Real-time communications over wired/wireless PROFIBUS networks supporting inter-cell mobility

Mário Alves, Eduardo Tovar

Abstract

PROFIBUS is an international standard (IEC 61158, EN 50170) for factory-floor communications, with several thousands of installations worldwide. Taking into account the increasing need for mobile devices in industrial environments, one obvious solution is to extend traditional wired PROFIBUS networks with wireless capabilities. In this paper, we outline the major aspects of a hybrid wired/wireless PROFIBUS-based architecture, where most of the design options were made in order to guarantee the real-time behaviour of the overall network. We also introduce the timing unpredictability problems resulting from the co-existence of heterogeneous physical media in the same network. However, the major focus of this paper is on how to guarantee real-time communications in such a hybrid network, where nodes (and whole segments) can move between different radio cells (inter-cell mobility). Assuming a simple mobility management mechanism based on mobile nodes performing periodic radio channel assessment and switching, we propose a methodology to compute values for specific parameters that enable an optimal (minimum) and bounded duration of the handoff procedure.

Keywords

PROFIBUS; Wireless communications; Inter-cell mobility; Real-time communications

1. Introduction

1.1. Context and structure of the paper

The communication infrastructure of current process control and factory automation systems is usually based on fieldbus networks, since they provide adequate levels of performance, dependability, timeliness, maintainability and cost. Nevertheless, cabling starts to be an obstacle for an increasing number of industrial automation applications, which impose or benefit from the use of mobile devices (e.g. handheld computers or transportation equipment).

Within this context, there is a trend to extend fieldbus systems with wireless capabilities, leading to hybrid wired/wireless communication networks,

which must fulfil the same stringent requirements. Wireless communications must cope with real-time and dependability features at least similar to the ones found in traditional fieldbus networks. The support of inter-cell mobility turns this task even more difficult, since mobile nodes must handoff between radio cells in a transparent way.

In this paper, we address the problem of guaranteeing real-time communications in hybrid wired/ wireless PROFIBUS networks where wireless nodes can move between different radio cells. PROFIBUS (PROcess FIeldBUS) is the leading fieldbus technology with over 14 million nodes installed worldwide [1], for applications ranging from factory automation to process control.

The paper starts by outlining the most relevant aspects of the PROFIBUS lower layers. The basic features of a PROFIBUS-based architecture for hybrid wired/wireless networks are then presented, namely some mechanisms and approaches to support and guarantee real-time communications with such architecture. A particular focus is then given to how real-time communications can be guaranteed and to how (inter-cell) mobility can be supported without affecting the real-time performance of the network. We finalize the paper with an elucidative case study, where the proposed methodology is applied to a practical example.

We show that the duration of the selected mobility management mechanism is small (a little bit over 0.1% of time overhead in a typical hybrid network such as the one provided in the case study), bounded and can be computed *a priori* (pre-runtime). Importantly, the methodologies presented in this paper have already been successfully applied in the scope the European Project RFieldbus (IST-1999-11316) [2], namely to pilot field-tests where prototype cut-through repeaters and wireless interfaces were used [3].

1.2. Related work

Existing standardized fieldbus protocols (IEC, CENELEC) rely on wired buses, making them unsuitable for supporting wireless/mobile devices. Recently, several wireless LAN standards, like IEEE 802.11b [4], IEEE 802.15.1/Bluetooth [5] or IEEE 802.15.4/Zigbee [6,7] have appeared and could be used as a basis for the development of industrial wireless solutions. However, industrial communication systems with wireless/mobility capabilities must fulfill the same basic requirements of traditional wired fieldbus networks. The problem arising from the use of commercially available wireless technologies is that they were not designed having industrial applications in mind. Therefore, most of the research work in this field addresses the adaptation or extension of standard communication protocols to guarantee real-time performance and high reliability levels.

One of the first contributions in this scope was the definition of a MAC mechanism based on a Time Division Multiple Access (TDMA) scheme to provide a wireless extension for WorldFIP [8]. In [9], the author suggested the use of Digital Enhanced Cordless Telecommunications (DECT) (ETSI standard: ETS 300 175, Parts 1 to 8) for supporting wireless communications between MAP/ MMS nodes (where MAP is the acronym for Manufacturing Automation Protocol and MMS the acronym for Manufacturing Message Specification). In such architecture, the mobility of wireless nodes was supported by the native mobility mechanisms offered by DECT.

In [10], the authors addressed IEC/ISA (ISA is the acronym for the Instrumentation, Systems, and Automation Society) fieldbus wireless extensions by proposing a modified IEEE 802.11b protocol. Support for real-time communications in wireless domains uses the Point Coordination Function (PCF) offered by the IEEE 802.11b protocol. This functionality periodically creates a contention free period, which can be used to exchange periodic data without interference from the remaining traffic in the network. This approach very much resembles the Flexible Time Triggered operation, which is described in [31], albeit not in the wireless context. In [11], the authors studied the behaviour of an IEEE 802.11 network and how its mobility mechanisms would affect the real-time traffic in the network. [12,13] proposed a wireless protocol based on the IEEE 802.11b MAC and PROFIBUS, in conjunction with a polling mechanism that ensures a deterministic performance. Nonetheless, the proposed mechanism requires changes to the PROFIBUS protocol operation. Additionally the proposed system does not support wireless PROFIBUS masters (only wireless slaves).

Willig analysed the capabilities of the PROFIBUS DLL together with the IEEE 802.11b PhL to support a wireless fieldbus network. In [14,15] the authors studied the ring stability of PROFIBUS over error prone links. They performed field measurements on the Bit Error Rate (BER) in an industrial environment, and used this data for the development of a simulation model of a wireless PROFIBUS MAC. The authors concluded that with the BER levels encountered in most common modems (available at that time), PROFIBUS could be inadequate as a wireless protocol, since the probability of losing or corrupting a token on the wireless domains could be very high. Nevertheless, recent advances on wireless modems (like the REKA transceivers) enable a lower BER [32] and consequently the probability of token loss to acceptable values. Notably, the REKA transceivers are used in RFieldbus. Also, in [15], the authors propose some changes to the PROFIBUS protocol and provide some guidelines which improve the operation of PROFIBUS ring management mechanisms in the presence of errors. Based on these findings, in [16] the author proposed and compared the use of polling-based communication algorithms with standard PROFIBUS, again in a wireless error prone environment, and concluded that these algorithms were capable of offering higher reliability characteristics.

More recent works address the extension of PROFIBUS-DP to operate over a Bluetooth wireless link [17] and a hybrid wired/wireless network supported by Ethernet and Bluetooth [18]. The proposed architecture maintains the compatibility at the Application Layer (AL) level but the PROFIBUS MAC sub-layer is replaced by the MAC sublayer of Ethernet or Bluetooth protocols, which are provided with extensions that insure their real-time operation.

There are also some commercially available products for providing wireless extensions to traditional (wired) fieldbus networks, usually based on interconnecting devices operating as simple repeaters. For example, ALSTOM provides a radio extension to WorldFIP networks and KVASER provides a wireless extension to CAN (WAVEcan). It is interesting to point out that this latter solution partially results from the MOFDI (Mobile Fieldbus Devices in Industry) European project (referred later in this section). Elprotech, Satel, HMS, RadioLinx, Siemens, Prosoft Technology and Phoenix Contact are just examples of companies providing basic wireless (radio or infra-red based) extensions to PROFIBUS. Nevetheless, only one wireless domain is supported, therefore inter-cell mobility is not considered.

Concerning existing research efforts, [19] summarizes some architectural approaches for hybrid wired/wireless fieldbus networks. It is worthwhile to highlight the proposals related to the RFieldbus European Project, both on the original architecture (based on repeaters) [20] and on an alternative architecture (based on bridges) [21]. These provide complete architectures where PROFIBUS is used as the federating communication technology, where multiple wired segments and multiple wireless cells can interoperate, inter-cell mobility of nodes (and wired segments) are supported and real-time communications are guaranteed. A comparative performance analysis between the two approaches is provided in [22].

Besides RFieldbus, some European Projects targeted wireless extension solutions for fieldbuses. One of the firsts was ESPRIT project 7210 – Open Low-Cost Time-Critical Wireless Fieldbus Architecture (OLCHFA) [23] which developed a wireless extension for the WorldFIP fieldbus. This project was followed by MOFDI (ESPRIT 27035 – Mobile Fieldbus Devices in Industry), already mentioned, that developed point-to-point wireless links based on the Bluetooth technology for CAN networks [24]. Unfortunately, to the authors' best knowledge, there is almost no available information about these projects.

Most existing mobility management mechanisms (e.g. IEEE 802.11) are based on explicit registering mechanisms, where devices structuring the wireless cells (e.g. access points) are responsible of maintaining (exchanging) updated information of all nodes belonging to their wireless cell. This paper addresses the timing analysis and engineering of an implicit mobility management mechanism, since we envisaged the design of a PROFIBUS-based hybrid wired/wireless architecture supporting inter-cell mobility with the requirement of the proposed addons to be backward compatible with the PROFIBUS protocol. This will be further explained in Section 3.3.

2. Relevant features of PROFIBUS data link and physical layers

2.1. Overview

The PROFIBUS protocol [25] is based on the OSI (Open Systems Interconnection) reference model [26], although only the Physical Layer (PhL), the Data Link Layer (DLL) and the Application Layer (AL) are defined and implemented.

