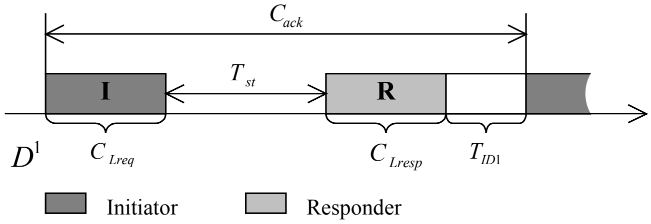


Figure 5.12: Worst-case PC-PC/PLC/MM/ET Wired Domain

Experimental results

These are the experimental results of S1:

$$C^m_{Lreq} = 121 \text{ bit times } \; C^m_{Lresp} = 121 \text{ bit times}$$

$$C^m_{ack} = 1160 \text{ bit times} > C_{ack} = 710 \text{ bit times}$$

$$C^m_{ack\_withoutidletime} = 510 \text{ bit times}$$

$$T^m_{ID1} = C^m_{ack} - C^m_{ack\_withoutidletime} = 1160 - 510 = 650 \text{ bit times}$$

$$> T_{ID1} \Rightarrow \text{delay of } 650 - 393 = 257 \text{ bit times}$$

$$T^m_{st} = 270 \text{ bit times} > T_{st} = 75 \text{ bit times}$$

Figure 5.13 depicts the experimental results between PC and PC in the wired domain.

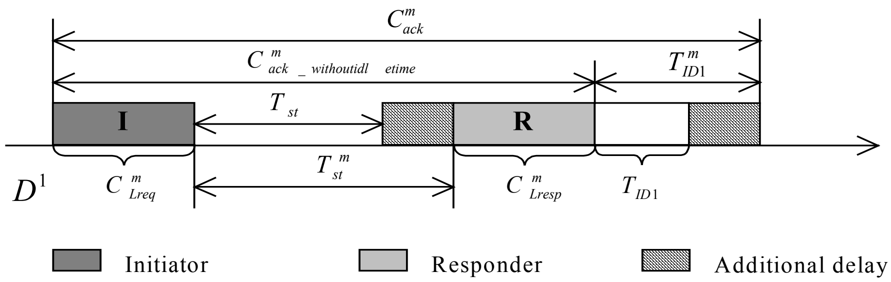


Figure 5.13: Experimental PC-PC Wired Domain

The real PDU duration of both request and response, corresponds with the theoretical duration. The system turnaround time T^m_{st} (Slave PC6) - here equal to responder's turnaround time (one Communication Domain)- is much higher than the initially assumed maximum turnaround time T_{st} of 75 bit times, due to an unexpected delay inserted by the responder.

The initiator (master PC1) also inserts an extra inactivity time of 257 bit times on top of the idle time T_{ID1}. The implementation or the RFieldbus-boards (in all PCs) are the cause of these extra delays.

Analogous results were obtained for S12 (Table 5.2).

Stream 2 (S2) and Stream 3 (S3)

The Robuter Slave Connection DP-stream between PC1 and mobile PC3 (S2) and the RC2 Slave Connection DP-stream between PC1 and static PC4 (S3) cross 3 different Communication Domains (Wired-Wireless-Wired) (Figure 5.14). The PDUs have to be relayed by two ISs.

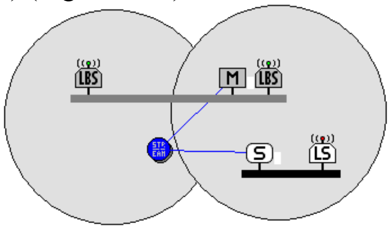


Figure 5.14 : PC1-PC3/PC4

Theoretic (worst-case)
The worst-case values, calculated by the SPA, are the following:

$$C_{Lreq} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$
$$C_{Lresp} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$

$$T_{st} = 1046{,}7\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 1570 \text{ bit times}$$

$$C_{ack} = 121 + 1570 + 121 + 393 = 2205 \text{ bit times} = 1470\ \mu s$$

Figure 5.15 depicts the worst case situation between PC and PC for the message stream that is transmitted via the wired (D^1), wireless (D^2), and wired (D^3), domain.

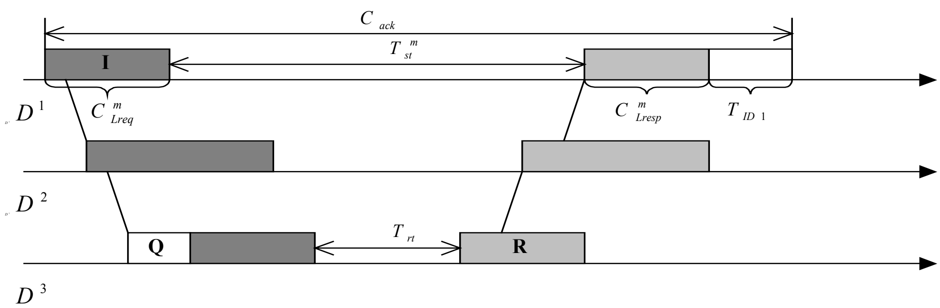


Figure 5.15: Worst-case PC-PC Wired-Wireless-Wired

Experimental results
The experimental results for S2 were as follows :

$$C^m_{Lreq} = 121 \text{ bit times } \; C^m_{Lresp} = 121 \text{ bit times}$$
$$C^m_{ack} = 1642 \text{ bit times} < C_{ack} = 2205 \text{ bit times}$$
$$C^m_{ack\_withoutidletime} = 995 \text{ bit times}$$
$$T^m_{ID1} = C^m_{ack} - C^m_{ack\_withoutidletime} = 1642 - 995 = 647 \text{ bit times}$$
$$> T_{ID1} \Rightarrow \text{delay of } 647 - 393 = 254 \text{ bit times}$$
$$T^m_{st} = 754 \text{ bit times} < T_{st} = 1570 \text{ bit times}$$

Figure 5.16 depicts the experimental results.

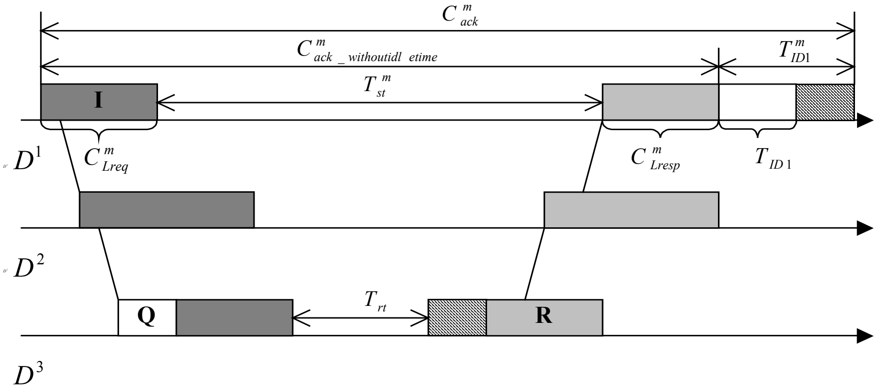


Figure 5.16 Experimental PC-PC Wired-Wireless-Wired

The real PDU length of both request and response, correspond with the theoretical length. T^{m}_{st}=754 fits in to the worst-case value T_{st}=1570.
The real PDU length of both request and response, correspond with the theoretical length. T^{m}_{st}=754 fits in to the worst-case value T_{st}=1570. The measured message duration C^{m}_{ack} is also smaller than the worst-case value C_{ack}, even with the initiator (PC1) inserting again an additional inactivity time of 254 bit times besides T_{ID1}. Probably the responder also experiences this delay, but this is not be possible to notice because the SPA takes the Q into account for the computation of both T_{st} and C_{ack}.
The same conclusions can be drawn for S10 and S13 (Table 5.2).

Stream 4 (S4) and stream 5 (S5)

The DP-streams between PC1 and the PROFIBUS MicroMasters: MM1 (S4) or MM2 (S5), occur in a Wired Communication Domain (Figure 5.17). The interfaces in the MMs are native PROFIBUS boards.

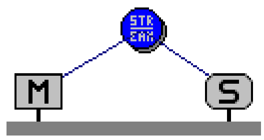


Figure 5.17 : PC1-MM1/MM2

Theoretic (worst-case)
The worst-case values, calculated by the SPA, are the following:
Figure 5.12 depicts the worst case situation between PC and MM in the wired domain.

$$C_{Lreq} = 11\text{bits x } 21\text{chars} = 231 \text{ bit times}$$

$$C_{Lresp} = 11\text{bits x } 21\text{chars} = 231 \text{ bit times}$$

$$T_{st} = 50\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 75 \text{ bit times}$$

$$C_{ack} = 231 + 75 + 231 + 393 = 930 \text{ bit times} = 620\ \mu s$$

Experimental results

The experimental results for S4 were as follows :

$$C^m_{Lreq} = 231 \text{ bit times} \quad C^m_{Lresp} = 231 \text{ bit times}$$

$$C^m_{ack} = 1120 \text{ bit times} > C_{ack} = 930 \text{ bit times}$$

$$C^m_{ack\_withoutidletime} = 474 \text{ bit times}$$

$$T^m_{ID1} = C^m_{ack} - C^m_{ack\_withoutidletime} = 1120 – 474 = 646 \text{ bit times}$$

$$> T_{ID1} \Rightarrow \text{delay of } 646 – 393 = 253 \text{ bit times}$$

$$T^m_{st} = 12 \text{ bit times} < T_{st} = 75 \text{ bit times}$$

The experimental results are shown Figure 5.18.

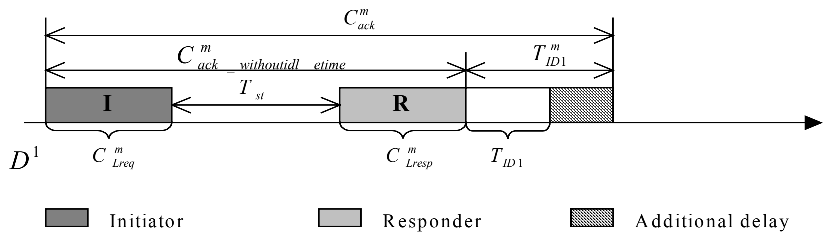


Figure 5.18: Experimental PC-PLC/MM/ET Wired Domain

The real PDU length of both request and response, correspond to the theoretical length.
The measured system turnaround time T^m_{st} (equal to responder's turnaround time) fits in the worst-case value T_{st} . Here we can conclude that the PROFIBUS native board (in the MM) does not insert a delay, but the initiator (PC1) inserts still an additional delay of 253 bit times (when the idle time T_{ID1} is issued). That is why the measured message duration C^m_{ack} is higher then the worst-case value C_{ack}. We notice that this strange delay is apparently only inserted by the RFieldbus boards in the PCs and not by the native PROFIBUS boards in the MMs.
The same conclusions can be drawn for S6,S7 and S8 (Table 5.2).

