How hard is hard real-time communication on field-buses?

Paulo Veríssimo
pjv@di.fc.ul.pt
FC/UL *

José Rufino
ruf@digitais.ist.utl.pt
IST†

Li Ming
minglz@cs.sfu.ca
EPFL‡

Abstract

Continuity of service and bounded and known message delivery latency, are reliability requirements of a number of real-time applications, such as those served by field-buses. The analysis and design of such networks, w.r.t. timing properties, has been based on no-fault scenarios, rather than under a performability perspective. In this paper, we do an analytical study of the inaccessibility of CAN and PROFIBUS. The study reveals that previous studies on the access delays of these field-buses were too optimistic, from a performability viewpoint.

1 Introduction

Continuity of service and bounded and known message delivery latency, are reliability requirements of a number of real-time applications. One area where these attributes are specially sensitive is field-bus communication. Field-buses are expected to exhibit reliable hard real-time behavior, since they convey the information to and from the extremities of a time-critical system: the sensors and actuators.

The reliability of field-buses is worth studying, for with the adequate techniques, one can build reliable, even fault-tolerant, hard real-time systems based on LANs (local area networks) or field-buses [6, 15]. In reliable real-time systems, the fundamental requirement of communications is that there be a bounded and known message delivery latency, in the presence of disturbing factors such as overload or faults. Amongst other factors, this implies the correct evaluation of the timeliness properties of communication, in the presence of those varying conditions, that is, a performability analysis [4]. However, the analysis and design of standard LAN and field-bus networks w.r.t. timing properties, has routinely been based on no-fault scenarios, with very few exceptions [7, 14], rather than under a performability perspective. Even if one excludes solid faults such as physical partitioning, the performability of field-buses in normal operation has a large variance.

One key issue with this regard, is that networks are subject to periods of *inaccessibility*. They derive from incidents in the local area network or field-bus protocol operation (e.g. token loss or entry of stations) that affect non-faulty components, and are often disregarded, leading to failures of expected hard real-time properties of the network. We address this problem under an inaccessibility model [15].

In this paper, we do an analytical study of the inaccessibility of CAN (Control Area Network) and PROFIBUS (Process Field Bus), two of the most important field-buses in use. The study reveals striking figures showing that previous studies on the access delays of these field-buses were too optimistic, from a performability viewpoint. We present a comparison of our previous results for the token-bus, token-ring and FDDI, with those for CAN and PROFIBUS. We show that CAN exhibits shorter inaccessibility than PROFIBUS, and that overall inaccessibility of LANs is worse than that of field-buses.

The paper is organized as follows: Sections 2 and 3 introduce the inaccessibility problem and the CAN and PROFIBUS field-buses. Sections 4 and 5 analyze the accessibility constraints of CAN and PROFIBUS respectively [1]. Section 6 presents the analytical study results. The paper concludes with some remarks about future work.

2 The inaccessibility problem

The problem of inaccessibility was equated for the first time in [16]. We briefly recapitulate it here, the interested reader is referred to the cited work, or [15]. The

*F.C.U.L. - University of Lisboa, Bloco C5, Campo Grande, 1700 Lisboa - Portugal. Tel. +(351) 1 750 00 84 (secretary), Navigators Web page: http://www.navigators.di.fc.ul.pt/.

†Instituto Superior Técnico - Technical University of Lisboa.

‡Ecole Polytechnique de Lausanne. Currently with SFU, Vancouver, Canada.

[1] Section 4 uses extensive portions of a previous analysis of CAN [8], made in a paper that appeared at the CiA conference, a non-copyrighted event.

impact of inaccessibility on real-time communication is the error it introduces in the specifications of timing bounds, such as deadlines, timeouts, etc. Most analyses of message transmission delays or network schedulability concentrate on the queuing delays caused by the studied arrival patterns. They do so while modeling the network as always functioning normally [2, 1, 13].

In consequence, bounds are established that will be violated upon the (even if rare) occurrence of inaccessibility events. These faults temporarily prevent communication, with the effect of increasing the network access delay as seen by one or more stations. They may in consequence lead to the failure of task or protocol timing assumptions and ultimately, to the failure of the hard real-time system.