A maximum of 32 nodes (masters or slaves) can be supported in a single segment. However, the network can be extended to a maximum of 126 nodes by using more segments in a linear or tree-like topology, provided that the segments use the same physical layer protocol and no more than three repeaters exist in the path between any pair of nodes. The maximum cable length for a single segment depends on the bit rate, ranging from 1200 m (for lower bit rates ~~9~~.6 up to 93.75 kbit/s) down to 100 m, if higher bit rates (3 up to 12 Mbit/s) are used.

A master can send a message on its own initiative, once it receives the token, which circulates between masters in a logical ring fashion. Slaves do not have bus access initiative; therefore they only acknowledge or respond to requests from masters. A message cycle (or transaction) comprises the request frame sent by an *initiator* (always a master) and the associated acknowledgement or response frame from the *responder* (usually a slave).

The PROFIBUS Medium Access Control (MAC) protocol, being based on the measurement of the actual token rotation time, induces a well-defined timing behaviour for the transferred messages, since the token cycle duration can be estimated prior to run-time [27]. Both high and low priority messages are supported, as well as three acyclic data transfer services: *Send Data with Acknowledgement* (SDA), *Send Data with No acknowledgement* (SDN) and *Send and Request Data* (SRD). The SDN service is of particular importance in the scope of this paper, since it is used in the mobility management mechanism.

2.2. Message transactions

After a master issues a request frame, the corresponding acknowledgement or response frame must arrive before the expiration of the slot time (T_{SL}), otherwise the initiator repeats the request (a predefined number of times) or aborts the transaction (if no more retries are allowed). Therefore, message turnaround times (the time span since a request frame is completely transmitted by the initiator, until it starts receiving the corresponding response frame), must always be smaller than T_{SL}. Fig. 1 illustrates a timing diagram with a scenario where a first message transaction has succeeded, followed by another message transaction where an error occurred (response did not arrive to the master before the expiration of T_{SL}).

Before issuing a request (or token) frame, the master must wait a time interval defined by the idle time (T_{ID}) parameter (also illustrated in Fig. 1), in order to create an inter-frame synchronising period of idle bits (at least 33 idle bit periods) [25].

Both T_{SL} and T_{ID} are standard PROFIBUS parameters that must be properly set (in master nodes) prior to run-time. As it will be clear throughout the remainder of this paper, these parameters are of particular importance for engineering heterogeneous PROFIBUS networks, and therefore additional reasoning on these two parameters is provided in Sections 2.3 and 2.4 for T_{ID} and T_{SL}, respectively.

2.3. Further details on the idle time (T_{ID}) parameters

The idle time is a period of physical medium inactivity that is inserted by master stations between consecutive message transactions. After an acknowledgement, response or token frame, a master station inserts an idle time with a value given by

$$T_{\mathrm{ID1}} = \max\{T_{\mathrm{SYN}} + T_{\mathrm{SM}}, \min\{T_{\mathrm{SDR}}^{i}\}, T_{\mathrm{SDI}}\}$$
$$\forall i \in \text{Responders Set}. \qquad (1)$$

T_{SYN} (synchronisation time) is the minimum time interval during which each station must receive idle state from the physical medium (33 bits); T_{SM} is a safety margin; T_{SDR}^{i} is the station delay of responder *i*; T_{SDI} is the station delay of the initiator. Refer to the PROFIBUS standard [25] for further details on these parameters.

Conversely, after an unacknowledged request frame, a master station must insert an idle time given by

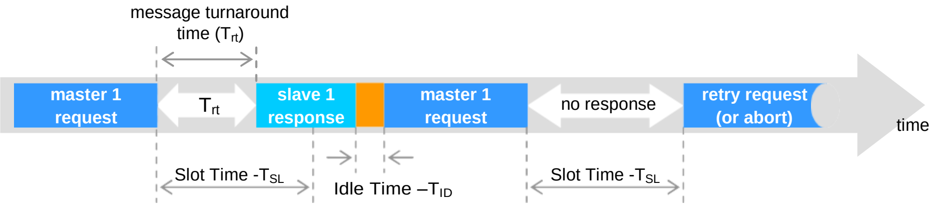


Fig. 1. The PROFIBUS slot time (T_{SL}) and idle time (T_{ID}) parameters.

$$T_{\mathrm{ID2}} = \max\{T_{\mathrm{SYN}} + T_{\mathrm{SM}}, \max\{T^{i}_{\mathrm{SDR}}\}\}$$
$$\forall i \in \text{Responders Set.} \qquad (2)$$

Fig. 2 illustrates the use of T_{ID1} (Fig. 2a) and T_{ID2} (Fig. 2b).

Throughout the paper, parameters denoted as 'T' represent bits, while parameters denoted as 't' represent time. Additionally, the station delay of the responder t_{SDR} (T_{SDR} in time units) will be referred as *responder's turnaround time* – t_{rt}. This is the time span since a request frame is completely received by the responder until it starts transmitting the related response frame.

The idle time parameters can be set in a per-station basis; that is, each master can hold a different value for the (T_{ID1}, T_{ID2}) pair. Eqs. (1) and (2) are valid for a single segment network. In a network composed of several domains with different physical media (different bit rates and (Physical Layer) frame formats), the idle time parameters must be derived differently [20].

2.4. Further details on the slot time (T_{SL}) parameter

The slot time is a parameter used by a master node to detect communication or node errors that lead to abnormal medium inactivity. A master node always checks if the time elapsed between the transmission of the last bit of a request (or token) frame and the reception of the first character of the following frame (transmitted by another node) is smaller than T_{SL}. If this does not happen, the master retransmits the frame (request or token) or aborts the transmission.

To set the T_{SL} parameter, it is necessary to compute two different components: 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 frame from the responder (R), after transmitting the last bit of the request frame (Fig. 3a).

T_{SL1} can be computed as follows:

$$T_{\mathrm{SL1}} = 2 \cdot T_{\mathrm{TD}} + \max\{T^{i}_{\mathrm{SDR}}\} + 11 + T_{\mathrm{SM}}$$
$$\forall i \in \text{Responders Set.} \qquad (3)$$

T_{TD} is the transmission (propagation) delay; T^{i}_{SDR} is the station delay of responder i; T_{SM} is a safety margin.

T_{SL2} is the maximum time a master node (I^{0} in Fig. 3b) waits after having transmitted the last bit of the token frame until it completely receives the first character of a frame (either a request or the token) transmitted by the master node that received the token (I^{00} in Fig. 3b). T_{SL2} can be computed as follows:

$$T_{\mathrm{SL2}} = 2 \cdot T_{\mathrm{TD}} + \max\{T^{i}_{\mathrm{ID1}}\} + 11 + T_{\mathrm{SM}}$$
$$\forall i \in \text{Initiators Set.} \qquad (4)$$

Contrarily to the idle time parameters, the slot time parameter must be set with the same value in every master in the network (this is imposed by the token

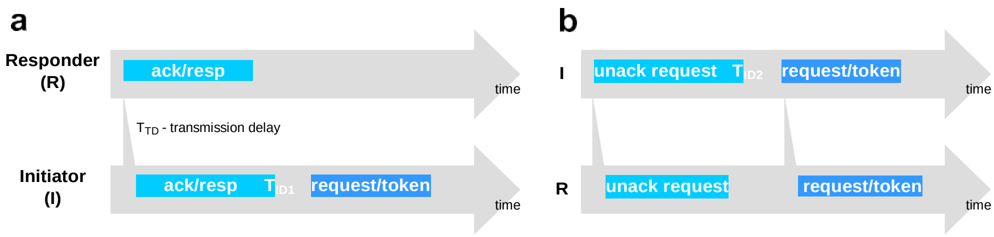


Fig. 2. Illustration of the idle time parameters: (a) TID1 and (b) TID2.

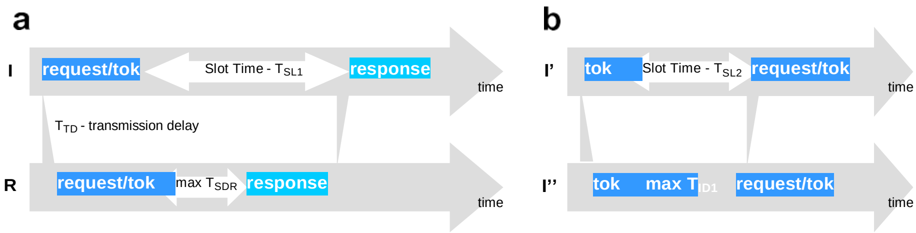


Fig. 3. Illustration of the slot time components: (a) TSL1 and (b) TSL2.