Token passing

This is the token-passing 'stream' between Master 1 (PC1) and the Mobility Master (PC2)(Figure 5.19).

Figure 5.19: Token passing PC1-PC2

Theoretic

The worst-case values, calculated by the SPA, can be found below. The same $T_{st\_token} = T_{ID1}$ is set in both masters.

$$C_{Ltoken} = 11 \text{ bits x 3 chars} = 33 \text{ bit times}$$

$$T_{st\_token} = T_{ID1} = 262\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 393 \text{ bit times}$$

Figure 5.20 depicts the token passing between masters in theory.

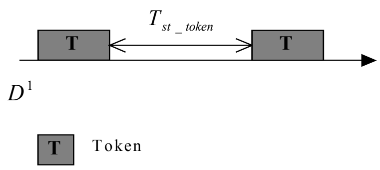


Figure 5.20: Worst-case Master-Master Token Passing

<u>Experimental results</u>
These are the experimental results:

$$C^m_{Ltoken} = 33 \text{ bit times}$$

$$T^m_{st\_token} = T^m_{ID1} = 590 \text{ bit times} > T_{st\_token} = T_{ID1} = 393 \text{ bit times}$$

=> delay of 590 – 393 = 197 bit times

The token passing, as it was measured, is depicted in Figure 5.21.

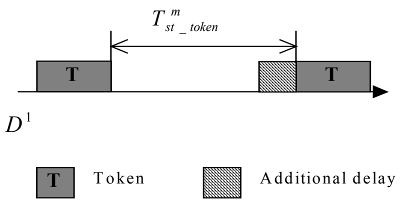


Figure 5.21: Experimental Master-Master Token Passing

The Masters (both with RFieldbus board interfaces) introduce a delay of around 200 bit times, when passing the token to one another. This delay is smaller then the delay introduced in message transactions. Even when we change the *min* T_{SDR}, determining $T_{ID1} = T_{st\_token}$, in the configuration of the MAF to 0 bit times or to 1000 bit times the delay stays around 200 bit times. In case of the message transactions the delay stays around 250 bit times, when changing the min T_{SDR}.

Slot Time parameter T_{SL}

This parameter is essential for guaranteeing the real-time behaviour of the network.

<u>Theoretic</u>
This is the value computed by the SPA:

$$T_{sl} = 2775 \text{ bit times} = 1850\mu s \times 1{,}5\text{Mbit/sec (fig 5.5)}$$

Figure 5.22 shows the expected T_{SL}.

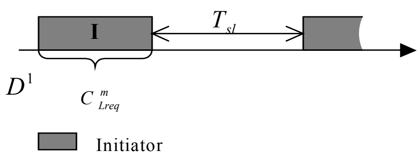


Figure 5.22: Theoretic T_{SL}

<u>Experimental results</u>
This is the experimental result:

$$T^m_{SL} = 2925 \text{ to } 2975 \text{ bit times}$$

The experimental situation is depicted in Figure 5.23.

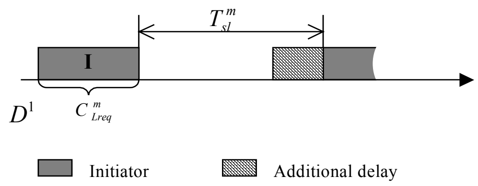


Figure 5.23: Experimental T^m_{SL}

The Master introduces a additional delay of around 150 to 200 bit times to the Slot Time, this means that the timeout occurs later. This delay is also smaller than the delay introduced in message transactions. Even when we change the T_{SL}, in the configuration of the MAF to 3000 bit times or to 5000 bit times the additional delay stays around 150 to 200 bit times.

5.3.2 Coping with the additional delays

As previously described, an unexpected timing behaviour was noticed, when matching the theoretical with the experimental results A thorough study led to the conclusion that the RFieldbus boards introduce additional delays, both when acting as responder and as initiator. In the case of acting as a responder, this delay has a considerable influence on the T_{rt} and T_{st} of every message stream. In case of acting as an initiator, this has consequences for T_{ID1} and C_{ack}. Therefore these delays must be considered in the theoretical computation.
For the study of the behaviour of the responder it was originally assumed that T_{rt_min} = 15 bit times and T_{rt_max} = 75 bit times where realistic values, because native PROFIBUS slave boards (e.g. PLC) usually have a short T_{rt}. After several measurements, it was concluded that the RFieldbus boards have a turnaround time that can be in the order of 370 bit times. These additional delays must be taken into account, by changing the T_{rt_min} and the T_{rt_max} parameters in the SPA, according to the real $T_{rt\ min}$ = 12 bit times and T_{rt_max} = 370 bit times that where measured.
In the case where the RFieldbus board is used for a master station, we have concluded that the RFieldbus board in the master (initiator) inserts an additional inactivity time of around 250 (=562-393) bit times, which happens for every message transaction. In the case of the token passing between the two masters, this is extra delay is only 200 bit times. The master station also inserts T_{SL} + (150 to 200) bit times instead of the T_{SL} =2775 bit times.
The value of 393 bit times for T_{ID1}, calculated by the SPA, was indirectly set in the configuration file of the master by manipulating the min T_{SDR} parameter. This problem can be partially solved by putting indirectly T_{ID1}=146 (=396–250) bit times into the configuration file of the master. Unfortunately, this change will still not give the correct value for T_{ID1} in case of the token passing. Note that the new value calculated by the SPA for $T_{ID\ 1}$= 396 bit times in stead of 393 bit times because T_{rt_min} influences this value and is changed from 15 to 12 bit times. Similarly, the new value for T_{SL}= 2967 bit times will be set as 2767 (= 2967-200) bit times.
Note that we also took the PDU lengths into consideration, especially for the IP-transactions this had to be changed. The size of the request PhL PDU for S10, S11, S12 and S13 was not 6 bytes but 14 bytes measured. S14 had a measured request PhL PDU of 126 bytes and an equal answer.

5.4 Analytical vs. experimental results

5.4.1 Detailed results

In the configuration files of the master ESs the following parameters have been changed, according to the new values computed by the SPA:

T_{ID1} = 396-250 = 146 bit times
T_{SL}= 2966-200 = 2766 bit times

The new worst-case scenarios are compared against the new measurements. We will describe in detail the same streams as in section 5.3.1. Table 5.3 (at the end of this section) shows the results for S1-S13.

Stream 1 (S1)

This stream is depicted in Figure 5.24.

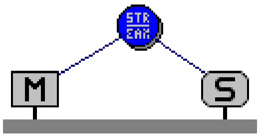


Figure 5.24: PC1-PC6

<u>Theoretic (worst-case)</u>

The worst-case values, calculated by the SPA, are the following:

$$C_{Lreq} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$
$$C_{Lresp} = 11\text{bits x } 11\text{chars} = 121 \text{ bit times}$$
$$T_{st} = 247\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 370 \text{ bit times}$$
$$C_{ack} = 121 + 370 + 121 + 396 = 1008 \text{ bit times} = 672{,}3 \ \mu s$$

Figure 5.25 depicts the worst case situation between PC and PC in the wired domain.

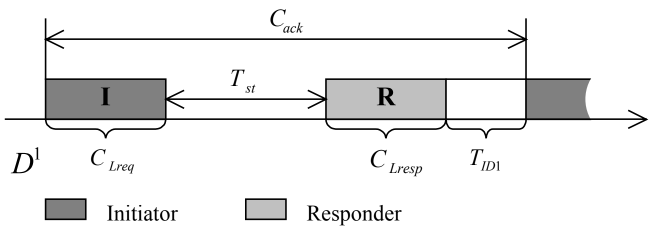


Figure 5.25: Worst-case PC-PC/PLC/MM/ET Wired Domain

<u>Experimental result</u>

These are the experimental results of S1:

$$C^m_{Lreq} = 121 \text{ bit times} \quad C^m_{Lresp} = 121 \text{ bit times}$$
$$C^m_{ack} = 916 \text{ bit times} < C_{ack} = 1008 \text{ bit times}$$
$$C^m_{ack\_withoutidletime} = 519 \text{ bit times}$$
$$T^m_{ID1} = C^m_{ack} - C^m_{ack\_withoutidletime} = 916 - 519 = 397 \text{ bit times}$$
$$= T_{ID1} \Rightarrow \text{no more delay detected}$$
$$T^m_{st} = 271 \text{ bit times} < T_{st} = 370 \text{ bit times}$$

Figure 5.26 depicts the experimental results between PC and PC in the wired domain.

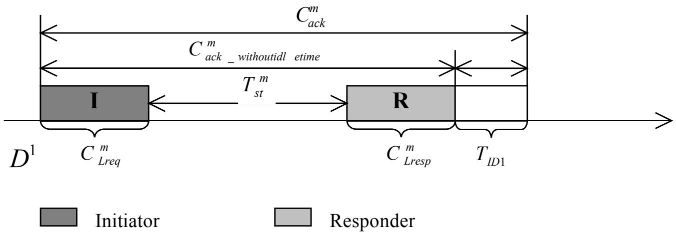


Figure 5.26: Experimental PC-PC Wired Domain

The measured system turnaround time T^m_{st} is now shorter than the worst-case value. This is what we expected, since we adapted the maximum t_{rt} in the from 75 to 370 bit times. This must be correct for every message stream.
This time, the initiator inserts the correct T_{ID1}, because we forced the master to do this (Section 5.3.2).
No more irregularities were detected here.
We can conclude the same for S9 and S12.

Stream 2 (S2) and Stream 3 (S3)
This stream is depicted in Figure 5.27.