Let a network be partitioned when there are subsets of the stations which cannot communicate with each other[2]. This is the classical definition, and it is normally attached to a notion of physical disconnection. However, in a "sane" network the occurrence of certain events in its operation— e.g. entry or leave of new stations— or of individual failures— e.g. bit errors; transmitter or receiver glitches; station failures; token loss— produce side-effects on the other stations, which are a subtle form of partitioning, virtual rather than physical. Standard LANs and field-buses have their own means of recovering from these situations, but since this recovery process takes time, the network will exhibit periods where service is not provided to some or all of the stations. In short, an **inaccessibility** fault occurs when a component *temporarily refrains* from providing service, on account of a foreseen (specified) transition in its internal state. In consequence, the next step is to compute the duration of the inaccessibility events that may take place in the operation of the network, which we do in the next sections.

3 A brief introduction to CAN and PROFIBUS

Field-buses are simplified local area networks, from the viewpoint of network dimension, frame size, throughput, protocol generality. On the other hand, they have other services and mechanisms, aimed at achieving hard real-time communication among low-level devices, such as PLCs (programmable logic controllers) and intelligent sensors and actuators.

We decided to analyze the impact of inaccessibility in CAN and PROFIBUS, two of the most important standard field-buses. We assume the reader to be fairly familiar with their operation. In any case, we forward the reader to the relevant standards [17, 18, 19], for details about the protocols.

3.1 CAN

The Controller Area Network (CAN) is a bus with a multi-master architecture [17]. The transmission medium is usually a twisted pair cable. The network maximum length depends on data rate. Typical values are: 40m @ 1 Mbps; 1000m @ 50 kbps. Bit transmission takes two possible representations: *recessive*, which only appears on the bus when all the stations send recessive bits; *dominant*, which needs only to be sent by one station to stand on the bus. This feature is exploited for bus arbitration. Data transmission is subject to a bit-stuffing technique that prevents transmission of more than five consecutive bits of identical polarity. The Controller Area Network [17] is a *carrier sense multi-access with deterministic collision resolution* network: while transmitting a communication object (frame) identifier, each station monitors the bus; if the transmitted bit is recessive and a dominant bit is monitored, the station gives up from transmitting and starts to receive incoming data; the station transmitting the object with the lowest identifier goes through and gets the bus. Two important parameters characterize the basic operation of any CAN network:

- ◇ **Data Rate** - the nominal rate of data signaling, on the bus. It gives a meaning to t_{bit}, the nominal duration of a single bit.
- ◇ **Intermission Field Period** - t_{IFS} - the minimum bus idle period that mandatorily precedes any data or remote frame transmission.

Only four types of frames are used by CAN. Their duration is given in Table 1. Their functions are described next:

- ◇ **Data Frame** - used for dissemination of communication objects.
- ◇ **Remote Frame** - to explicitly request the dissemination of a communication object.
- ◇ **Error Frame** - used for error signaling
- ◇ **Overload Frame** - used to extend inter-frame spacing, upon detection of overload.

The integrity of data and remote frames is checked at each receiver by the first 15-bit of the 16-bit CRC field.

3.2 PROFIBUS

The PROFIBUS is also a bus with a multi-master architecture [18]. The transmission medium is a twisted

[2] The subsets or partitions may have a single element. When the network is completely down, *all* partitions have a single element, since each station can communicate with no one.

Frame	Symbol	Duration (μs)	
		min.	*max.*
Data frame	t_{data}	44.0	132.0
Remote frame	t_{rdata}	44.0	52.0
Error frame	t_{error}	14.0	20.0
Overload frame	t_{oload}	14.0	20.0

Table 1: Duration of CAN frames (Rate: 1Mbps)

pair cable using RS-485 signaling. The network maximum length depends on data rate. Typical values are: 200m @ 500 Kbps; 1200m @ 9.6 kbps. PROFIBUS is inspired from the Token-bus, with a simplified protocol, and some enhancements devoted to improve device-to-device real-time communication. In PROFIBUS, 127 station addresses are available. A station address is comprised between 0 and 126. Address 127 is reserved for broadcast and multi-cast messages. HSA is the Highest Station Address installed.