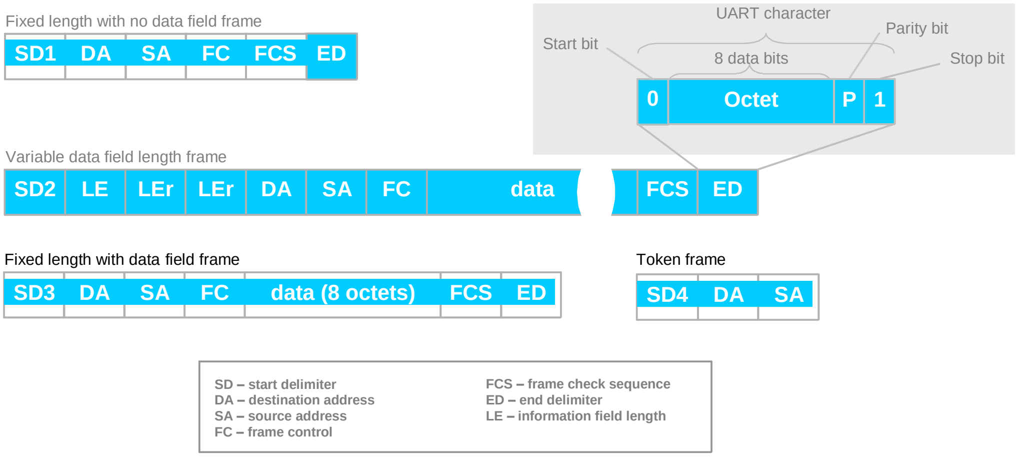


Fig. 4. PROFIBUS frame formats (PhL v1).

management mechanism), which is the maximum between T_{SL1} and T_{SL2}:

$$T_{SL} = \max\{T_{SL1}, T_{SL2}\}. \qquad (5)$$

While Eqs. (4) and (5) are valid for a single segment network, for a network with multiple heterogeneous segments interconnected by repeaters, the appropriate T_{SL} value must be determined using a much more elaborated reasoning [20], as outlined in Sections 3.5 and 3.6.

2.5. Frame formats

PROFIBUS defines four types of Data Link Layer (DLL) frames, each of them characterised by a different Start Delimiter (SD) identifier. Frame formats and contents for these four types are depicted in Fig. 4.

In the RS-485 physical layer, usually used in PROFIBUS-DP implementations, each frame is coded using UART (Universal Asynchronous Receiver/Transmitter) characters, each comprising 11 bits: 1 start bit, 8 data bits, 1 (even) parity bit and 1 stop bit.

3. Major aspects of the hybrid architecture

3.1. Overview of the hybrid architecture

A traditional fieldbus network consists of several nodes (or stations) physically connected through a wired bus. The considered hybrid network architecture is constituted by *End Systems* (ESs) and *Intermediate Systems* (ISs), as illustrated in Fig. 5. End

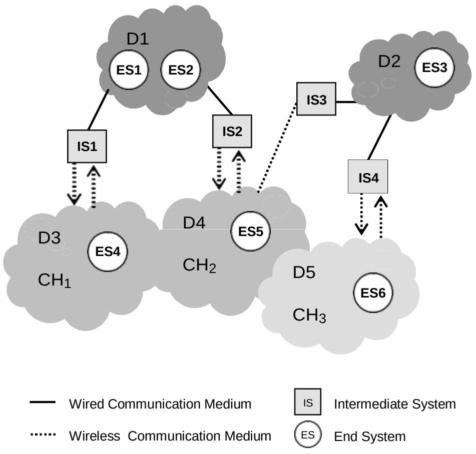


Fig. 5. Example of a hybrid wired/wireless network.

Systems are classically defined as devices that support end-user applications/services and network interfaces. In our context, End Systems are PROFIBUS nodes (masters/slaves), which can have a wired (ES1, ES2, ES3) or wireless interface (ES4, ES5, ES6). Intermediate Systems are devices that are used for network interconnection.

A set of End Systems and Intermediate Systems that communicate directly via a wired physical medium is called a *Wired Domain* (e.g. D1, D2), while a set of End Systems and Intermediate Systems that communicate directly via a wireless physical medium is called a Wireless Domain (e.g. D3, D4, D5).

In order to support inter-cell mobility, radio cells must overlap (such as D3/D4 and D4/D5), to provide continuous radio coverage to wireless/mobile

End Systems (ES4, ES5 and ES6). Moreover, overlapping radio cells must operate in different sets of radio channels (CH1/CH2 and CH2/CH3). Additionally, radio cells must be structured (instead of ad-hoc), i.e. the Intermediate Systems (IS1, IS2, IS4) must implement a ''base station'' functionality, so they are called Structuring Intermediate Systems. All communications in a structured wireless domain are relayed through the correspondent structured IS (in D3, all frames are relayed by IS1; in D4, by IS2 and in D5, by IS4). A set of two radio channels is used within each radio cell – one uplink and one downlink (note the dashed arrows in opposite directions, associated to IS1, IS2 and IS4).

IS1, IS2 and IS4 not only serve as base stations for their wireless domains, but also interconnect a wired domain and a wireless domain (D1 to D3, D1 to D4 and D2 to D5, respectively), i.e. they have a ''linking'' functionality. Additionally, IS3 is a type of Intermediate System that only has ''linking'' functionality (it interconnects D2 and D4), i.e. it does not behave as a Structuring Intermediate System.

All Intermediate Systems are assumed to behave as repeaters, i.e. they relay frames at the physical layer level, without any address filtering. The wireless domains can therefore be seen as simple extensions of the wired parts, resulting in a ''broadcast'' type of network, where all End Systems listen to every transmitted frame.

A more detailed and formal definition of the network architecture and a description of interoperability rules can be found in [20].

3.2. Protocols for wired and wireless domains

Fig. 6 illustrates the protocols used by the Application (AL), Data Link (DLL) and Physical (PhL) Layers of the wired and wireless domains. The AL protocol is considered to be PROFIBUS-DP, for both Wired and Wireless Domains. Wired Domains use the PROFIBUS PhL v1 (RS-485 asynchronous version), while a PhL protocol similar to the one defined in the IEEE802.11b standard is assumed for the Wireless Domains. Wireless ESs use a specific wireless PhL (non-PROFIBUS), thus wired and wireless ESs use the same DLL protocol (i.e. PROFIBUS).

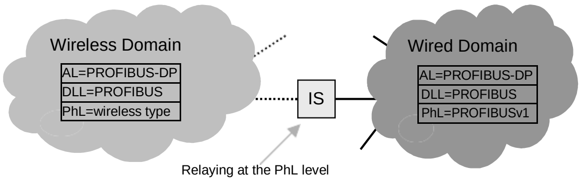


Fig. 6. Protocols for wired and wireless domains.

A network based on repeaters permits to implement a simple handoff mechanism based on radio channel assessment and switching, as it is briefly described in next. However, it should be noted that the use of repeaters in such hybrid architecture still introduces additional communication latencies (when compared to the pure wired solution) that must be taken into consideration.

3.3. Outline of the mobility management mechanism

The mobility management mechanism assumed for the approached architecture was developed within the RFieldbus Project (IST-1999-11316). It was firstly described in an project internal document [28] and then published in [29,20]. This mechanism provides a seamless handoff for all kinds of mobile ESs (master/slave) and also for Wired Domains.

The fact that the interconnection between wired and wireless domains is performed by repeater devices (Layer 1 Intermediate Systems) leads to a ''broadcast network'' (every message is listened by every node in the network, one logical domain, a single token circulating throughout all the network). Since the underlying communication protocol is PROFIBUS, only one node in the overall network is able to communicate at a given time instant. This permits a different approach concerning the handoff mechanism.

More specifically, it is not mandatory to have node location information in each SIS (''access point''), since all messages are broadcast throughout the overall network. Thus and since there is no need for explicit registration mechanisms (kind of routing tables) in the ''access points'', a simpler handoff mechanism can be implemented, bypassing the usually complex handoff mechanisms that are based on registration and location awareness information in all ''access points''.

The mobility management mechanism is therefore rather straightforward, just encompassing a procedure for radio channel assessment and switching. Basically, a master station (MobM) has the responsibility of periodically triggering the mobility mechanism by broadcasting a beacon trigger (BT) frame throughout the network. Upon reception of the BT frame, each SIS sends a predefined number of beacons in its own radio channel and mobile nodes just have to assess the quality of the available radio channels and switch to the best one.

Let us report to the network topology depicted in Fig. 5. One ES – the *mobility master (MobM)* – is responsible for triggering the mobility management procedure (let us assume that ES1 is the MobM), with a periodicity that is dependent on the maximum speed of the mobile stations [28]. This BT frame is broadcast to the entire network and causes every Structuring IS (IS1, IS2 and IS4) to send a number of beacons using its downlink radio channel (CH1, CH2 and CH3, respectively). Within a certain time interval – the *beacon period*, all mobile ESs (ES4, ES5, ES6) are expected to assess the quality of the different radio channels (CH1-3) based on these beacons, finally switching to the channel considered as having the best quality. After this period for the mobility management procedure, the MobM is able to resume "normal" data communication or to pass the token to another master.

Importantly, the proposed mobility management mechanism guarantees that there is no loss of frames (considering no faults) and permits to fulfill stringent real-time requirements. In fact, mobility management is restricted to a reduced, well-determined and bounded period of time (as it will be shown in Section 4).

3.4. Outline of the analytical models of physical media and repeaters

In this section, we outline the analytical models of the network components that most affect the timing behaviour of the network – repeaters and physical media. The model for the Physical Media mainly defines the bit rate and the Physical Layer frame format, while the model for the repeaters characterises its relaying behaviour.

A physical medium can be modelled with 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.

The generic format of a PhL frame is as depicted in Fig. 7. We assume that the DLL frame is embedded in the data field of the PhL frame. It should be noted that the offset o is a relevant parameter for the definition of the timing behaviour of the repeaters (which will be only briefly outlined in this paper).