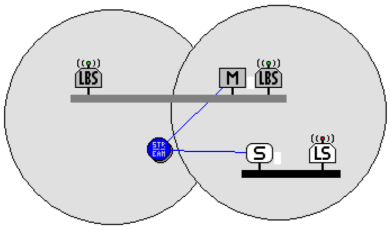


Figure 5.27 : PC1-PC3/PC4

Theoretic (worst-case)
The worst-case values, calculated by the SPA, are the following:

$$C_{Lreq} = 11\text{bits} \times 11\text{chars} = 121 \text{ bit times}$$
$$C_{Lresp} = 11\text{bits} \times 11\text{chars} = 121 \text{ bit times}$$

$$T_{st} = 1243{,}7\mu s \times 1{,}5 \text{ Mbit/s} = 1866 \text{ bit times}$$

$$C_{ack} = 121 + 1866 + 121 + 396 = 2504 \text{ bit times} = 1669 \ \mu s$$

Figure 5.28 depicts the worst case situation between PC and PC for the message stream that is transmitted via the wired (D^1), wireless (D^2), and wired (D^3), domain.

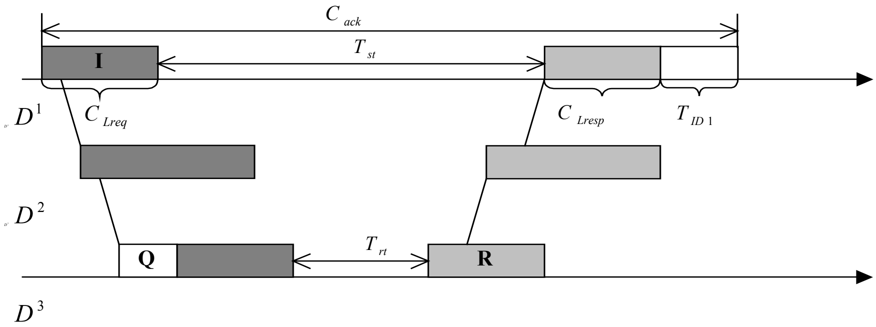


Figure 5.28: Worst-case PC-PC Wired-Wireless-Wired

Experimental results

The experimental results for S2 were as follows :

$$C^m_{Lreq} = 121 \text{ bit times } \; C^m_{Lresp} = 121 \text{ bit times}$$
$$C^m_{ack} = 1400 \text{ bit times} < C_{ack} = 2504 \text{ bit times}$$
$$C^m_{ack\_withoutidletime} = 1004 \text{ bit times}$$
$$T^m_{ID1} = C^m_{ack} - C^m_{ack\_withoutidletime} = 1400 - 1004 = 396 \text{ bit times}$$
$$= T_{ID1} \Rightarrow \text{no more delay detected}$$
$$T^m_{st} = 758 \text{ bit times} < T_{st} = 1866 \text{ bit times}$$

Figure 5.29 depicts the experimental results.

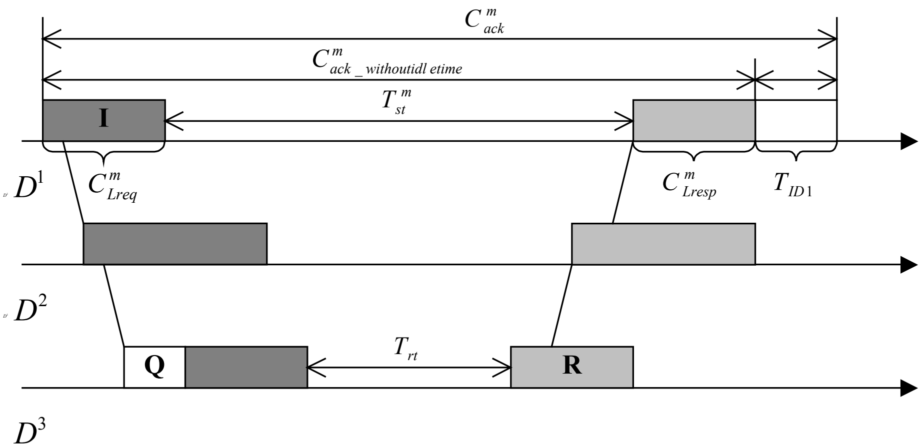


Figure 5.29 Experimental PC-PC Wired-Wireless-Wired

The responder's (PC3 and PC4) answer (T^{m}_{st}) fit in the worst-case value T_{st} and the measured message duration C^{m}_{ack} is also smaller than the worst-case value C_{ack}. The correct T_{ID1} of 396 bit times was inserted.
The same conclusions can be drawn for S10.

Stream 4 (S4) and stream 5 (S5)
This stream is depicted in Figure 5.30.

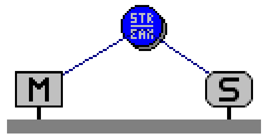


Figure 5.30 : PC1-MM1/MM2

Theoretic (worst-case)
The worst-case values, calculated by the SPA, are the following:

$$C_{Lreq} = 11\text{bits x } 21\text{chars} = 231 \text{ bit times}$$

$$C_{Lresp} = 11\text{bits x } 21\text{chars} = 231 \text{ bit times}$$

$$T_{st} = 247\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 370 \text{ bit times}$$

$$C_{ack} = 231 + 370 + 231 + 396 = 1228 \text{ bit times} = 819\ \mu s$$

Figure 5.12 depicts the worst case situation between PC and PC in the wired domain.

Experimental results
The experimental results for S4 were as follows :

$$C^{m}_{Lreq} = 231 \text{ bit times} \quad C^{m}_{Lresp} = 231 \text{ bit times}$$

$$C^{m}_{ack} = 875 \text{ bit times} < C_{ack} = 1228 \text{ bit times}$$

$$C^{m}_{ack\_withoutidletime} = 478\text{bit times}$$

$$T^{m}_{ID1} = C^{m}_{ack} - C^{m}_{ack\_withoutidletime} = 875 - 478 = 397 \text{ bit times}$$

$$= T_{ID1} \Rightarrow \text{no additional delay}$$

$$T^{m}_{st} = 12 \text{ bit times} < T_{st} = 370 \text{ bit times}$$

The experimental results are shown Figure 5.31.

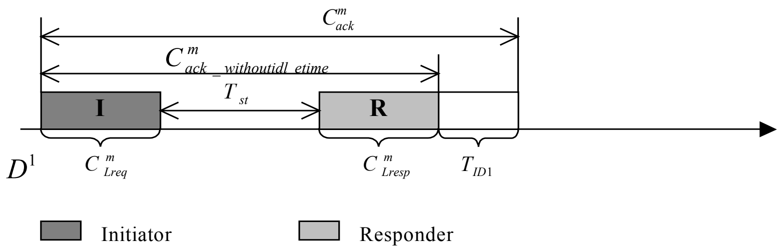


Figure 5.31: Experimental PC-PLC/MM/ET Wired Domain

The analytical match the experimental results.

Token passing

The token-passing ‘stream’ between Master 1 (PC1) and the Mobility Master (PC2)-Figure 5.32.

Figure 5.32: Token passing PC1-PC2

<u>Theoretic (worst-case)</u>
The worst-case values, calculated by the SPA, can be found below.

$$C_{Ltoken} = 11 \text{ bits x } 3 \text{ chars} = 33 \text{ bit times}$$
$$T_{st\_token} = T_{ID1} = 264\mu s \text{ x } 1{,}5 \text{ Mbit/s} = 396 \text{ bit times}$$

Figure 5.33 depicts the token passing between masters in theory.

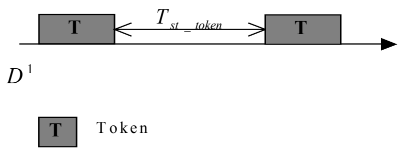


Figure 5.33: Worst-case Master-Master Token Passing

<u>Experimental results</u>
These are the experimental results:

$$C^{m}_{Ltoken} = 33 \text{ bit times}$$
$$T^{m}_{st\_token} = T^{m}_{ID1} = 345 \text{ bit times} < T_{st\_token} = T_{ID1} = 396$$

The token passing, as it was measured, is depicted in Figure 5.34.

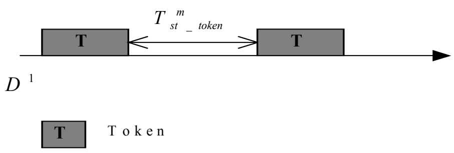


Figure 5.34: Experimental Master-Master Token Passing

We chose to compensate the delay inserted by the initiator on top of T_{ID1}, by putting T_{ID1}- 250 bit times into the configuration file of the master. We also know that the inserted T_{ID1} during the token passing was only 200 bit times higher than. The result is now, that $T^{m}_{st\_token}$ is 50 bit times too small.

***Slot Time parameter* T_{SL}**

<u>Theoretic</u>
This is the value computed by the SPA:

$$T_{sl} = 2966 \text{ bit times} = 1977{,}7\mu s \text{ x } 1{,}5\text{Mbit/sec}$$

Figure 5.35 shows the expected T_{SL}.

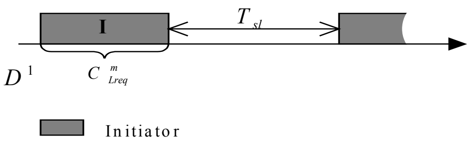


Figure 5.35: Worst-case T_{SL}

Experimental results
This is the experimental result:

$$T_{SL}^{m} = 2915 \text{ to } 2965 \text{ bit times}$$

The experimental situation is depicted in Figure 5.36.

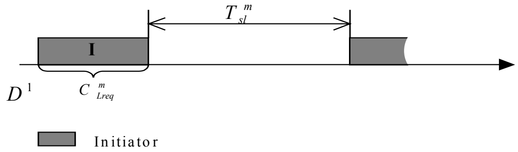


Figure 5.36: Experimental T_{SL}^{m}

The problem of the extra delay of 150 to 200 bit times inserted on top of T_{SL} is solved by putting T_{SL}-200 into the configuration files of the masters.

5.4.2 Considerations about the experimental results

After adapting the parameters in the configuration files of the masters, we can conclude that the measurements match the theoretical predictions (Table 5.3).

The table also provides a column with the pessimism introduced by the SPA, which is calculated as follows :

$$\left[\left(C_{ack} / C_{ack}^{m}\right) - 1\right] \times 100\%$$

.