For our purposes, two important parameters characterize the basic operation of a PROFIBUS network:

- ◇ **Data Rate** - rate of data signaling on the bus. It defines bit and octet times (t_{bit}, t_{oct}), which are used to scale several parameters.
- ◇ **Slot time** - t_{SLOT} - the worst-case time any station must wait for a response or immediate acknowledgment.

Additionally, three important variables and structures for our study are:

- ◇ **Station Distance** - $D_{i,j}$ - from a station S_i with address A_i to a station S_j with address A_j is defined by: $A_j - A_i$ for $A_j > A_i$; or $HSA - A_i + A_j$ for $A_j < A_i$. We use D_{max} to denote the maximum distance between any two adjacent stations in PROFIBUS.
- ◇ **List of Active Stations (LAS)** - a membership table, which identifies the currently alive stations that are members of the logical ring, and their succession order in token passing. It is used in fault recovery, to find out the new successor in the case of station failures.
- ◇ **Gap List (GAPL)** - a list kept at each station, that identifies the status of all stations in *this station*'s GAP. GAP_i is the address range from station S_i to its successor. *Status* is one of `not ready`, `slave station`, or `non-operational`. It is used for ring membership management, for example, new station insertions.

According to the PROFIBUS standard, the slot time is expressed in bit times in the following equation, where t_{TD} is the worst-case end-to-end propagation delay of the physical layer, and t_{SD} is the station delay from the end of a reception to the start of its own transmission (typical values are given in Table 2):

$$t_{SLOT} = 2t_{TD} + t_{SD} + 11t_{bit} \tag{1}$$

Rate (Kbps)	Bit Time $t_{bit}(\mu s)$	Octet T. $t_{oct}(\mu s)$	Station Dly $t_{SD}(\mu s)$	Slot Time $t_{SLOT}(\mu s)$
500	2	16	200	225

Table 2: PROFIBUS parameters for a network length of 500 m (station delay is implementation dependent)

The function of relevant PROFIBUS frames, as well as their duration, is described in Table 3.

Frame	Symbol	Duration (μs)
REQUEST FDL STATUS (check the status of a given station)	t_{RFS}	204
RESPONSE (response of a station to a request)	t_{RESP}	204
TOKEN (token frame)	t_{TK}	160

Table 3: PROFIBUS frames (Rate: 500Kbps)

4 Inaccessibility of CAN

This section will be devoted to a study of CAN inaccessibility. The interested reader is referred to [8], for more details. The analysis is based on the protocol specification in the standard [17]. Its objective is to derive analytical expressions for all the inaccessibility events, show that their durations are bounded, and determine their upper bound. We signal the worst-case expressions with superscript ^{wc}.

4.1 Bit corruption errors

The first class of inaccessibility events we consider is concerned with the corruption of a single bit of a frame, for instance, due to electro-magnetic interference. Transmitter-based and receiver-based error handling mechanisms exist to detect and correct these errors.

Transmitter-based error detection is achieved by listening to the bus while transmitting, and comparing both streams on a bit-by-bit basis. A recessive level issued on the bus, by a given transmitter, can only be heard back as dominant, without that being considered an error, in the following two circumstances: (i) inside the *arbitration field*, meaning the loss of the arbitration process; (ii) during the *acknowledge slot*, meaning at least one station has not detected, so far, an error in the current transmission. All other situations are errors, signaled on the bus by starting the transmission of an error frame at the next bit slot. While the transmission of the damaged frame plus the error frame last, the network is inaccessible.