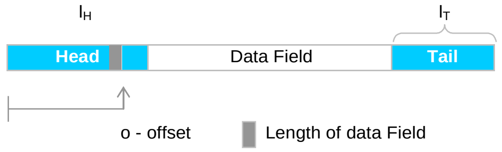


Fig. 7. Generic format of a PhL frame.

In order to compute the duration of a PhL frame, two Data Link Layer parameters must be considered: L – length of the DLL frame; d – number of bits per DLL char. The duration (C) of a PhL frame in segment D^i is then given by

$$C^i = \frac{l_H^i + L \cdot (d + k^i) + l_T^i}{r^i} \tag{6}$$

The model for the Intermediate Systems (repeaters) comprises both cut-through and store and forward relaying behaviours. A minimised latency repeater (cut-through behaviour) is a repeater that starts relaying PhL frames as early as possible. A store and forward behaviour is a particular case of the generic cut-through behaviour, where a PhL frame must be completely received by the input port of the repeater before being retransmitted to the output port.

Since Intermediate Systems interconnect heterogeneous physical media, it is assumed that they must support some sort of encapsulation/decapsulation mechanism (due to different PhL frame formats) and that they are able to receive/transmit at different bit rates.

In order to define the timing behaviour of the repeater, a start-relaying instant function – $t_{\mathrm{sr}}^{i\,!\,j}$ – is defined. It enables the computation of the earliest time instant for start relaying a specific PhL frame from domain D^i to domain D^j, measured from the beginning of the transmission of the PhL frame in domain D^i. The start-relaying instant for a specific repeater depends on its behaviour – either store and forward or cut-through. For a cut-through repeater, the following was assumed:

1. relaying a frame from D^i to D^j cannot start before the first char of the DLL frame of D^i is completely received by the repeater;
2. the PhL frame cannot start being relayed before the length of the DLL frame is known (by the repeater);
3. when relaying a frame from D^i to D^j, the instant for start relaying the PhL frame must take into account that the repeater cannot run out of bits

to relay from D^i to D^j, i.e. the transmission of a PhL frame in D^j must be continuous, without time gaps.

Taking these assumptions into account, the start-relaying instant for a cut-through repeater is given by

$$t_{sr}^{i\to j} = \max\left\{t^i_{dr}, t^i_{lk}, t^{i\to j}_{ng}\right\} \quad \forall i,j \in \text{Domain Set.} \qquad (7)$$

Concerning Eq. (7), t^i_{dr}, the *data ready* instant, is the time instant at which a predefined amount of DLL data has been received from D^i (ready to be relayed). For the cut-through behaviour, it is considered that it is the time instant at which the first DLL char is completely received. t^i_{lk}, the *length known* instant, is the time instant at which the length of the DLL frame in D^i is known. In this case, the offset value for the correspondent Physical Medium is used.

$t^{i\to j}_{ng}$, the *no gaps* instant, is the earliest time instant to start relaying the PhL frame from D^i to D^j in a way that guarantees that the transmission in D^j is continuous. It may be computed by subtracting the duration of the PhL frames (neglecting the tail) in D^i and D^j, and subtracting the duration $((d + k^j)/r^j)$ of the last DLL frame char in D^j.

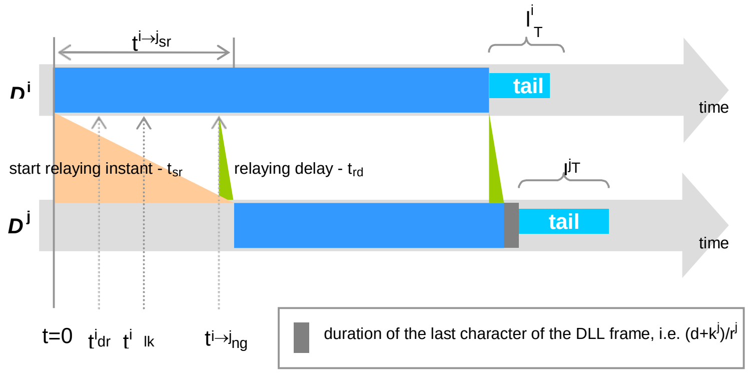
Consider the example depicted in Fig. 8. The first time instant is data readyðt^i_{dr}Þ followed by the time instant when the length of the frame is knownðt^i_{lk}Þ. The last time instant (thus the highest of the three) is the time instant that guarantees a continuous retransmission of the PhL frame ð$t^{i\to j}_{ng}$Þ. This situation usually happens when the duration of the PhL frame in D^j is smaller than in D^i. Nevertheless, and for the general case, any of these time instants can be the highest value between them. Further details on how to compute these time instants can be found in [20].

Two additional parameters must also be set in every IS, prior to run-time: t_{rd} and t_{IDm}. t_{rd} is the (minimum) relaying delay inherent to every IS and is assumed to be the same for every IS in the network. It is also assumed that the ISs always introduce a minimum inactivity period – t_{IDm} (idle time) – between any consecutive PhL frames, according to the requirements of the PROFIBUS protocol. Similarly, we assume the same value for t_{IDm} in every IS in the network.

3.5. The unbounded queuing delays problem

As mentioned before, a master must receive the response to a request within the slot time (T_{SL}). If a timeout occurs, the master retries the request or aborts the transmission. In a network composed of several heterogeneous domains (different physical layer frame formats and different bit rates) interconnected by repeaters (ISs), message turnaround times will increase, due to relaying latencies in the repeaters. These latencies result from the fact that the repeaters must relay frames between domains with different physical layer frame formats and different bit rates.

Consider the network scenario previously outlined in Fig. 5. As depicted in the timing diagram of Fig. 9a), the turnaround time for a message transaction between an initiator and a responder belonging to the same network domain (e.g. ES1 and ES2) is the traditional responder's turnaround time for a PROFIBUS responder node (t_{rt}). However, when initiator and responder belong to different domains (e.g., ES1 and ES4), the turnaround time will increase, as a consequence of the relaying action



$t^{i\to j}_{sr}$
l^i_T
D^i
tail
time
start relaying instant - t_{sr}
relaying delay - t_{rd}
l^j_T
D^j
tail
time
t=0
t^i_{dr}
t^i_{lk}
$t^{i\to j}_{ng}$
duration of the last character of the DLL frame, i.e. $(d+k^j)/r^j$

Fig. 8. Relaying behaviour of a (cut-through) repeater.

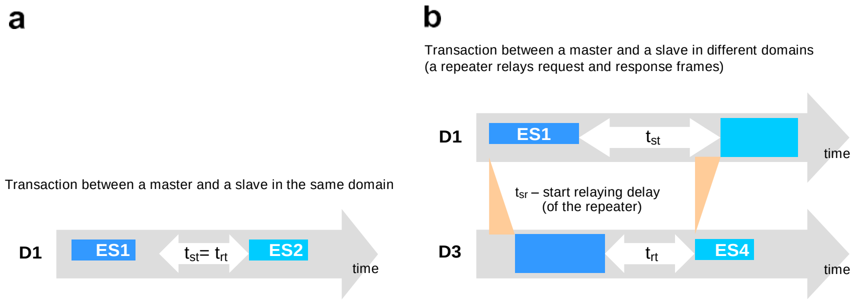

a
Transaction between a master and a slave in the same domain
D1
ES1
tst= trt
ES2
time
b
Transaction between a master and a slave in different domains
(a repeater relays request and response frames)
D1
ES1
tst
time
tsr – start relaying delay
(of the repeater)
D3
trt
ES4
time

Fig. 9. Turnaround times with one (a) and two (b) domains.

performed by the repeater (Fig. 9b)). In this paper, this end-to-end turnaround time is denoted as system turnaround time (t_{st}), which includes a start relaying delay (t_{sr}) introduced by the repeaters. In Fig. 9, the domains are assumed to have the same bit rate and PhL frame format.

An obvious problem is that frames may be affected by unbounded queuing delays in the repeaters. This is exemplified in Fig. 10, where media heterogeneity is assumed to result from different bit rates and/or different PhL frame formats (and therefore different frame durations) for the two domains. The bit rate is assumed to be lower in Domain 2 (D2).

As can be seen in Fig. 10, the fourth message transaction (between master ES1 and slave ES4) is affected by queuing delays that were originated by a sequence of 3 transactions between ES1 and slave ES2 in D1, imposing a system turnaround time t_{st4} that is much longer than the "traditional" responder's turnaround time (t_{rt}).

In fact, the queuing delay can be unbounded. Consider the following elucidative example. Assuming ES1 as the only master in the network, several token frames could be transmitted consecutively during a certain time interval, due to the master having no messages to transmit. If after that sequence of self-passing the token ES1 initiates a message transaction with a slave in another segment (e.g. ES4), that request frame will experience a significant queuing delay in the repeater, since the repeater must first relay all pending token frames to the "slower" network domain (D3). This results from the "broadcast" nature of the system.

Since, generically, these queuing delays cannot be bounded [20], it would not be possible to compute an upper bound for the system turnaround time of message transactions between initiator and responder in different domains. Note that computing the worst-case system turnaround times is crucial for finding a minimum value for setting the T_{SL} parameter in the master nodes. Additionally, high values for t_{st} (and therefore for T_{SL}) may result in an inadmissibly low responsiveness to failures.