Stream	C_{ack} (bit times)	C_{ack}^{m} (bit times)	Pessimism (%)
S1	1008	916	10
S2	2504	1392	80
S3	2504	1400	79
S4	1228	879	40
S5	1228	876	40
S6	1042	682	53
S7	1042	683	53
S8	1317	966	36
S10	5490	4609	37
S12	3726	3044	22
S13	5490	4973	10

Table 5.3 Adapted results

Lets take a closer look at some different values of the pessimism. For a stream with T_{st}=T_{rt_max}=370 bit times, where the initiator and responder are in the same domain, the pessimism must be 0%. In our topology only stream 13 has this T_{st}≈T_{rt_max}=368 bit times, but the pessimism is 10%. In stream 1 it this value is low (10%), because this stream has a high T_{st}=270 bit times, with $C_{ack} \approx C^{m}_{ack} + T_{rt\_max} - T_{st} \approx$ 916+370-270 ≈1008 bit times. In case of stream 8, the T_{st}= $T_{rt\ min}$ =12 bit times is very low, with C_{ack} ≈ 966+370-12≈1317 bit times. This results in a higher pessimism of 36%.
The percentages for message streams from which initiator and responder are situated in different domains are often more pessimistic than those from which they are in the same domain. This is because the Q is taken into account in the worst-case computation and real queuing delay in the case study depends on the amount of traffic on the network. For this reason, the pessimism for stream 2 and 3 is around 80%.

6 Monitoring of the IS devices

6.1 Introduction

In RFieldbus, the intermediate systems used to relay the network traffic between wired and wireless PROFIBUS operate at the Physical Layer level. In spite of their repeating operation, there are several important parameters that must be monitored. For this purpose, we have extended an already existing local monitoring application to remote and multiple IS monitoring via TCP/IP.

6.1.1 Local/Remote monitoring of IS

A complex hybrid network can contain several ISs. The Local Monitoring Application (LMA) can run on each PC connected to an IS. Then, it is possible to monitor this traffic on remote PC’s with the Remote Monitoring Application (RMA), that can be situated anywhere in the Intranet/Internet. For this purpose, a client-server approach can be used. If the PC connected to the IS has no access to the Internet/Intranet, the exchange of data between the LMA and the RMA takes place via the existing hybrid network. This causes an extra stream on the network. The different possibilities are shown in Figure 6.1.

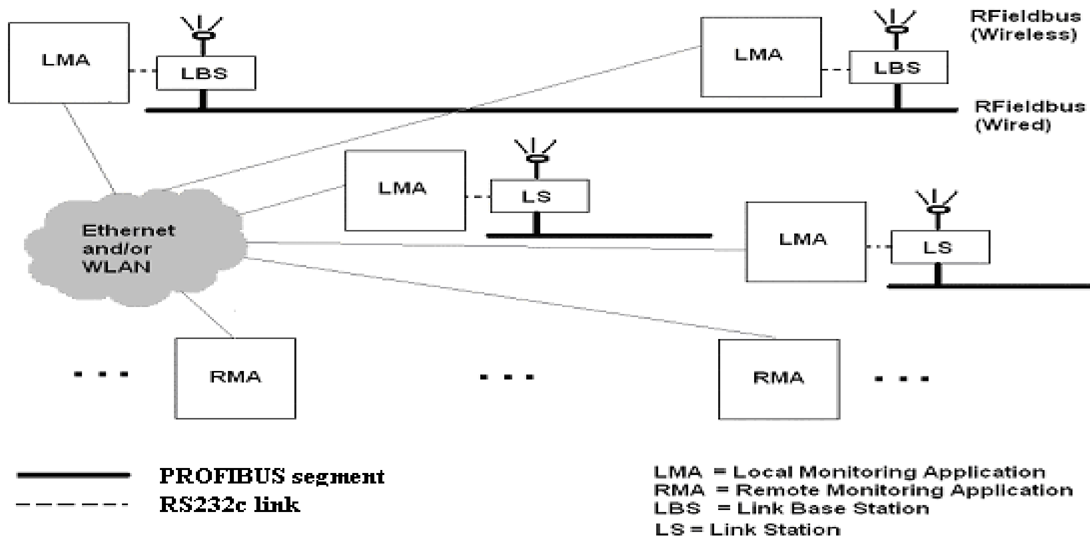


Figure 6.1: Global concept

6.1.2 Case study

Assuming the Manufacturing Automation Field Trial, we developed an application that permits to monitor the traffic that passes through the two Link Base Stations on the PROFIBUS backbone on a remote computer via TCP/IP and Switched Ethernet. We have adopted a client-server approach, where a Local Monitoring Application (LMA) runs on the two PC’s (servers) connected to the two Link Base Stations (LBS) and collects the local traffic information via a COM port. A Remote Monitoring Application (RMA) can connect to the servers to monitor the data on a remote computer (client). Several clients on the Intranet/Internet can access the servers at the same time . This concept is visualised in Figure 6.2.

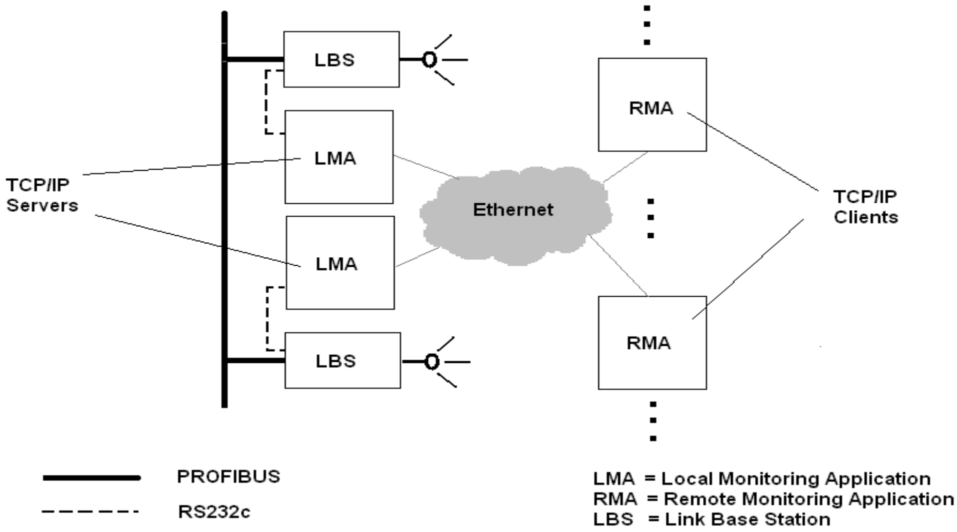


Figure 6.2: Case study

6.2 The Monitoring Application

6.2.1 The Local Monitoring Application

The Local Monitoring Application permits to set the traffic timing measurements sampling rate, to control the IS state machine (e.g. Reset) and to set the moving range of the mobile IS. Figure 6.3 shows this part of the HMI, i.e. the Control panel.

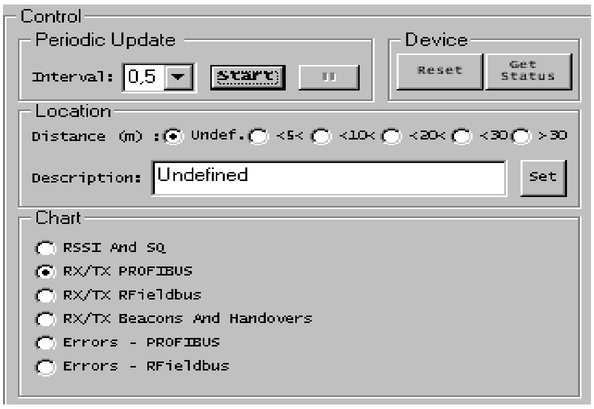


Figure 6.3: Control panel

It is also possible to choose the chart output to the screen (e.g. RX/TX-PROFIBUS). Figure 6.4 shows a snapshot of the chart that monitors the sent and received data PDUs on the PROFIBUS side of the LBS.

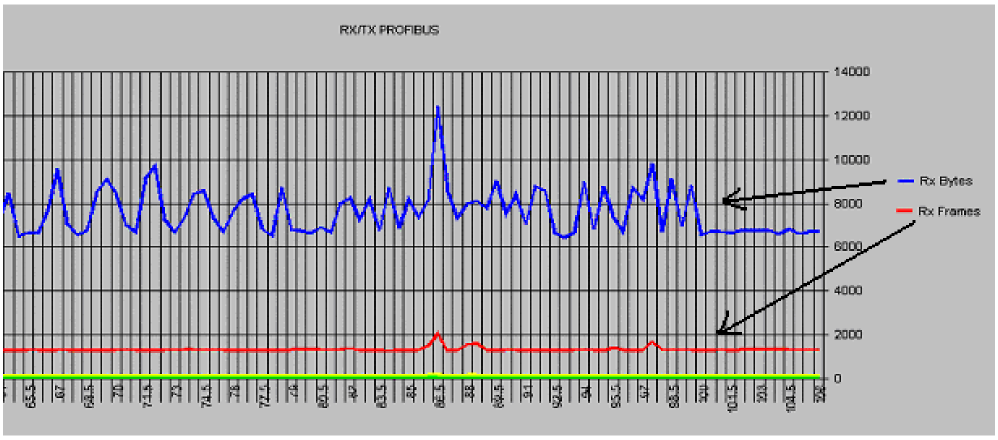


Figure 6.4: Monitor

The chart above depicts the data PDUs and bytes, that were received on the PROFIBUS side of the LBS, are depicted. The horizontal axis represents the time the vertical axis represents the signal power.
Besides the charts, the Local Monitoring application also monitors several parameters concerning both the PROFIBUS and the RFieldbus sides of the ISs. This part of the HMI is illustrated in Figure 6.5. A more detailed description of the parameters and the functionalities of the Control panel is given in ANNEX A.

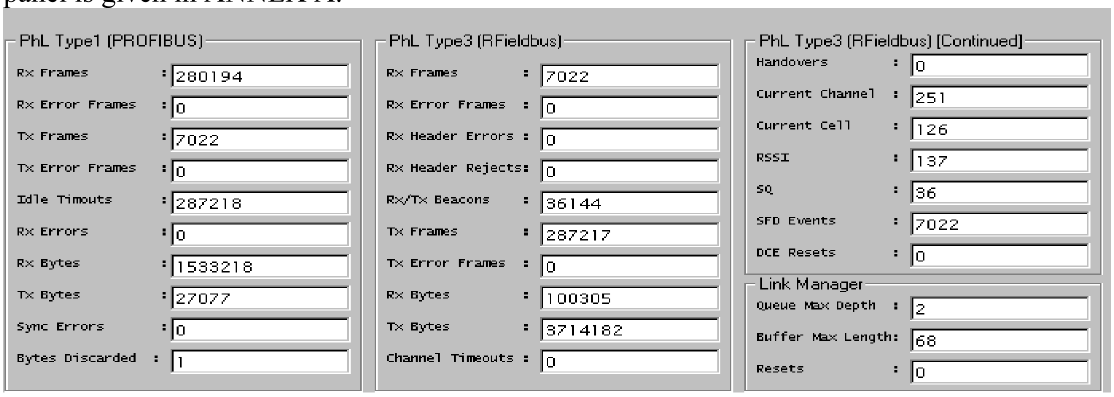


Figure 6.5: Parameters

For example the number of RX (Received) PDUs on the PROFIBUS side of the LBS is 280194 and the number of TX (Transmitted) PDUs to the same side is 7022.