Corruption of the transmitted bit-stream cannot occur later than the transmission of the last bit of the *end-of-frame* delimiter. The worst-case inaccessibility time due to *single bit errors* is obtained when considering the transmission of data and error frames with maximum size:

$$t^{wc}_{ina \leftarrow berr} = t^{wc}_{data} + t^{wc}_{error} + t_{IFS} \tag{2}$$

Transmitter-based error detection is helpless in the detection of errors affecting only sets of receiving stations. Receiver-based error detection mechanisms remedy this problem. A *bit-stuffing* coding scheme is used in the transmission of data and remote frames, up to the last but one of the 16-bit CRC field. Whenever a receiving station monitors l_{stuff} consecutive bits of identical level, it automatically deletes the next received (stuff) bit. Under error free operation, the deleted bit presents a polarity opposite to the preceding ones; should this condition be violated, an error will be signaled on the bus, by starting the transmission of an error frame.

A *bit-stuffing error* cannot occur after reception of the 16-bit CRC field. The worst-case inaccessibility duration occurs for the transmission of maximum size data and error frames. Having both these aspects in mind, we have:

$$t^{wc}_{ina \leftarrow stuff} = t^{wc}_{data} - t_{EFS} + t^{wc}_{error} + t_{IFS} \tag{3}$$

where t_{EFS}, accounts for the duration of the fixed form sequence – not subject to bit-stuffing coding – that ends every data or remote frame.

On the other hand, bit errors can occur in a way that they do not produce a violation of the stuffing policy. In this case, the error will be detected only when the end of frame sequence is received, either by *CRC checking* or through detection of a frame format violation.

After reception of a frame, a station should do one of two things, depending on whether the CRC was good: (i) if the CRC is correct, changes the bus level from recessive to dominant, during the *acknowledge slot*; (ii) otherwise, it passes the *acknowledge slot*, and signals the CRC error through the transmission of an error frame, that starts immediately after the *acknowledge delimiter*. The corresponding worst-case inaccessibility time due to a *CRC error* occurs for the longest data and error frames, and is given by expression:

$$t^{wc}_{ina \leftarrow crc} = t^{wc}_{data} - t_{EOF} + t^{wc}_{error} + t_{IFS} \tag{4}$$

where t_{EOF} represents the duration of the *end-of-frame* delimiter.

Whenever a transmitter does not monitor a dominant level on the bus during the *acknowledge slot*, it interprets that as an error, and the transmission of an error frame is started at the next bit. So, the duration of the corresponding worst-case inaccessibility due to *acknowledgment error* is given by equation:

$$t^{wc}_{ina \leftarrow ack} = t^{wc}_{data} - t_{EFS} + 2 \,.\, t_{bit} + t^{wc}_{error} + t_{IFS} \tag{5}$$

All the frames used by the CAN protocol obey to a few pre-defined formats. Data and remote frames have a fixed form end-sequence. An error in this sequence implies the transmission of an error frame. The worst-case inaccessibility caused by a *form error* is when it occurs, at the latest, while receiving the last but one bit of the *end-of-frame* delimiter, because a receiver monitoring a *dominant* level at the last bit of this delimiter does not take that as a form error[3]. As usual, the worst-case expression relates to the maximum size data and error frames:

$$t^{wc}_{ina \leftarrow form} = t^{wc}_{data} - t_{bit} + t^{wc}_{error} + t_{IFS} \tag{6}$$

4.2 Overload errors

In this section, we treat overload frames. Existing commercial devices do not implement the standard completely w.r.t. overload frame transmission. Nevertheless, we examined the relevant situations, discovering (cf.§ 6.2) that their impact on inaccessibility is generally small. Due to space reasons, we provide a summarized account of that study, which is detailed in [8].

For correct operation, the CAN protocol requires data and remote frames to be separated from each other and from any other frame by a minimum amount of three recessive bits, known in CAN terminology as *intermission field*. However, the next data or remote frame

[3] As we will see ahead, this will be considered a *reactive overload error*.