3.6. *The media adaptation solution*

An intuitive solution to this problem relies on delaying request frames by inserting additional idle time between every transmitted frame, in master nodes [20]. This is depicted in the timing diagram of Fig. 11, where t_{st4} is significantly reduced when compared to the scenario of Fig. 10, if ES1 inserts additional idle periods (t_{ID1+}) before issuing request frames.

Importantly, this mechanism relies on standard features of the PROFIBUS protocol – the idle time parameters. The detailed reasoning and methodologies to compute all network parameters (e.g. the minimum values for t_{ID1} and t_{ID2}, the worst-case

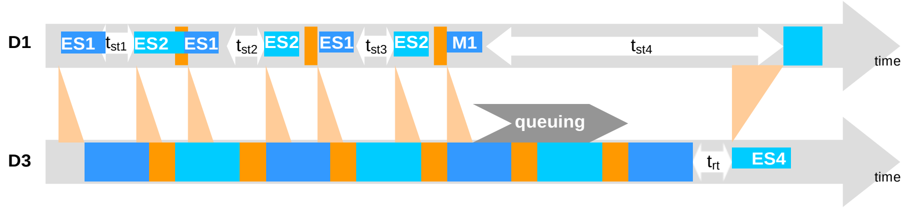


Fig. 10. The unbounded queuing delays problem.

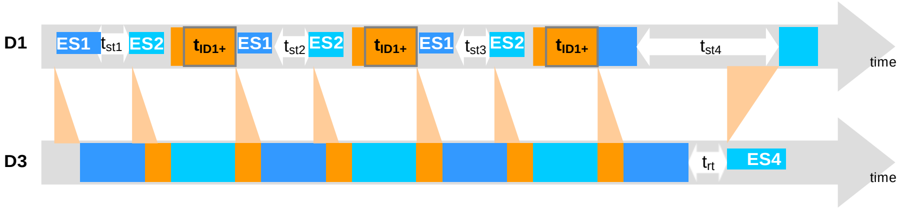


Fig. 11. The media adaptation solution – inserting extra idle times.

system turnaround times – t_{st}, the worst-case duration of message transactions – C and the slot time parameter – T_{SL}) can be found in [20]. In this paper, we focus on the methodology to compute and set the mobility management parameters, which is outlined in Section 4.

The broadcast nature of this hybrid architecture network leads to lower responsiveness to failures (compared to traditional single segment networks). Nevertheless, our methodology for inserting idle time reduces this problem to a minimum, since T_{SL} will be potentially smaller (if errors occur, retransmissions are undertaken sooner). Note that the insertion of these inactivity times reduces network throughput, if most of the message transactions are between nodes belonging to the same network domain. However, eliminating unpredictable delays is mandatory for the real-time operation of the system. Additionally, this repeater-based architecture provides no error containment, e.g. if a token is corrupted, it will propagate to all network domains.

These problems triggered the development of an alternative solution based on IS operating as bridges (Layer 2), leading to a "multiple logical ring" network (e.g. in [21]). The previously referred problems are mitigated, at the cost of a higher system complexity. A comparative performance analysis between the two approaches is provided in [22].

4. Computation of the mobility management parameters

4.1. Initial assumptions

While the basics of the mobility management mechanism adopted for the addressed hybrid wired/wireless network were already presented in Section 3.3, this section presents a methodology to set the mobility management parameters.

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) frame is a PROFIBUS SDN (unacknowledged request) frame, the idle time to be inserted by the MobM after transmitting the BT frame must be implemented using T_{ID2} (refer to Section 2.4).

Obviously, the value of T_{ID2} is related to the worst-case duration of the mobility management procedure (which occurs periodically). The duration of this mobility management procedure depends on the number of beacons that each Structuring Intermediate System must transmit, after having received (and relayed) the BT frame. The number of beacons transmitted by each Structuring IS can be different, since they can receive the BT frame at different time instants (depending on the number of ISs and on the Physical Media in the path between the MobM and each Structuring IS). The way to compute the appropriate number of beacons to be transmitted by each Structuring IS is also a focus of this section.

The mobility management mechanism imposes that the MobM cannot be a mobile ES nor belong to a mobile Wired Domain, i.e. the relative physical position between the MobM and the Structuring ISs in the network cannot change. The reason for this is that the mobility management parameters (number of beacons for each Structuring IS and idle time inserted by the MobM) are computed and set prior to run-time. Therefore, there can be no changes in the path between the MobM and the Structuring ISs.

The mobility management functionality can be under the responsibility of a dedicated master ES or can be integrated in a normal master ES. In the former case, the MobM just has to transmit the BT frame, thus there is the guarantee that the MobM will pass the token to its successor, after issuing the BT frame (and waiting a predefined idle time – T_{ID2}). In the latter case, the BT frame coexists with the other message streams of that master ES.

For the remainder of this section, where the reasoning for the computation of the mobility management parameters will be presented, we will assume the network scenario depicted in Fig. 5.

4.2. Preliminary value for the mobility management duration (t^0_{mob})

The MobM must insert an appropriate idle time before passing the token (in case the MobM is a dedicated master) or issuing another request frame (in case the MobM is a normal master), in order to guarantee that the last mobile ES to receive the BT frame still has enough time (listening to a sufficient number of beacons) to perform channel assessment and switching.

This value for the idle time is roughly the sum of the worst-case latency of the BT frame (t_{bt}) and the worst-case duration of the handoff procedure (t):

$$t'_{\mathrm{mob}} = t_{\mathrm{bt}} + t_{\mathrm{ho}}. \tag{8}$$

Fig. 12 illustrates an example of the value of t^0_{mob}.

The mobility management duration must be computed for every structured Wireless Domain, in order to obtain the worst-case value, which is just a preliminary value for the worst-case mobility management duration. As mentioned before, this result must be readjusted depending on the (integer) number of beacons required to be sent by each Structuring IS. This will be explained later on in detail in Section 4.5.

4.3. Worst-case latency of the BT frame (t_{bt})

The BT frame takes a time interval t_{bt} to reach a specific structured Wireless Domain, which is given by

$$t_{\mathrm{bt}} = Q + t_{\mathrm{btn}}, \tag{9}$$

where Q is the maximum queuing delay affecting the BT frame, from the Domain of the MobM (D1, for the example considered) to the structured Wireless Domain being considered (D4, for the same example).

The second component (t_{btn}) represents the latency of the BT frame along the previously referred path, without considering the queuing delay, and is given by

$$\begin{cases} t_{\mathrm{btn}} = \mathrm{Sum\_}t_{\mathrm{srBT}} + C^{\mathrm{ndp}}_{\mathrm{BT}} - C^{1}_{\mathrm{BT}} \\ \mathrm{Sum\_}t_{\mathrm{srBT}} = \sum\limits_{j=1}^{\mathrm{ndp}-1} \left(t^{j\to j+1}_{\mathrm{srBT}} + t_{\mathrm{sd}}\right) \end{cases} \Rightarrow t_{\mathrm{btn}} = \sum_{j=1}^{\mathrm{ndp}-1} \left(t^{j\to j+1}_{\mathrm{srBT}} + t_{\mathrm{sd}}\right) + C^{\mathrm{ndp}}_{\mathrm{BT}} - C^{1}_{\mathrm{BT}}, \tag{10}$$

with definition of parameters as described in Table 1.

The computation of Q (the worst-case queuing delay affecting the BT frame) depends on the role of the MobM, i.e. if the MobM is a normal master (coexisting with other message streams in that master) or if the MobM is a master exclusively dedi-

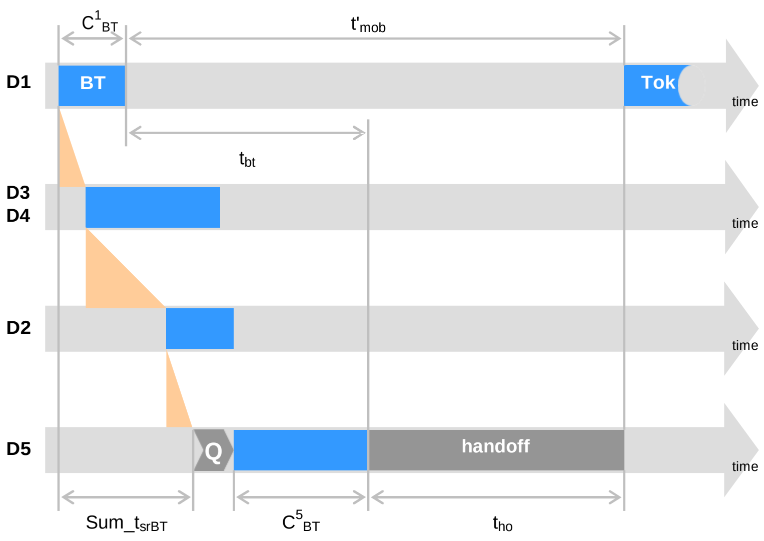


Fig. 12. Timing diagram illustrating the value of t^0_{mob}.