The Local Monitoring Application also creates logfiles with all these parameters. Figure 6.6 shows a snapshot of both files. 'WDMonitor.log' registrates traffic information in a not so structured, briefly way with periodic time indications (Figure 6.6.a). 'WDOutput.log' registrates traffic information in a structured, user-friendly way, without periodic time indications (Figure 6.6.b).

```
Session Date:21-05-2003

Period = 500ms

Nr,Time,Rx Frames,Rx Error Frames,Tx Frames,Tx Error Frames,Idle Timouts,Rx Errors,Rx Bytes,Tx
Bytes,Sync Errors,Bytes Discarded,Rx Frames,Rx Error Frames,Rx Header Errors,Rx Header Rejects,Rx/Tx
Beacons,Tx Frames,Tx Error Frames,Rx Bytes,Tx Bytes,Channel Timeouts,Handovers,Current
Channel,Current Cell,RSSI,SQ,SFD Events,DCE Resets,Queue Max Depth,Buffer Max
Length,Resets,Distance,Location Description
1,11:52:26,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,275557,0,251,123,29,0,0,0,0,0,0,Undefined,Undefined
2,11:52:27,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,275561,0,251,123,30,0,0,0,0,0,0,Undefined,Undefined
```

Figure 6.6.a: WDMonitor.log

```
Session Date:21-05-2003
Period = 500ms
-------------------- 1 --------------------

  PhL Type1 (PROFIBUS)
  Rx Frames            =    0
  Rx Error Frames      =    0
  Tx Frames            =    0
  Tx Error Frames      =    0
  Idle Timouts         =    1
  Rx Errors            =    0
```

Figure 6.6.b: WDOutput.log

6.2.2 The Remote Monitoring Application

The RMA is similar to the LMA, therefore it also monitors the parameters and presents the different charts. Switching between the monitoring of LBS1 and LBS2 is easily done by a mouse click. There is also the possibility to clear the charts and to connect or disconnect to the LMA. Because the RMA only monitors the data that the LMA collects, it is not possible to control the measurements, device, ... from the client side.

6.3 Implementation aspects

6.3.1 Implementation approach

The Remote Monitoring Application was developed in Visual Basic 6, using the WinSock control. This is a tool invisible to the user, that provides easy access to TCP and UDP network services by the use of sockets. By setting properties and invoking methods of the control, we can easily connect to a remote machine and exchange data in both directions.
Our next consideration was whether to use the TCP or UDP protocol. The most significant difference between these two protocols is their connection state:

- The TCP protocol control is a connection-oriented protocol, and is analogous to a telephone (the user must establish a connection before proceeding). The Transfer Control Protocol permits to create and maintain a connection to a remote computer. Using the connection, both computers can stream data between themselves.
- The UDP protocol is a connectionless protocol, and the transaction between two computers is like passing a note (a message is sent from one computer to another, but there is no explicit connection between the two).

The nature of the application determines the protocol that we want to use. Since we are working with an extremely large amount of data, that is send continously from the clients to the server, we have opted for the TCP protocol.

6.3.2 Basics to set up connection with sockets via TCP

To make the client- server connection via sockets we used TCP basic functions.
To create the client application we must know the server computer's name or IP address (**RemoteHost** property), as well as the port (**RemotePort** property) on which it will be "listening." Then we invoke the **Connect** method.
To create the server application, we set a port (**LocalPort** property) on which the server socket has to listen, and invoke the **Listen** method. When the client computer requests a connection, the ConnectionRequest event will occur. To complete the connection, we invoke the **Accept** method within the ConnectionRequest event.

Once a connection has been made, client and server computer can send and receive data. To send data, the **SendData** method is used. Whenever data is received, the DataArrival event occurs. We invoke the **GetData** method within the DataArrival event to retrieve the data.

6.3.3 Flow Chart of the algorithm

Figure 6.7 shows the algorithm used in the Local Monitoring Application that handles connection requests and at the same time gets the data from the COM port, makes the logfiles and monitors the data if the 'start' condition is true. Next, the LMA sends the collected information to all the RMA that are connected.

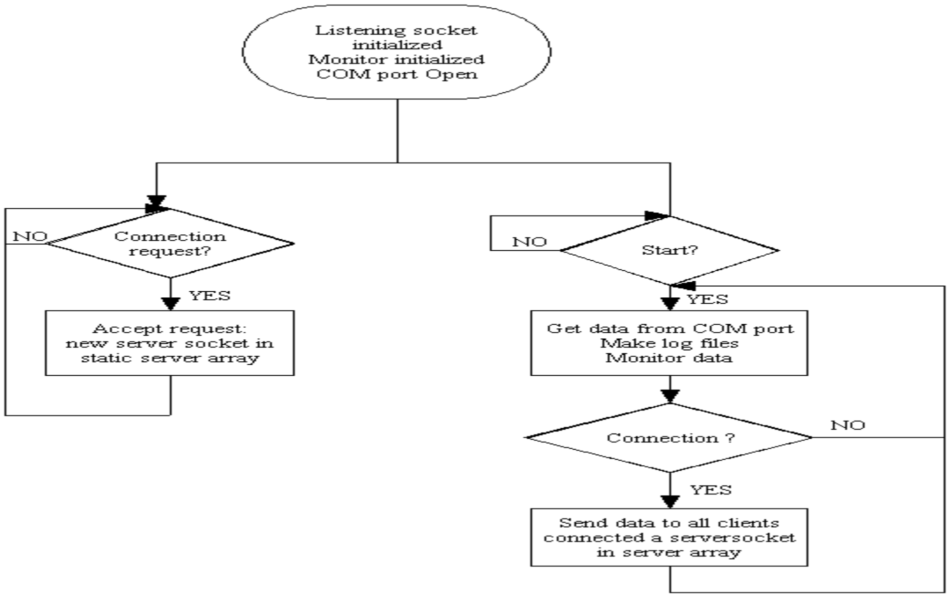
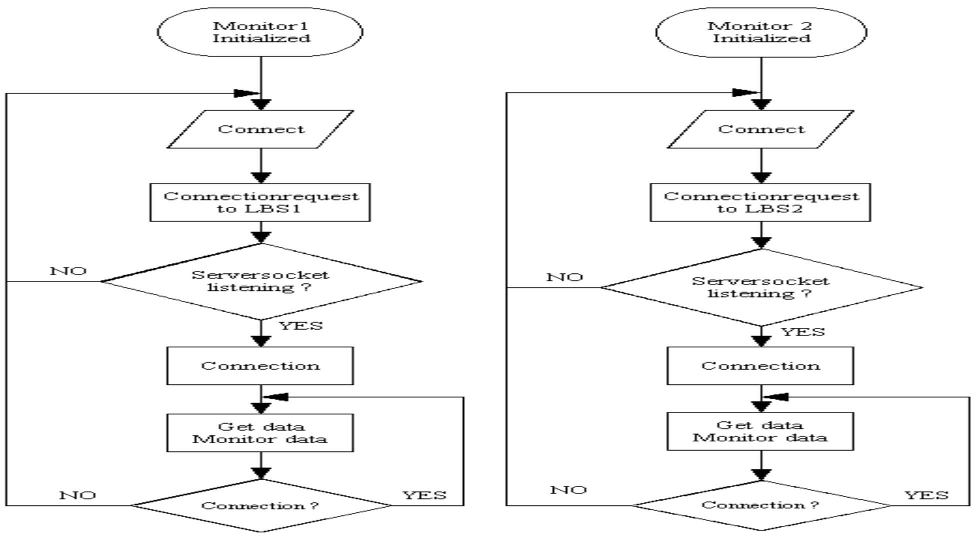


Figure 6.7.: Local Monitoring Application

Figure 6.8 shows the algorithm used in the Remote Monitoring Application to make a connection and then collect the data. The application runs two monitors at the same time i.e. from LBS1 and LBS2. When the 'connection' button is pushed, a connection request is sent to the LMA. If the connection is established, the corresponding monitor will show the data.



.

Fig 6.8: Remote Monitoring Application

6.3.4 Scalability of the application

This client-server application sets up two socket connections between two servers (LBS1 and LBS2) and the client (remote computer), but is easily extendable to monitor more intermediate systems (IS). This can be realised by adding an extra form on the client side for each extra server. Monitoring a IS which is situated on the wireless part is possible, if we use the existing hybrid network to set up the connection. To avoid this extra traffic on the RFieldbus network, we used an Ethernet network.

7 Conclusion

The realization of our final project in Porto has been a great and unique experience. We had the opportunity to integrate ourselves into the IPP HURRAY! Research Group [9], full of motivated and helpful people. We learned how to communicate and work together with different people.
Besides this, we dug into the scientific work of the team [3], learned about the architectural aspects of RFieldbus[1,2] and made ourselves comfortable to understand and experiment with the Manufacturing Automation Field Trial [5,6].
After integrating ourselves in this framework, we had enough knowledge to fulfill our task in the team. Lots of measurements concerning the timing behavior of the message streams in the pilot, lead us to conclude that the RFieldbus boards inserted additional unexpected delays, that were not taken into consideration in the theoretical analysis. Further research work and contact with the company that produced this boards, confirmed what we noticed.
Knowing this, we adapted the input to the System Planning Application. We changed the configuration files of the masters in the system, according to the new valid output parameters and repeated all measurements. Finally, it was possible to match the theoretical with the new experimental results and to draw some relevant conclusions.
On the open day of the MAF we also had contact with the industrial part of the RFieldbus project. As analyzers of the traffic, we gave a demonstration in the RFieldbus workshop (ANNEX B).
Additionaly, we cooperated in a paper published in the 2nd International Workshop on Real-time LANs in the Internet Age (http://www.hurray.isep.ipp.pt/rtlia2003/) . The paper is added in ANNEX C.