transmission can be explicitly delayed, whenever the receiver circuitry is not ready to receive those frames. That is achieved through transmission of a special frame known, in CAN terminology, as *requested overload* frame. The corresponding upper inaccessibility bound is simply given by:

$$t^{wc}_{ina \leftarrow oload} = 2 \,.\, t^{wc}_{oload} \quad (7)$$

Whenever a transmitter or a receiver detects a dominant bit within the *intermission field* or a receiver detects a dominant value in the last bit of the *end-of-frame* delimiter, the transmission of a special type of frame, a *reactive overload* frame, is initiated one bit after the detection. The corresponding worst-case inaccessibility bound will be given by:

$$t^{wc}_{ina \leftarrow roload} = t^{wc}_{oload} + t_{IFS} \quad (8)$$

Any error in the overload frame fixed form triggers the transmission of an error frame at the next bit slot. The corresponding inaccessibility time is then given by:

$$t^{wc}_{ina \leftarrow oform} = t^{wc}_{ina \leftarrow oload} + t^{wc}_{error} \quad (9)$$

Now we consider that only a subset of the stations detect, either correctly or erroneously, a dominant level in the third bit of the *intermission field*. This scenario leads to the most serious inaccessibility situation on account of overload frames.

The set of stations detecting such an event initiate the transmission of a *reactive overload* frame. However, this action will be perceived in a different manner by the set of stations that did not monitor the *intermission field* violation: such stations will interpret the first of the six dominant bits that made up the *overload flag* as a *start-of-frame* delimiter. The sixth dominant bit will violate the bit-stuffing rule and cause a signaling error condition.

For the evaluation of the worst-case inaccessibility time for this scenario, we take the largest of those error detection bounds and additionally assume that the inconsistency in *intermission field* monitoring only occurs after the extension of the *inter-frame space* though two consecutive *requested overload* frames. Therefore:

$$t^{wc}_{ina \leftarrow iload} = 2 \,.\, t^{wc}_{oload} + t^{wc}_{data} - t_{bit} + t^{wc}_{error} + t_{IFS} \quad (10)$$

4.3 Multiple errors

We now extend our single-error analysis to a multiple-error one, by making the following **omission degree assumption**: *no more than n error and/or reactive overload frame transmissions are required to recover from errors in the transmission of a data frame*[4].

We first assume that all the n errors are consecutive in the network[5]. In this case, error signaling will also be subject to disturbances: any station detecting a deviation from the error frame fixed form, starts the transmission of a new error frame. For the worst-case scenario, we consider that the first error occurs at the end of a maximum length data frame transmission and that the following $(n-1)$ errors are only detected at the end of a maximum duration error signaling period. The recovery actions end with the successful transmission of the n^{th} error frame. The worst-case inaccessibility time due to *consecutive errors* is then given by:

$$t^{wc}_{ina \leftarrow mcerr} = t^{wc}_{data} + n \,.\, t^{wc}_{error} + t_{IFS} \quad (11)$$

Let us now assume that errors occur in a way that only data or remote frame transmissions are affected; the corresponding error signaling, always succeeds. This scenario is realistic enough to be considered: a failure in a transmitter may lead to this behavior.

The corresponding worst-case inaccessibility bound directly results from the application of the omission degree assumption, taking into account the worst-case durations of data and error frames:

$$t^{wc}_{ina \leftarrow mserr} = n \,.\, (t^{wc}_{data} + t^{wc}_{error} + t_{IFS}) \quad (12)$$

4.4 Station Failures

The error confinement mechanisms provided by the CAN protocol are based on two different error counters at each station, recording transmit and receive errors. These counters have a non-proportional update method, with an error causing an increment larger than the decrement resulting from a successful data or remote frame transfer. The rules used in error counting have been defined in order that stations closer to the error-locus will experience, with a very high probability, the highest error count increase. This way, disturbances due to a faulty station can be localized and its action restricted, according to the following classes:

Error-Active - the normal operating state; able to transmit and receive frames; fully participates in error detection/signaling.

[4]Under given circumstances, CAN data frame losses are masked-off at the MAC layer, thus rendering the user-level omission degree bound, k, different from the network-level omission degree bound, n.

[5]Meaning they are not interleaved with good transmissions.

Error-Passive - able to transmit and receive frames, but after transmitting a data or remote frame the station is obliged to an extra eight bit bus idle period following the *intermission field*, before it can start a new transmission; enters this mode after any error counter exceeds 127; goes back to error-active when both counters are equal to or lower than 127; only succeeds to signal errors while transmitting.