Table 1
Notation and description of relevant parameters

Notation	Description
ndp	Number of Domains in the path between the MobM and an ES in the structured Wireless Domain that is being considered
$C_{\text{BT}}^{\text{ndp}}$	Duration of the BT frame in Domain ndp (structured Wireless Domain being considered)
C_{BT}^{1}	Duration of the BT frame in Domain 1 (location of the MobM)
Sum_t_{srBT}	Total latency of the BT frame along the path from the MobM until the structured Wireless Domain that is being considered
$t_{\text{srBT}}^{j,j+1}$	Start relaying instant of the BT frame, from Domain j to Domain j+1
t_{rd}	Relaying delay of an IS (assumed equal for all ISs)

cated to managing mobility. However, taking into account the scope of this paper, the methodology to compute Q is not presented (refer to [20]).

4.4. Worst-case duration for the handoff procedure (t_{ho})

Consider again the example network depicted in Fig. 5. Each structured Wireless Domain (D3, D4 and D5) operates in a different radio channel (CH1, CH2 and CH3, respectively). Fig. 13 exemplifies a timing diagram for the mobility management procedure (queuing delays are not considered, as they are not relevant for the computation of the worst-case duration of the handoff procedure), assuming ES6 as the mobile ES for which t_{ho} is being computed.

The worst-case duration of the handoff procedure (t_{ho}) is computed as follows. It is assumed that a mobile ES (ES6) starts the handoff procedure immediately after receiving the BT frame, beginning the assessment of its current radio channel (CH3, in the example). After that, the ES switches to another radio channel (CH2) and does the assessment, switches to the other radio channel (CH1) and does the assessment, and finally switches to the radio channel with the best quality (not shown in the timing diagram of Fig. 13, since it is irrelevant for the addressed computation).

The time elapsed in the assessment of the radio channel ($t_{\text{ass}}^{\text{curr}}$ in Fig. 13) currently being used by IS6 (CH3, in the example) is

$$t_{\text{ass}}^{\text{curr}} = t_{\text{bgap}} + C_{\text{beacon}}, \qquad (11)$$

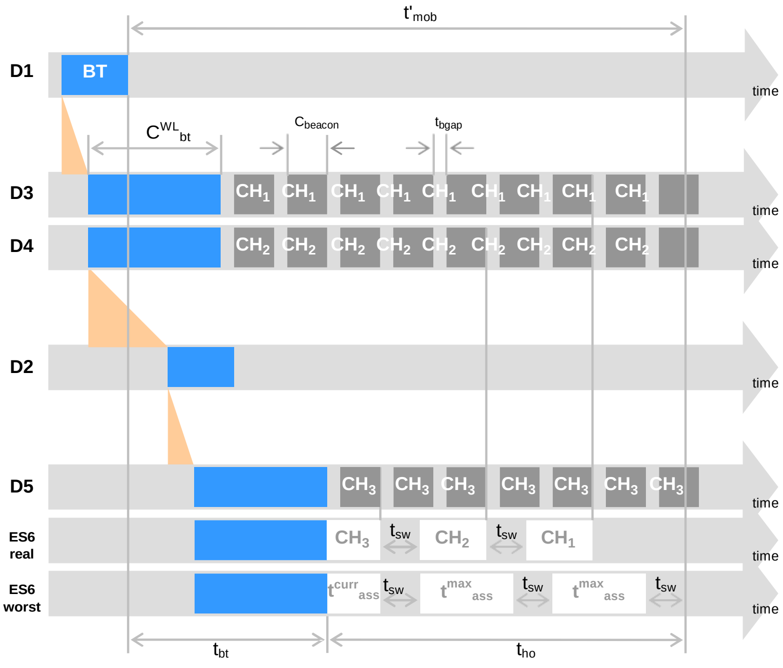


Fig. 13. Mobility management timing diagram.

where C_{beacon} is the duration of a beacon (physical layer frame) and t_{bgap} is the time interval between beacons.

As mobile ESs belonging to a certain structured Wireless Domain start assessing their current radio channel immediately after the complete reception of the BT frame, the correspondent Structuring IS starts transmitting beacons immediately after having (completely) relayed the BT frame. As a result, the Structuring IS and the associated mobile ESs are synchronised, at the beginning of the beacon period. Therefore, in Eq. (11), $t_{\mathrm{ass}}^{\mathrm{curr}}$ is just the sum of the beacon gap (t_{bgap}) and the duration of a beacon (C_{beacon}).

Oppositely, there is no guarantee of synchronisation between mobile ESs in different Wireless Domains, since these may receive the BT frame at different time instants (as in this case). Considering the worst-case situation when assessing CH1 and CH2, i.e. the mobile station (ES6) starts assessing the channel immediately after the beginning of the beacon. Therefore, the maximum assessment duration for each of those channels is

$$t_{\mathrm{ass}}^{\mathrm{max}} = 2 \cdot C_{\mathrm{beacon}} + t_{\mathrm{bgap}}. \tag{12}$$

Considering that the number of radio channels (to assess) is denoted as *nch* and the switching time is defined as t_{sw}, the maximum duration of the handoff procedure in the mobile ES can be computed as

$$t_{\mathrm{ho}} = t_{\mathrm{ass}}^{\mathrm{curr}} + t_{\mathrm{sw}} + (nch - 1) \cdot \left(t_{\mathrm{ass}}^{\mathrm{max}} + t_{\mathrm{sw}}\right) \Longleftrightarrow$$
$$t_{\mathrm{ho}} = (2 \cdot nch - 1) \cdot C_{\mathrm{beacon}} + nch \cdot \left(t_{\mathrm{bgap}} + t_{\mathrm{sw}}\right) \tag{13}$$

For the particular case under consideration, three different radio channels are used ($nch = 3$), resulting in a worst-case duration of the handoff procedure of:

$$t_{\mathrm{ho}} = 5 \cdot C_{\mathrm{beacon}} + 3 \cdot t_{\mathrm{bgap}} + 3 \cdot t_{\mathrm{sw}}. \tag{14}$$

Then, the (preliminary) maximum duration of the mobility management period can be computed as (recalling Eq. (8)):

$$t'_{\mathrm{mob}} = t_{\mathrm{bt}} + t_{\mathrm{ho}}, \tag{15}$$

where t_{bt} and t_{ho} are given by Eqs. (9) and (13), respectively.

4.5. Number of beacons for each Structuring IS (n_b)

After having computed a preliminary value for the mobility management duration (t^{0}_{mob}), it is possible to determine the number of beacons that each Structuring IS must transmit. Remember that a Structuring IS starts transmitting beacons upon having received and (completely) relayed a BT frame.

In order for the mobility management procedure to work properly, each Structuring IS must know the exact number of beacons it must transmit (this number can vary for the different Structuring ISs, as in the example under consideration). This is a parameter which is set in the Structuring ISs. Moreover, considering that the beacon transmission is non pre-emptive (i.e., once a Structuring IS starts transmitting a beacon, it must complete the transmission until the end), there will be the need to adjust the preliminary mobility management period (as given by Eq. (15)).

For every structured Wireless Domain in the network, the following approach is then used:

1. compute the preliminary duration of the beacon period (t^{0}_{bp});
2. compute the number of beacons that must be transmitted by the correspondent Structuring IS (n_{b});
3. re-compute the beacon period duration (t_{bp}) for that structured Wireless Domain;
4. compute the mobility management duration (t_{mob}) for that structured Wireless Domain.

Obviously, the maximum value between the mobility management duration of all Structuring ISs will be chosen as the mobility management duration for the network. Consequently, this value will be used to set the Idle Time parameter T_{ID2} in the MobM.

4.5.1. Computing the preliminary duration of the beacon period – t^{0}_{bp}

The preliminary (maximum) duration of the beacon period, for a particular Structuring Wireless Domain can be computed as follows:

$$t'_{\mathrm{bp}}(D) = t'_{\mathrm{mob}} - t_{\mathrm{btn}}(D). \tag{16}$$

The reason why t_{btn} (latency of the BT frame assuming no queuing) is used instead of t_{bt} (latency with queuing) is that a maximum value for the beacon period must be computed. As can be derived from Eq. (16), the maximum value for t^{0}_{bp} occurs for the lowest BT frame latency, which is t_{btn}. Fig. 14 depicts the timing diagram for the case of D3.

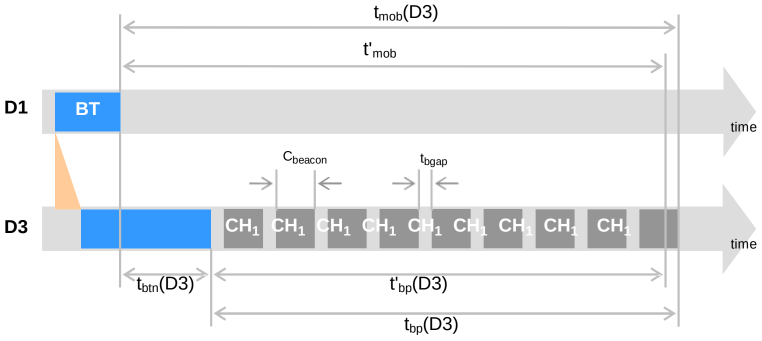


Fig. 14. Example for the number of beacons (D3).

4.5.2. Computing the number of beacons – n_b

The number of beacons that a particular Structuring IS must issue (n_b), in a structured Wireless Domain (D) will be given by the following ceiling (d e) function:

$$n_b(D) = \left\lceil \frac{t'_{bp}(D)}{t_{bgap} + C_{beacon}} \right\rceil. \qquad (17)$$

Note that for the case expressed in Fig. 14, the number of beacons is 9.