REFERENCES

[1] M. Alves; B. Batista; L. Ferreira; G. Hammer; R. Heidel; G. Kalivas;A. Kalogeras; V. Kapsalis; C. Karavasilis; S. Koubias; A. Koukourgiannis;F. Pacheo; G. Papadopoulos; L.M. Pinho; H. Pipinis; N. Rebakos;P. Speckmeier; E. Tovar and F. Vasques (2000c) Requirements for the R-FIELDBUS system. Deliverable D 1.1. RFieldbus project IST–1999-11316, May 2000

[2] M. Alves; T. Bangemann; B. Batista; R. Breithaupt; L. Ferreira; J. Haehniche; G. Hammer;R. Heidel; G. Kalivas; A. Kalogeras; V. Kapsalis; S. Koubias; C. Koulamas; M. Kutschenreuter; S. Monforte; F. Pacheco; K. Roether; P. Speckmeier; E. Tovar and F. Vasques (2000d). General System Architecture of the RFieldbus. Deliverable D 1.3. RFieldbus project IST–1999-11316

[3] M. Alves (2003), Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-Based Networks, Doctoral Thesis, Faculty of Engineering of the University of Porto

[4] Profibus – Open Solutions for the World of Automation (2001) Technical overview. http://www.profibus.com

[5] M. Alves; V. Brandão, E. Tovar; F. Pacheco and M. Pinho (2001) Specification of the Manufacturing Automation Field Trial – Brief Technical Report (HURRAY-BTR-0131)

[6] RFieldbus Manufacturing Field Trial – http://www.hurray.isep.ipp.pt/rfpilot/

[7] Softing GmbH (1998) PROFIBUS Analyzer User Manual

[8] IST (Information Society Technology) project RFieldbus (High performance Wireless Fieldbus in Industrial Multimedia-Related Environment) – http://www.RFieldbus.de

[9] HUgging Real-time and Reliable Architectures for computing sYstems Research Group - http://www.hurray.isep.ipp.pt/

[10] 2nd Intl WORKSHOP ON REAL-TIME LANS IN THE INTERNET AGE Polytechnic Institute of Porto, Portugal, July 1, 2003 - http://www.hurray.isep.ipp.pt/rtlia2003/

[11] E. Tovar, F. Vasques (1999a). Real-time Fieldbus Communications Using PROFIBUS Networks. In IEEE Transactions on Industrial Electronics, Vol. 46, nr.6, pp. 1241-1251.

[12] M. Alves, E. Tovar, F. Vasques, G. Hammer, K. Röther (2002). Real-time Communications over Hybrid Wired/Wireless PROFIBUS-based Networks. Proc. of the 14th Euromicro Conference on Real-Time Systems – ECRTS'02, Vienna, Austria, pp. 142-150

[13] M. Alves, E. Tovar, F. Vasques (2001a). On the Adaptation of Broadcast Transactions in Token-Passing Fieldbus Networks with Heterogeneous Transmission Media. In proc. of the 4th IFAC Conference on Fieldbus Systems and their Applications (FET'01), Nancy, France, pp. 278-284.

[14] ifak system GmbH (1999-2001) _IS Pro Multiboard User manual v 2.2

ANNEX A

PARAMETERS MONITORED BY THE LOCAL IS MONITORING APPLICATION

Periodic Update Panel	
• Interval	Interval between two traffic measurement samples
• Start/Stop	Starts/Stops the traffic measurement Makes connection with server if server is available
• Pause	Pauses the traffic measurement
Device Panel	
• Reset	Resets the device softly
• Get Status	Gets status of the device
Location	Sets the Location of the mobile IS, i.e. over what distance it can move
Chart Panel	
• RSSI And SQ	Monitors the signal quality of the traffic
• RX/TX PROFIBUS	Monitors the sent and received data frames on the PROFIBUS side of the LBS
• RX/TX RFieldbus	Monitors the sent and received data frames on the RFieldbus side of the LBS
• RX/TX Beacons And Handovers	Monitors the sent and received beacons and the handovers
• Errors – PROFIBUS	Monitors errors on PROFIBUS side of the LBS
• Errors – RFieldbus	Monitors errors on RFieldbus side of the LBS
Phl Type1 (PROFIBUS)	
• RX Frames	Number of data frames received from the PROFIBUS side by the LBS
• RX Error Frames	Number of RX frames with an error
• TX Frames	Number of data frames sent to the PROFIBUS side by the LBS
• TX Error Frames	Number of TX Frames with an error
• Idle Time Outs	Number of idle times inserted by the IS
• RX Errors	Number of errors in the RX frames
• RX Bytes	Number of data bytes received from the PROFIBUS side by the LBS
• TX Bytes	Number of data bytes sent to the PROFIBUS side by the LBS
• Sync Errors	Errors in the synchronisation sub-field
• Bytes Discarded	Rejected bytes
PhL Type3 (RFieldbus)	
• RX Frames	Number of data frames received from the RFieldbus side by the LBS
• RX Error Frames	Number of RX frames with an error
• RX Header Errors	Number of RX Headers with an error
• RX Header Rejects	Number of RX Headers that are rejected
• RX/TX Beacons	Number of received and transmitted beacons
• TX Frames	Number of data frames sent to the RFieldbus side by the LBS
• TX Error Frames	Number of TX Frames with an error
• RX Bytes	Number of data bytes received from the RFieldbus side by the LBS
• TX Bytes	Number of data bytes sent to the RFieldbus side by the LBS
• Channel Timeouts	Number of timeouts when there is no traffic detected passing through the IS
• Handovers	Number of handovers
• Current Channel	Channel where the IS is sending/receiving the data
• Current Cell	Current cell the IS is situated in
• RSSI & SQ	Received Signal Strength Indication & Signal Quality: Parameters to decide if a signal can be detected as Carrier Detection (CD) or not
• SFD Events	Starting Frame Delimiter:
• DCE Resets	Number of resets of Data Communication Equipment
The Link Manager*	
• Queue Max Depth	
• Buffer Max Length	
• Resets	

*Meaning unknown

ANNEX B

IPP Hurray! GROUP

http://www.hurray.isep.ipp.pt

isep instituto superior de engenharia do porto

INSTITUTO POLITÉCNICO DO PORTO

16 Maio 2003

dia aberto

RFieldBus

ANNEX C

EUROMICRO

2^{nd} Intl WORKSHOP ON REAL-TIME LANS IN THE INTERNET AGE

Polytechnic Institute of Porto, Portugal, July 1, 2003

http://www.hurray.isep.ipp.pt/rtlia2003/

in conjunction with the

15^{th} Euromicro Intl Conference on Real-Time Systems.

Porto, Portugal. July 2-4, 2003

CALL FOR PAPERS (text)

ANNOUNCEMENT: A template for RTLIA Full papers (PDF) (DOC) is available.

Workshop Chairperson:

Lucia Lo Bello
Dipartimento di Ingegneria Informatica
e delle Telecomunicazioni
University of Catania
Italy
phone: +39-095-7382379
fax:+39-095-338280/+39-095-7382397
Lucia.LoBello@diit.unict.it

Important dates (deadlines):

- Submission of extended abstracts
May 4
- Notification of acceptance
May 26
- Final submission of papers
June 13
- Broadcast of final papers
June 20
- RTLIA Workshop
July 1
- Euromicro Conf. on RTS
July 2-4
- Final version of the paper
July 20

The Euromicro Technical Committee organizes a number of satellite events attached to its 15th International Real-Time Systems Conference.

After the enormous success of last year's edition (http://www.hurray.isep.ipp.pt/rtlia2002/), we are proud to announce the 2nd International Workshop on Real-Time LANs in the Internet Age (RTLIA03).

One of the main technological challenges today is on developing communication infrastructures that are real-time, reliable, pervasive, interoperable and capable of accommodating the requirements imposed by new applications and services.

There are several research directions. For instance, Networked Embedded Systems pose a number of new challenging issues in terms of real-time, robustness, flexibility and dependability.

The rapid development of new work methods and collaborative work environments has also determined an increasing request for interoperable mobile, wireless technologies and the convergence of fixed and mobile communication infrastructures.

Workshop Website
http://www.hurray.isep.ipp.pt/rtlia2003/

EUROMICRO website
http://www.hurray.isep.ipp.pt/ecrts03

Organising committee:
:

Lucia Lo Bello
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
University of Catania
Italy
phone: +39-095-7382379
fax:+39-095-338280/+39-095-7382397
Lucia.LoBello@diit.unict.it

Eduardo Tovar
IPP-HURRAY Research Group
Polytechnic Institute of Porto , Portugal
Tel: +351 22 8340502
emt@dei.isep.ipp.pt

Luis Miguel Pinho
IPP-HURRAY Research Group
Polytechnic Institute of Porto , Portugal
Tel: +351 22 8340502
lpinho@dei.isep.ipp.pt

Future computer systems will be intimately tied to real-time computing and to communication technologies. Ubiquitous computing networks combining sensors and actuators with computing and communication systems will be used to mitigate the impact of a given disaster on people and their environments. These networks will support the exchange of sensitive real-time information to be used for intelligent assistance to decision-making, rapid planning, resource allocation, incident command response, first-aid.

In Factory Communication Systems, existing solutions will be improved in order to comply with the need of providing not only real-time guarantees to control traffic but also adequate bandwidth to multimedia traffic in a flexible way. New frontiers for real-time Ethernet, in terms of both technological challenges and application areas (e.g. mission critical systems), are going to be investigated.

The goal of this workshop is to bring together people from industry and academia that are interested in all aspects of using commodity LAN technologies to support real-time and dependable applications in the Internet Era. The workshop will provide a relaxed forum to present and discuss new ideas, new research directions and to review current trends in this area. The workshop will be based on short presentations that should encourage discussion by the attendees.

Prospective authors are encouraged to address real-time, QoS and dependability issues of the following topics:

- LAN Technologies: industrial Ethernet, fieldbus networks, Myrinet, etc.
- Control methods for communication over LAN networks
- Wireless sensor networks
- IPV6 networking
- Wired or wireless communication in intelligent buildings
- Ubiquitous computing for disaster response, mitigation and recovery
- Interoperability between LANs and PANs and between LANs and WANs
- Interoperability between hybrid wired/wireless networks
- Communication Technologies for e-Activities: e-health, e-monitoring.etc.
- Support of IP-based applications

- Multimedia communication
- Networked embedded systems
- Evaluation and case studies
- Tools
- Industry experience reports

Statements which are innovative, controversial or that present new approaches are specially sought.