Bus-Off - does not participate in any bus activity, being unable to send or receive frames; enters this mode after the transmit error counter exceeds 255; only leaves after reset.

In order to complete our analysis of CAN inaccessibility, we proceed with the study of two examples, where the occurrence of permanent failures is assumed.

Let us analyze in first place, a scenario where the transmitter of a given error-active CAN station fails in a way that errors only affect the data and remote frames it sends. Such an error pattern may be due to a malfunction in the station's CRC generator. As we have just seen, *transmitter failure* may only systematically affect bus activity until the station enters the error-passive state. The number of transmission attempts causing this transition is bounded by:

$$n_{txfail} = \left\lceil \frac{err_{a\_thd}}{\Delta_{txerr}} \right\rceil \tag{13}$$

where $\lceil \quad \rceil$ represents the *ceiling* function[6]; err_{a_thd} is the error-active count threshold and Δ_{txerr} is the *Transmit-Error-Count* increment produced by each transmission error [17].

Consecutive transmissions issued by the failed transmitter appear in the network interleaved with error frames and occasionally with good transmissions from other stations. The longest inaccessibility period due to *transmitter failure* occurs in the absence of these last events and is obtained applying expression (13) to equation (12):

$$t^{wc}_{ina \leftarrow txfail} = n_{txfail} \cdot (t^{wc}_{data} + t^{wc}_{error} + t_{IFS}) \tag{14}$$

The second scenario is when an *error-active CAN receiver* is affected by a similar failure, i.e. a malfunction in the CRC checker. Errors are also successive in the network because, once again, only data and remote frame transfers are affected: error signaling occurs without errors. However, no data/remote transfer succeeds until the defective station becomes error-passive, which occurs after the following attempts:

$$n_{rxfail} = \left\lceil \frac{err_{a\_thd}}{\Delta_{rxer1} + \Delta_{rxer2}} \right\rceil \tag{15}$$

where the terms under the fraction represent two different contributions for *Receive-Error-Count* increase. They derive from the application of more than one error counting rule to the same data transfer, as defined in [17]. The worst-case inaccessibility time of the network due to *receiver failure* is thus obtained using expression (15) in equation (12). Thus:

$$t^{wc}_{ina \leftarrow rxfail} = n_{rxfail} \cdot (t^{wc}_{data} + t^{wc}_{error} + t_{IFS}) \tag{16}$$

5 Inaccessibility of PROFIBUS

This section will be devoted to a study of PROFIBUS inaccessibility. The interested reader is referred to [5], for more details. The section follows the same objectives as the previous one: to derive analytical expressions for all the inaccessibility events, show that their durations are bounded, and determine their upper bound. The analysis is based on the protocol specification in the standard.

5.1 Insertion of new stations

The first class of inaccessibility events we consider is concerned with the insertion of new stations, that requires reformation of the logical token-passing ring.

In PROFIBUS, each station in the logical token ring is responsible for the insertion of new stations. For that purpose, each station periodically scans its GAP address range (from its own address to its successor's) looking for stations in that address range that wish to enter. A station checks only one address per token visit, using a control frame (Request FDL Status frame), which may be repeated N_{retry} times, in case of omissions. In consequence, at most one station can enter at a time.

The first inaccessibility situation thus occurs when the MAC sub-layer performs the actions required for the insertion of a new station.

We denote a station S_i with address A_i. We denote $t_{ina \leftarrow join}$ as the total inaccessibility time for the insertion of a station S_j between stations S_i and S_{i+1}. Since a station checks only one address per token visit, it must scan the range from address $A_i + 1$ to address A_j in successive opportunities. So, the inaccessibility periods alternate with the flow of data in the network.

[6] The *ceiling* function $\lceil x \rceil$ is defined as the smallest integer not smaller than x.