4.5.3. Re-computing the beacon period duration – t_{bp}

Now, it is necessary to re-compute the beacon period duration for the structured Wireless Domain, taking into account the real number of beacons that must be transmitted by the corresponding Structuring IS, as given by Eq. (17):

$$t_{bp}(D) = n_b(D) \cdot (t_{bgap} + C_{beacon}). \qquad (18)$$

From Fig. 14, it is clear that t_{bp}(D3) is greater than t^{0}_{bp}ðD3Þ.

4.5.4. Computing the mobility management duration – t_{mob}

As a consequence, the worst-case mobility management duration due to that structured Wireless Domain (D) must also be computed, i.e.

$$t_{mob}(D) = t_{bt}(D) + t_{bp}(D). \qquad (19)$$

In this case, t_{bt} is considered, i.e. the latency of the BT frame including the maximum queuing delay (*Q*), since a worst-case value for the mobility management duration is envisaged. Referring to the example depicted in Fig. 14, it can be seen that t_{mob}(D3) is greater than the preliminary mobility management duration t^{0}_{mob}.

4.6. Setting the idle time parameter (T_{ID2}) in the MobM

After having computed the mobility management duration for all structured Wireless Domains (and the number of beacons for all the corresponding Structuring ISs), it is necessary to determine the maximum between them, i.e.

$$t_{mob} = \max\{t_{mob}(D)\}, \quad \forall D \text{ in the network}. \qquad (20)$$

Finally, the idle time parameter T_{ID2} in the mobility master should be set to a (minimum) value of:

$$T_{\mathrm{ID2}} = \lceil t_{\mathrm{mob}} \cdot r_{\mathrm{MobM}} \rceil, \qquad (21)$$

where r_{MobM} represents the bit rate in the physical medium of the mobility master.

Note that this value for the idle time parameter is the minimum value that still guarantees that there will be no collisions/jamming when the MobM resumes normal operation (transmits a token or a request frame). In this way, the overhead of the mobility management procedure is minimized.

4.7. Additional remarks

4.7.1. Location of the mobility master

The mobility management duration depends on the Domain where the mobility master is located. In order to optimise the performance/throughput of the network, this duration should be minimised at system design phase (pre-run-time). Therefore, the mobility management duration should be computed for different locations of the mobility master (Wired or Wireless Domains), in order to get its optimal (minimum) value. As a rule of thumb, the mobility master should be located in a "central" Domain (from the perspective of all the structured Wireless Domains in the network), in order to "balance" the latency of the BT frame from the MobM until the Structuring ISs.

4.7.2. Priority of the BT frame

Considering the PROFIBUS message dispatching algorithm [25], if a master ES receives a late token, it is only allowed to transmit one high priority message. Therefore, in order to guarantee the proper operation of the network, it is advisable to set the BT frame as a high priority message (SDN) and to use a dedicated master ES as the MobM (no additional message streams in the master). This guarantees the transmission of the BT frame upon reception of the token (and if the mobility management timer has expired).

5. Case study

In this section, we will present a case study based on the network scenario depicted in Fig. 5, highlighting how the mobility management parameters can be computed and set in a practical example. It is assumed that the mobility manager (MobM) is located in D1 (e.g. ES1). All the figures were computed using a system planning tool that implements the algorithms proposed and described in [20]. Note that this software tool was already used for engineering a real application, in the scope of the RFieldbus project [30].

5.1. Defining values for some network parameters

A significant number of network parameters are required by the system planning tool. However, and for the sake of simplicity, we will only address the parameters that are most relevant for the mobility management mechanism.

According to the physical media model that was outlined in Section 3.4, the physical media of the Wired and Wireless Domains are defined by the parameters presented in Table 2 (reasoning for the values chosen for these parameters can be found in [20]). Additionally, each DLL character is assumed to have 8 data bits (at the Data Link Layer).

Table 2
Physical layer parameters

	Wired domains	Wireless domains
Bit rate – r (Mbit/s)	$r^{WR} = 1.5$	$r^{WL} = 2$
Head – l_H (bits)	l_H^{WR} ¼ 0	l_H^{WL} ¼ 200
Tail – l_T (bits)	l_T^{WR} ¼ 0	l_T^{WL} ¼ 0
Overhead per char – k (bits/char)	$k^{WR} = 3$	$k^{WL} = 0$
Offset – o (bits)	$o^{WR} = 33$	$o^{WL} = 150$

Table 3
Mobility-related parameters

Parameter description	Symbol	Value
Length of the BT frame	L_{BT} (chars)	10
Number of radio channels	*nch*	3
Duration of the beacon frame	C_{beacon} (1s)	100
Beacon gap	t_{bgap} (1s)	25
Channel switching time	t_{sw} (1s)	100

Table 3 presents the values for parameters that are relevant for the mobility management mechanism. The BT frame is assumed to be a variable length frame with 1 data octet, corresponding to 10 DLL characters. The duration of the BT frame can be computed using Eq. (6), resulting in 73.3 1s for the Wired Domains and in 140 1s for the Wireless Domains.

5.2. Computation of the mobility management parameters

Table 4 summarises some of the most relevant Mobility Management parameters for each Structuring IS in the network (*IS1*, *IS2* and *IS4*). These values were computed using the previously referred software tool and the way to compute some of them is presented throughout this subsection.

IS1 and *IS2* have equal values for all the parameters, since the path between the MobM and the correspondent structured Wireless Domains (D3 and D4, respectively) is equal (frames are transmitted through identical physical media).

The maximum duration of the handoff procedure (t_{ho}) can be computed using Eq. (13), considering *nch* = 3:

$$t_{ho} = 5 \cdot 100 + 3 \cdot 25 + 3 \cdot 100 = 875\ \mu s.$$

The preliminary (worst-case) duration of the mobility management procedure (t^{0}_{mob}) can be computed for every structured Wireless Domain using Eq. (8). Note that the worst-case latency of the BT

Table 4
Parameters for mobility management

IS	t_{bt} (1s)	t^{0}_{mob} (1s)	t^{0}_{bp} (1s)	n_b	t_{bp} (1s)	t_{mob} (1s)
IS1	113.(6)	988.(6)	1051	9	1125	1238.(6)
IS2	113.(6)	988.(6)	1051	9	1125	1238.(6)
IS4	289.(6)	1164.(6)	875	7	875	1164.(6)

frame from the MobM until a specific structured Wireless Domain (t_{bt}) can be computed using Eqs. (9) and (10).

Therefore, t^{0}_{mob} for each structured Wireless Domain can be computed using Eq. (15) (not shown in Table 4):

$$t'_{\text{mob}}(\text{D3}) = t'_{\text{mob}}(\text{D4}) = 113.(6) + 875 = 998.(6)\ \mu\text{s},$$
$$t'_{\text{mob}}(\text{D5}) = 289.(6) + 875 = 1164.(6)\ \mu\text{s}.$$

Since the maximum value between them is the one to be considered, the preliminary value for the mobility management duration will be:

$$t'_{\text{mob}} = 1164.7\ \mu\text{s}.$$

The preliminary (worst-case) duration of the beacon period (t^{0}_{bp}), for D3/D4 and D5 can be computed using Eq. (16) as follows:

$$t'_{\text{bp}}(\text{D3}) = t'_{\text{bp}}(\text{D4}) = 1164.(6) - 113.(6) = 1051\ \mu\text{s},$$
$$t'_{\text{bp}}(\text{D5}) = 1164.(6) - 289.(6) = 875\ \mu\text{s}.$$

The number of beacons that *IS1* and *IS2* must issue, in structured Wireless Domains D3 and D4 (n_{b}(D3), n_{b}(D4)) can then be computed using Eq. (17) as follows:

$$n_{\text{b}}(\text{D3}) = n_{\text{b}}(\text{D4}) = \left\lceil \frac{1051}{25 + 100} \right\rceil = 9.$$

Re-computing the beacon period duration for D3 and D4 using Eq. (18), results in:

$$t_{\text{bp}}(\text{D3}) = t_{\text{bp}}(\text{D4}) = 9 \cdot 125 = 1125\ \mu\text{s}.$$

Finally, the worst-case mobility management duration due to structured Wireless Domains D3 and D4 can be computed using Eq. (19), as follows:

$$t_{\text{mob}}(\text{D3}) = t_{\text{mob}}(\text{D4}) = 113.(6) + 1125 = 1238.(6)\ \mu\text{s}.$$

The number of beacons that *IS4* must issue, in structured Wireless Domains D5 (n_{b}(D5)) can also be computed using Eq. (17):

$$n_{\text{b}}(\text{D5}) = \left\lceil \frac{875}{25 + 100} \right\rceil = 7.$$

Re-computing the beacon period duration for D5 (using Eq. (18)), results in:

$$t_{\text{bp}}(\text{D5}) = 7 \cdot 125 = 875\ \mu\text{s}.$$

Finally, the worst-case mobility management duration due to structured Wireless Domain D5 can be computed using Eq. (19), as follows.

$$t_{\text{mob}}(\text{D5}) = 289.(6) + 875 = 1164.(6)\ \mu\text{s}.$$

After having computed the mobility management duration for all structured Wireless Domains (and the number of beacons for all the corresponding Structuring Intermediate Systems), the maximum duration of the mobility management procedure is (Eq. (20)):

$$t_{\text{mob}} = \max\{1238.(6), 1238.(6), 1164.(6)\}$$
$$= 1238.(6)\ \mu\text{s}.$$

Finally, the idle time parameter T_{ID2} for the MobM should be set to (Eq. (21)):

$$T_{\text{ID2}} = \lceil 1238.(6) \times 1.5 \rceil = 1858\ \text{bits}.$$

Note that the bit rate – r_{MobM} in Eq. (21) – is equal to 1.5 Mbit/s, since the MobM belongs to a Wired Domain (D1).