SUBMISSION OF PAPERS: People who would like to participate in this event are asked to submit a pdf or postscript version of a 2 page extended abstract to the workshop chairperson (Lucia.LoBello@diit.unict.it). The papers will be reviewed by an internal committee. Upon acceptance, a final paper of maximum 4 pages should be then be prepared and submitted. All papers will be made available to all participants a week before the workshop so that contributions can be examined prior to the workshop. The workshop sessions will involve 5-10 minute presentations followed by a 20-25 minute discussion. Authors will be asked to produce a final version of their paper that includes the issues covered in the discussions by 20 July, 2003. The final papers will be published by the Polytechnic Institute of Porto.

To foster the workshop character the number of participants will be limited to 30. There will be a small charge for workshop registration.

The workshop will be held at the Polytechnic Institute of Porto in the 1st July 2003. See the Euromicro Conference on Real-Time Systems website for information on the venue, hotel reservations and Porto.

EUROMICRO – The European Association for Microprocessing and Microprogramming.
P.O. Box 2346, NL – 7301 EA Apeldoorn, The Nederlands
Telephone + 31 55 5795503 Telefax + 31 55 5795509 www.euromicro.org euromicro@standby.nl

Engineering Hybrid Wired/Wireless Fieldbus Networks - a case study

Mário Alves, Eduardo Tovar, Luís Marques
IPP-HURRAY! Group, Polytechnic Institute of Porto
Porto, Portugal
{malves@dee,emt@dei}.isep.ipp.pt

Steve Behaeghel, Katrijn Van Nieuwenhuyse
Katholieke Hogeschool Sint-Lieven
Dep. Industrial Engineering - Electronics
Gent, Belgium

Abstract

Advances in networking and information technologies are transforming factory-floor communication systems into a mainstream activity within industrial automation. It is now recognized that future industrial computer systems will be intimately tied to real-time computing and to communication technologies. For this vision to succeed, complex heterogeneous factory-floor communication networks (including mobile/wireless components) need to function in a predictable, flawless, efficient and interoperable way. In this paper we re-visit the issue of supporting real-time communications in hybrid wired/wireless fieldbus-based networks, bringing into it some experimental results obtained in the framework of the RFieldbus ISEP pilot [5].

1. A hybrid wired/wireless fieldbus

1.1 Basics on a hybrid wired/wireless architecture

A traditional fieldbus network consists of several nodes physically connected through a wired bus. Therefore, and due to market penetration, thinking about wireless means considering hybrid wired/wireless solutions able to inter-operate with legacy (wired) systems. We assume hybrid wired/wireless network topologies such as the one depicted in Figure 1, where PROFIBUS is considered as the federating system [1].

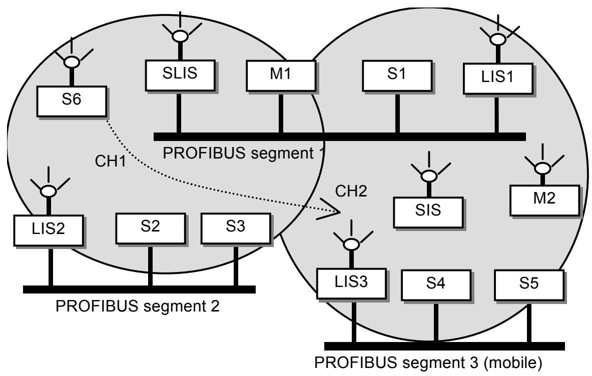


Figure 1: Hybrid wired/wireless network

Wired network master (M) and slave (S) nodes communicate with wireless/mobile nodes through Intermediate Systems (ISs), which can be of three types: Linking ISs (LISs); Structuring ISs (SISs); and Structuring and Linking ISs (SLISs). The latter merge the functionality of LISs and SISs. Additionally, mobility of wired PROFIBUS segments (and associated stations) is also considered (segment 3). For the example scenario, the SLIS and the SIS structure the wireless part of the network in two different radio cells, operating in different radio channels (CH1 and CH2, respectively), in order to support inter-cell mobility [2].

ISs operate at the Physical Layer level, leading to a broadcast network (stations receive every transmitted frame), with a unique MAC address space and a unique logical ring. PROFIBUS v1 (RS-485, asynchronous) is considered as the wired physical medium and the wireless physical medium is based on IEEE 802.11b [2].

A mechanism for supporting inter-cell mobility for all types of mobile stations is described in [1]. Due to the broadcast nature of the network, the use of explicit registration mechanisms could be avoided (the mobility management mechanism was almost reduced to a procedure for radio channel assessment and switching).

The *mobility master (*MobM*)* triggers the procedure by broadcasting a *Beacon Trigger* (BT) frame (Figure 2). SISs/SLISs start transmitting beacons in their own radio channel (CH1, CH2) and every wireless/mobile station (e.g. S6) is expected to assess the quality of the different radio channels, switching to the best quality one (CH2, assuming that S6 is moving to that radio cell).

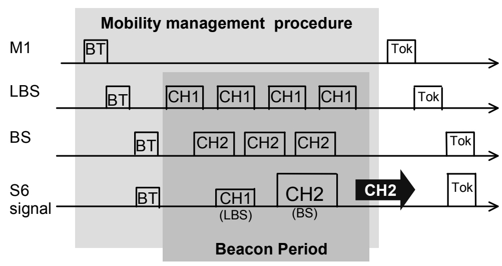


Figure 2: Mobility management timing diagram

Importantly, this mechanism guarantees no loss of data and permits to fulfil stringent real-time requirements. In fact, mobility management is restricted to a reduced and bounded period of time (typically below 4 ms overhead per second) [1].

1.2 Analytical models for the network

In order to analyse and guarantee the real-time behaviour of such a network, there was the need to define the characteristics of all network components, namely the ones that most affect the timing behaviour of the network. A complete analytical model for the addressed hybrid network was proposed in [3], covering the definition of attributes and timing behaviour for network components such as (wired and wireless) domains, end-systems, intermediate systems and physical media. In this paper, we will only briefly outline the models of the intermediate systems and of the physical media.

Defining a model for the Physical Media (e.g. bit rate and PhL frame format) is mandatory to compute the duration of a PhL frame and characterising the relaying behaviour of an IS. A physical medium (*M*) is defined by the following parameters: *r* - bit rate; l_H - overhead of the head per PhL frame; l_T - overhead of the tail per PhL frame; *k* - overhead per char for the PhL protocol; *o* - offset defining the total number of bits until knowing the length of the data field (Figure 3).

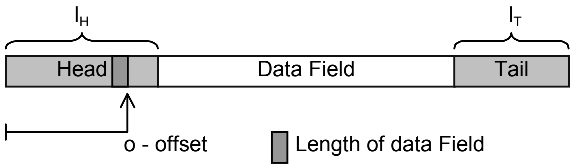


Figure 3: Generic format of a PhL frame

The offset *o* is relevant for the definition of the relaying behaviour of the ISs, as briefly described next.

The model for the intermediate systems proposed in [3] enables the definition of a minimised latency repeater (cut-through behaviour); that is, it permits to define a profile for a repeater that starts relaying PhL frames as early as possible. An Intermediate System is defined by several parameters, such as its type (SIS, LIS, SLIS) and the type of physical media it interconnects. Additionally, a start-relaying instant function – $t^{i\rightarrow j}_{sr}$ – determines the earliest time instant for start relaying a specific PhL frame from a communication domain D^i to a communication domain D^j, counted since the beginning of the PhL frame in domain D^i. It is assumed that relaying cannot start: while the first char of DLL data is not available (data ready - t_{dr}); while the length of the DLL frame is not known (length known - t_{lk}); the transmission of a PhL frame in D_j must be continuous, without time gaps (no gaps - t_{ng}). The start-relaying instant for a cut-through IS is defined as (Figure 4):

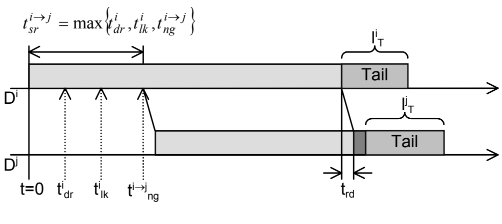


Figure 4: Relaying behaviour of a cut-through IS

2. Guaranteeing real-time communications

Several methodologies were proposed in [3] to guarantee the real-time behaviour of the hybrid network. This involves computing and setting several parameters: the Idle Times that must be inserted (by every master) in order to adapt different PhLs, the Slot Time and worst-case duration of message transactions, and the parameters for the mobility management mechanism.

2.1 Adapting heterogeneous physical media

In PROFIBUS, a response to a request has to be received within the Slot Time (T_{SL}), otherwise the master retries the request (token) or aborts the transmission. In a hybrid broadcast network such as the one considered, message turnaround times increase due to the relaying latencies in the ISs (Figure 5):

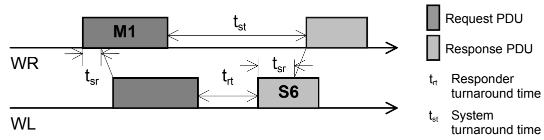


Figure 5: Responder and system turnaround times

Additionally, frames may be affected by unbounded queuing delays in the ISs, due to the interconnection of different physical media (bit rates and frame formats). Therefore, an upper limit for the system turnaround time of a message transaction could not be computed if an appropriate adaptation mechanism was not provided. A solution is to delay request frames through the insertion of additional idle time, in master stations [3,4] (Figure 6).

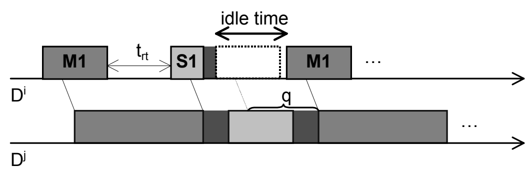


Figure 6: Inserting idle time to adapt different media

Importantly, this mechanism relies of standard features of the PROFIBUS protocol – the Idle Time parameters.

2.2 Computing T_{SL} and C_{ack}

In order to set an appropriate value for the Slot Time (T_{SL}) parameter, it is necessary to compute t_{st} (worst-case system turnaround time) for every possible message transaction in the network. The work described in [3] proposes methodologies in order to compute the Slot Time parameter – T_{SL}, and also the worst-case duration of every message transaction (C_{ack}):

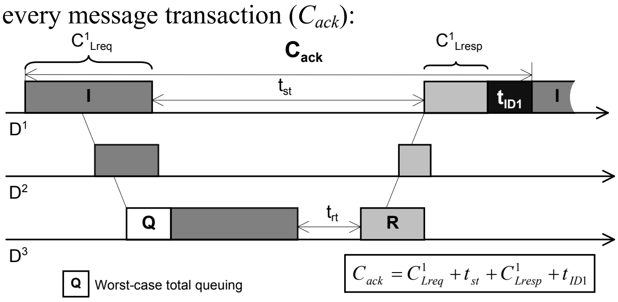


Figure 7: Duration of a transaction (C_{ack})

2.3 Computing the mobility management parameters

In [3] the evaluation of the proper value to set T_{ID2} in the mobility master, which is related to the worst-case duration of the mobility management procedure, is also detailed. This worst-case duration depends on the number of beacons that each SIS/SLIS must transmit, after having received (and relayed) the BT frame (can be different for every SIS/SLIS). Therefore, it is mandatory to compute the optimal (minimum) number of beacons (n_b) to be transmitted by each SIS/SLIS.