The time spent in checking one address is denoted as t_{exam}. The term $D_{i,j}.t_{exam}$ represents the time for checking all addresses in the range, up to A_j. The maximum (worst-case) value of this term is as follows. Recall that D_{max} is the maximum adjacent station distance, so in the worst-case $D_{i,j}$ is equal to $D_{max} - 1$. On the other hand, the worst-case value of t_{exam} occurs when the responder acknowledges or responds only after the last repeated Request FDL Status frame is received. In this case, it is expressed as follows:

$$\begin{aligned} t^{wc}_{exam} &= t_{RFS} + t_{SLOT} + \\ &\quad +(N_{retry} - 1)(t_{RFS} + t_{SLOT}) + \\ &\quad +t_{RFS} + t_{SD} + t_{RESP} \\ &= N_{retry}(t_{RFS} + t_{SLOT}) + \\ &\quad +t_{RFS} + t_{SD} + t_{RESP} \end{aligned} \tag{17}$$

For t_{TK}, t_{RFS} and t_{RESP} see Table 3. The term $(N_{retry} - 1)(t_{RFS} + t_{SLOT})$ represents the time spent in the first N_{retry} - 1 retries. In the last retry, the response is received by return.

There is another term to add to $t_{ina \leftarrow join}$, due to a specificity of the PROFIBUS protocol: a new station is only recognized by its downstream neighbor at the second token pass attempt, after waiting one slot time, that is, an extra $t_{SLOT} + t_{TK}$ delay before the join process is complete.

Hence the worst case total inaccessibility time $t^{wc}_{ina \leftarrow join}$ for the *insertion of a station* is given by:

$$\begin{aligned} t^{wc}_{ina \leftarrow join} &= \max\,(D_{i,j})\; t^{wc}_{exam} + t_{SLOT} + t_{TK} \\ &= (D_{max} - 1)[N_{retry}(t_{RFS} + t_{SLOT}) + \\ &\quad +t_{RFS} + t_{SD} + t_{RESP}] + \\ &\quad +t_{SLOT} + t_{TK} \end{aligned} \tag{18}$$

Let us consider now the case of several stations (K) that attempt to enter the logical ring in the same gap. They will enter one at a time. The GAP space exploration is done once, the token passing procedure is obviously repeated K times. The worst-case total inaccessibility time for *insertion of multiple stations* is given by:

$$\begin{aligned} t^{wc}_{ina-mjoin} &= (D_{max} - 1)[N_{retry}(t_{RFS} + t_{SLOT}) + \\ &\quad +t_{RFS} + t_{SD} + t_{RESP}] + \\ &\quad +K(t_{SLOT} + t_{TK}) \end{aligned} \tag{19}$$

5.2 Token Loss

In this section we consider the case where a station fails while it is holding the token. In Profibus, each station has a `Time-out` timer which is used to monitor bus activity. According to the PROFIBUS standard the `Time-out` in a station S with address A is:

$$t_{time-out} = 6t_{SLOT} + 2A \times t_{SLOT} \tag{20}$$

The second term $2A \times t_{SLOT}$ ensures that no two master stations claim the token at the same time after a token loss has occurred. The time-out expires first in the master station with the lowest address in the network, say A_{low}, because the time-out timers are set to values that are an increasing function of station addresses. Since there is no contention, inaccessibility ends the moment that station times-out, seizes the token and restarts transmitting.

The worst-case inaccessibility time for *token loss* is thus given by the maximum value of $t_{time-out}$, which on the other hand is attained when A_{low} is maximized (A^{wc}_{low}). Suppose that there are N_{st} active stations (masters) in the network. In the worst-case, the lowest address A_{low} is equal to $HSA - N_{st}$. We have:

$$\begin{aligned} t^{wc}_{ina \leftarrow tkloss} &= 6t_{SLOT} + 2A^{wc}_{low} t_{SLOT} \\ &= 6t_{SLOT} + 2(HSA - N_{st})t_{SLOT} \end{aligned} \tag{21}$$

5.3 Station failures

In this section we assume that a failed station is not holding the token. This is detected when the predecessor station tries passing it the token. After a period of inaccessibility, the new successor is found and the token passed to it.