It is shown in [28] that the mobility management mechanism should be triggered every 1 second, taking into account several assumptions, such as a maximum linear speed for any mobile ES of 20 km/h. This case study shows that the overhead for mobility management is just slightly above 0.1% (around 1.24 ms of idle communications every 1 s), therefore causing a minor impact on network throughput.

6. Conclusion

There has been an increasing trend for enabling industrial automation systems with wireless communication capabilities. The emerging use of mobile devices in industrial automation applications and the advent of industrial multimedia applications are pushing forward the need for wireless communications. Additionally, cabling costs start to be significant, against the decreasing cost of wireless communication COTS (Commercial Off-The-Shelf) technologies.

Within this context, this paper addresses the extension of the PROFIBUS protocol to encompass wireless/mobile nodes. We outline the major aspects of a hybrid wired/wireless PROFIBUS-based architecture, where most of the design options were made in order to guarantee the real-time behaviour of the overall network. We also outline the timing unpredictability problems resulting from the co-existence of heterogeneous physical media in the same network.

However, the major focus of this paper is on how to guarantee real-time communications in such a hybrid network, where PROFIBUS nodes (and whole segments) can move between different radio cells. Assuming a simple mobility management mechanism based on mobile nodes performing periodic radio channel assessment and switching, we propose a methodology to compute values for some parameters that enable an optimal (minimum) and bounded duration for the handoff procedure.

The mobility management mechanism assumed for the approached architecture was developed within the RFieldbus Project (IST-1999-11316). It was firstly described in a project internal document [28] and then published in [29,20]. The methodologies presented in this paper were already applied in the Manufacturing Automation Field Trial of the RFieldbus project [3], where a system planning tool was developed and used to compute all the necessary PROFIBUS and mobility management parameters [30].

As a conclusion, we show that the proposed mobility management mechanism permits to fulfill stringent real-time requirements, since it is restricted to a reduced, well-determined and bounded period of time. Additionally, we show that the impact of the mobility management mechanism on network throughput is insignificant (around 0.1% overhead), for a typical network scenario such as the one considered in the case study presented in Section 5.

References

[1] PROFIBUS official WWW site: <http://www.profibus.com>.

[2] WWW site of the RFieldbus project: <http://www.hurray.isep.ipp.pt/rfieldbus>.

[3] WWW site of the Manufacturing Automation field trial of the RFieldbus project: <http://www.hurray.isep.ipp.pt/rfpilot>.

[4] Wireless LAN Medium Access Control and Physical Layer Specification – 802.11b, IEEE Standard Board, USA, 1999.

[5] Wireless Medium Access Control (MAC) and Physical Layer (PHY), Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), LAN/MAN Standards Committee of the IEEE Computer Society, USA, 2002.

[6] IEEE 802.15.4 Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Technology, IEEE-SA Standards Board, 2003.

[7] ZigBee Alliance, ZigBee specification, June 2004. Available for download at <http://www.zigbee.org>.

[8] P. Morel, R. Muralt, J.-D. Decotignie, A wireless extension for fieldbus, in: Proceedings of the Second International Conference on Industrial Automation, Nancy, France, 1995, pp. 275–279.

[9] P. Morel, Mobility in MAP networks using the DECT wireless protocols, in: Proceedings of the First IEEE International Workshop on Factory Communication Systems (WFCS'95), Leysin, Switzerland, 1995, pp. 145–149.

[10] S. Cavalieri, D. Panno, On the integration of fieldbus traffic within IEEE 802.11 wireless LAN, in: Proceedings of the Second IEEE International Workshop on Factory Communication Systems (WFCS'97), Barcelona, Spain, 1997, pp. 131–138.

[11] A. El-Hoiydi, P. Dallemagne, Influence of roaming on real-time traffic in wireless networks, in: WIP Session of the Third IEEE International Workshop on Factory Communication Systems (WFCS'00), Porto, Portugal, 2000.

[12] K. Lee, S. Lee, Integrated network of PROFIBUS-DP and IEEE 802.11 Wireless LAN with hard real-time requirement, in: Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'01), Pusan, Korea, 2001, pp. 1484–1489.

[13] S. Lee, K. Lee, F. Harashima, M. Lee, Integration of mobile vehicles for automated material handling using profibus and IEEE 802.11 Networks, IEEE Transactions on Industrial Electronics 49 (3) (2002).

[14] A. Willig, Analysis of the PROFIBUS token passing protocol over error prone links, in: Proceedings of the 25th Annual Conference of the IEEE Industrial Electronics Society (IECON'99), San Jose, USA, 1999, pp. 1246–1252.

[15] A. Willig, A. Wolisz, Ring stability of the PROFIBUS token passing protocol over error prone links, IEEE Transactions on Industrial Electronics 48 (5) (2001) 1025–1033.

[16] A. Willig, Polling-based MAC protocols for improving real-time performance in a wireless profibus, IEEE Transactions on Industrial Electronics 50 (4) (2003) 806–817.

[17] D. Miorandi, S. Vitturi, A wireless extension of PROFIBUS DP based on the bluetooth system, Journal of Computer Communications 27 (10) (2004) 946–960.

[18] D. Miorandi, S. Vitturi, Hybrid wired/wireless implementations of PROFIBUS DP: a feaseability study based on ethernet and bluetooth, Elsevier Computer Communications Journal 27 (2004) 946–960.

[19] J.-D. Decotignie, Interconnection of wireline and wireless fieldbusses, in: The Industrial Information Technology Handbook, Industrial Electronics Series, CRC Press, 2005, ISBN 0-8493-1985-4.

[20] M. Alves, Real-Time Communications over Hybrid Wired/Wireless PROFIBUS-Based Networks, Ph.D. thesis, University of Porto, Portugal, 2003. Available for download at <http://www.hurray.isep.ipp.pt/>.

[21] L. Ferreira, A Multiple Logical Ring Approach to Real-time Wireless enabled PROFIBUS Networks, Ph.D. thesis, University of Porto, Portugal, 2005. Available for download at <http://www.hurray.isep.ipp.pt/>.

[22] P. Sousa, L. Ferreira, M. Alves, Repeater vs. bridge-based hybrid wired/wireless profibus networks: a comparative performance analysis, in: 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'06), Prague, Czech Republic, 20–22 October 2006.

[23] D.A. Roberts, OLCHFA a distributed time-critical fieldbus, in: Proceedings of IEE Colloquium in Safety Critical Distributed Systems, London, United Kingdom, 1993, pp. 6/1–6/3.

[24] M. Lawton, A new wave in industrial communication, 2001. <http://evolution.skf.com>.

[25] EN50170, General Purpose Field Communication System, vol. 2 – PROFIBUS, European Norm, 1996.

[26] ISO 7498, Information Processing Systems – Open Systems Interconnection – Basic Reference Model, 1984.

[27] E. Tovar, F. Vasques, Cycle time properties of the PROFIBUS timed token protocol, Computer Communications 22 (13) (1999) 1206–1216.

[28] C. Koulamas, A. Lekkas, G. Kalivas, S. Koubias, K. Roether, G. Hammer, M. Alves, L. Ferreira, E. Tovar, F. Vasques, Mobility Management in RFieldbus. White Paper of the RFieldbus Project. Version 1.0, January 2001.

[29] M. Alves, E. Tovar, F. Vasques, G. Hammer, K. Röther, Real-time communications over hybrid wired/wireless PROFIBUS-based networks, in: Proceedings of the 14th Euromicro Conference on Real-Time Systems – ECRTS'02, Vienna, Austria, 2002, pp. 142–150.

[30] S. Behaeghel, K. Nieuwenhuyse, L. Marques, M. Alves, E. Tovar, Engineering hybrid wired/wireless fieldbus networks – a case study, in: Proceedings of the Second International Workshop on Real-Time LANs in the Internet Age (RTLIA'03), Porto, Portugal, July 2003, 2003, pp. 111–114.

[31] J. Fonseca, E. Martins, L. Almeida , P. Pedreiras, P. Neves, Flexible time-triggered protocol for CAN – new scheduling and dispatching solutions, in: Proceedings of IEEE International Conference on Communications, New Orleans, USA, 2000.

[32] A. Miaoudakis, A. Koukourgiannis, A. Lekkas, G. Kalivas, G. Papadopoulos, M. Kutschenreuter, P. Coston, L. Beziel, F. Pacheco, L. Pinho, H. Pipinis, N. Rebakos, P. Speckmeier, E. Tovar, F. Vasques, Assessment and Selection of the Radio Technology. Deliverable D1.2. RFieldbus project IST-1999-11316, 2000.