3. System planning application

The computation of all parameters is quite complex, time-consuming and error-prone (e.g. calculation errors or missing of alternative paths due to mobility). These issues triggered the need to develop of a System Planning Application (SPA) to reduce the complexity and the time spent in system design and to minimize the probability of errors. This tool was successfully used to engineer the RFieldbus system described in [5].

The SPA user creates a virtual image of the network topology by dragging items from a toolbox (Figure 8, left) into the drawing area (Figure 8, right).

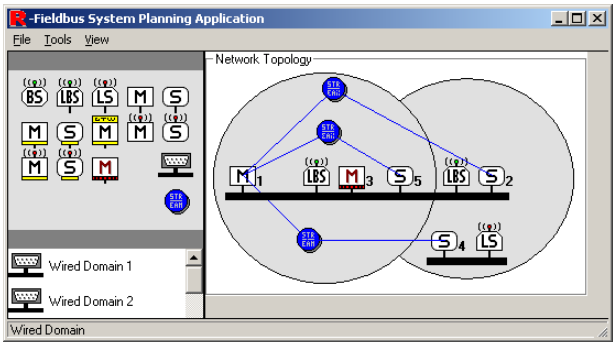


Figure 8. System Planning Application

When adding a new station, it will be automatically assigned an unused address. To add one wireless domain (represented by an oval gray area), we have to add one structuring intermediate system (Base Station – BS, or Link Base Station - LBS). To add a message stream, the user selects the Message Stream tool and then clicks in the Initiator and Responder (e.g. M1 and S2). Multiple message streams between two stations are grouped into the same connector. The physical media must be defined by setting the appropriate parameters (Figure 9).

After having configured all input parameters, the application computes the parameters for setting up the network. First, the network topology is verified to avoid closed-loops. Then, all alternative paths (due to mobility) are determined, for every message stream. Finally, all the parameters are computed and the main results window is presented (e.g. Figure 10), as briefly outlined next.

For the example scenario shown in Figure 8, two types of physical media are configured as shown in Figure 9, corresponding to the ones defined in the RFieldbus framework (1 – wired (PROFIBUS v1), 2 - Wireless).

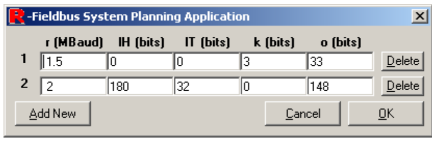


Figure 9. Physical media parameters

The first wired segment (topmost in Figure 8) includes four stations: M1 – Master, M3 – Mobility Master, S2 and S5 - Slaves, and two structuring intermediate systems (two Link Base Stations). The second segment holds a single station: S4 – Slave, and one intermediate system (a Link Station). The results are shown in Figure 10.

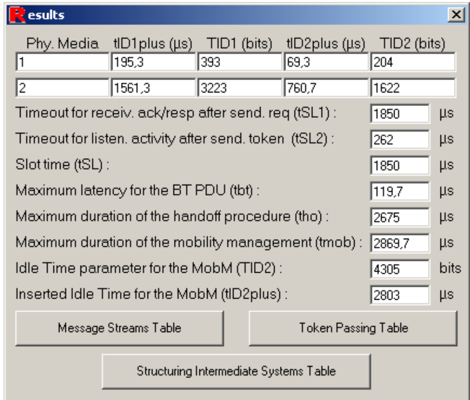


Figure 10. The main results window

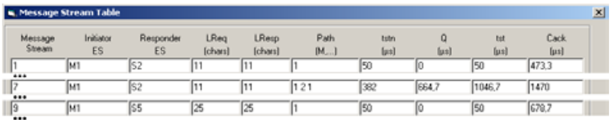


Figure 11. The results window for message streams

Each LBS must transmit 22 beacons, as detailed in the "Structuring Intermediate Systems Table" (not shown here).

4. Theoretical vs. experimental results

In this section, the worst-case results output by the System Planning Application (SPA) are compared against measurements carried out using a network analyser, in order to analyse their validity and degree of pessimism. When matching theoretical with experimental results, for the case study described in the previous section, an unexpected timing behaviour was noticed. A closer study led to the conclusion that the RFieldbus boards [6] introduce additional delays, both when acting as responder and as initiator. As a responder, this delay has a considerable influence on the turnaround times (t_{rt}, t_{st}) of every message stream. As an initiator, this impacts T_{ID1} and C_{ack}. Therefore these delays must be considered in the theoretical computation, as follows.

For the responder, $T_{rt\_min} = 15$ and $T_{rt\_max} = 75$ bit times were originally assumed as realistic values, because native PROFIBUS slave boards (e.g. PLC) usually have a short T_{rt}. After several measurements, it was concluded that the RFieldbus boards may have turnaround times of almost 400 bit times. For instance, this was noticed for the message transaction between two PC's in the same domain (Figure 12). The T_{st} of 270 bit times is much higher than the worst-case value of 75 bit times initially assumed as input to the SPA. These additional delays must be taken into account, by changing the T_{rt_min} and the T_{rt_max} parameters in the SPA, according to the real $T_{rt\ min}$ = 12 bit times and T_{rt_max} = 370 bit times that were measured.

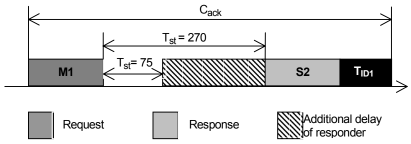


Figure 12. Transaction for message stream 1

Let's now address the case where the RFieldbus board is used as initiator. The value of 393 bit times for T_{ID1}, calculated by the SPA Figure 10, is indirectly set in the configuration file of the master by manipulating the min T_{SDR} parameter. Figure 13 depicts a message transaction between the master and the PLC. The inactivity time measured between receiving the response and transmitting a request is 562 bit times. Therefore, we conclude that the RFieldbus board in the master (initiator) inserts an additional inactivity time of around 250 (=562-393) bit times, which happens for every message transaction. This problem can be solved by setting T_{ID1}=143 (=393–250) bit times, in the master.

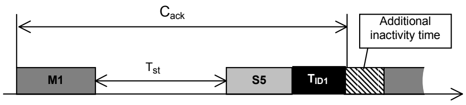


Figure 13. Transaction for message stream 9

The same conclusions could be drawn for a stream between two PC's in different domains (Stream 7, Figure 14). The only difference is that in most cases it is not possible to notice the delay inserted by the slave RFieldbus board, because the SPA takes the Q into account for the computation of both T_{st} and C_{ack}.

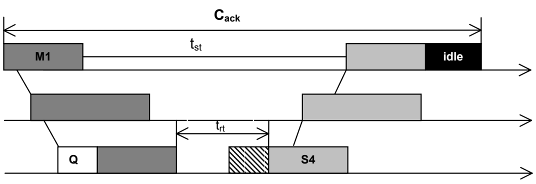


Figure 14. Transaction for message stream 7

Table 1 depicts the (re)-computed worst-case and the measured durations for the three streams.

Message Stream	C^{WC}_{ack} *(bit times)*	C^{M}_{ack} *(bit times)*	*Pessimism (%)*
1	1008	916	10
9	1317	966	36
7	2504	1400	79

Table 1. Worst-case vs. measured C_{ack} (C^{WC}_{ack}, C^{M}_{ack})

The table also provides a column with the pessimism introduced by the SPA, which is calculated as $[(C^{WC}_{ack}/C^{M}_{ack})-1]\times100\%$.

For stream 1, where the initiator and responder are in the same domain, T_{st}=T_{rt_max}=370 bit times. Since the actual T_{st}=270 bit times, there is a pessimism of 10% ($C^{WC}_{ack} \approx C^{M}_{ack}+T_{rt\_max}-T_{st} \approx 916+370-270 \approx 1008$ bit times). For stream 9, the fact that the actual T_{st} for the PLC is equal to T_{rt_min}=12 bit times, it results in a pessimism of 36% ($C^{WC}_{ack} \approx 966+370-12 \approx 1317$ bit times). Message streams with initiator and responder situated in different domains often have more pessimistic results. This is because Q is taken into account in the worst-case analysis and real queuing delay in the case study depends on the amount of traffic in the network. Therefore, the pessimism for stream 7 is 79%.

5. Conclusions

This paper started by summarizing the most important architectural issues for a hybrid wired/wireless fieldbus network based on the PROFIBUS protocol. Then, it outlined several aspects of paramount importance for guaranteeing real-time communications in such a network. The basics of a System Planning Application (SPA) were also presented. This tool turned out to be very important to compute all relevant parameters for putting the hybrid network into operation. Obviously, its advantages increase as networks get more complex, since manual computation would be very time consuming, complex and, most probably, involving errors. Finally, theoretical worst-case results were compared against experimental results, for a scenario within the context of the manufacturing automation field trial of the RFieldbus project. The use of both the SPA and a network analyzer permitted to analyse the validity and degree of pessimism of the theoretical results and also to detect some unexpected delays in the RFieldbus boards.

References

[1] Alves, M., Tovar, E., Vasques, F., Hammer, G. and K. Roether. "Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-based Networks", in Proc. of the ECRTS'02, pp. 142-150, 2002.

[2] Rauchhaupt, L., "System and Device Architecture of a Radio Based Fieldbus – The RFieldbus System", in Proc. of WFCS02, 2002.

[3] Alves, M., "Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-Based Networks", PhD Thesis, University of Porto, 2003.

[4] Alves, M., Tovar, E. and F. Vasques, "On the Adaptation of Broadcast Transactions in Token-Passing Fieldbus Networks with Heterogeneous Transmission Media", in Proc. of FET'2001, pp. 278-284, 2001.

[5] http://www.hurray.isep.ipp.pt/rfpilot

[6] Pereira, N. et al., "Integration of TCP/IP and PROFIBUS Protocols", in WIP Proc. of the WFCS02, 2002.