Let us start with the case of a single failed station. The token passing operation is repeated twice before the token holder gives-up, with a slot-time of interval between each attempt. Then, once the new successor identified, the token holder tries passing it the token. As discussed already in section 5.1, a new station is only recognized by its downstream neighbor at the second token pass attempt, after waiting one slot time. The time to recover from *single station failure* is thus given by:

$$\begin{aligned} t^{wc}_{ina \leftarrow sfail} &= 3t_{SLOT} + 3t_{TK} + t_{SLOT} + t_{TK} \\ &= 4(t_{SLOT} + t_{TK}) \end{aligned} \tag{22}$$

Let us now consider a less restrictive scenario by allowing several stations to fail simultaneously. Let us, for the moment, additionally assume that none of the failed stations are adjacent in the token passing order. As a consequence, all the failures can be recovered in a single token rotation. The current token rotation is slowed down by the amount of time needed to perform all recoveries.

The resulting inaccessibility time should be equal to the sum of the time spent in recovering from each failed station. Because we assume that none of the failed stations are adjacent in the token passing order, the time to recover from a failed station can be calculated by Eq. 22. Suppose that there are N_{fail} failed stations. Then, $t_{ina \leftarrow mfail} = N_{fail} \times t_{ina \leftarrow sfail}$.

It is easy to see that the inaccessibility time is proportional to N_{fail}. In the worst case, the maximum number of failed stations is $\lfloor N_{st}/2 \rfloor$, i.e., failed and non-failed stations are completely intermixed. N_{st} is the number of the active stations (faulty plus non-faulty stations) before the multiple station failure. The worst case inaccessibility due to *multiple station failure* is therefore given by:

$$\begin{aligned} t^{wc}_{ina \leftarrow mfail} &= \lfloor N_{st}/2 \rfloor \times t_{ina \leftarrow sfail} \\ &= 4\lfloor N_{st}/2 \rfloor (t_{SLOT} + t_{TK}) \end{aligned} \quad (23)$$

If we relax our failure assumption and allow failure of consecutive stations, token passing will be disrupted. Suppose that N_{fail} adjacent stations in the token-passing order fail. There will be N_{fail} attempts to pass the token until the next healthy station is found and the situation is recovered:

$$\begin{aligned} t_{ina-gfail} &= N_{fail}(3t_{SLOT} + 3t_{TK}) + \\ &\quad + t_{SLOT} + t_{TK} \\ &= (3N_{fail} + 1)(t_{SLOT} + t_{TK}) \end{aligned} \quad (24)$$

We can now generalize our previous result by allowing the existence of several groups of failed stations. The worst-case inaccessibility is generated when the several groups recover one at a time, each taking the time given by equation 24, and when that recovery is lengthier. The worst-case scenario occurs with the highest possible number of groups [5]. Given a group size Gf_{dim}, this happens for $Gf_{dim} = 2$ out of N_{fail} stations. This gives a worst case inaccessibility for *multiple failed groups*, of:

$$\begin{aligned} t^{wc}_{ina-mgfail} &= \sum_{i=1}^{\lfloor N_{st}/3 \rfloor} t_{ina-gfail}(i) \\ &= 7\lfloor N_{st}/3 \rfloor \times (t_{SLOT} + t_{TK}) \end{aligned} \quad (25)$$

6 Analytic results and comparison with other networks

We have analyzed the performability of several LANs before. In this sense, it will be interesting to compare the inaccessibility figures obtained for CAN and PROFIBUS with the ones obtained in our previous studies [9, 11, 12], concerning real-time LANs with the same number of stations. These results are summarized in Table 4 where, for each LAN, we have listed the best and worst-cases. Most of the scenarios are self-explanatory. Active Monitor Loss in the token-ring occurs when the station regenerating the ring's synchronism fails, which leaves the ring completely down until a new monitor is launched. Streaming Receiver failure in the same network occurs when a receiver partially decodes an incoming frame.

Analyzed Scenario	t_{ina} (ms)	
	min.	*max.*
ISO 8002/4 Token-Bus $(5Mbps)$		
Station Join	0.07	4.67
Multiple Joins	0.37	139.99
Station Leave	0.06	0.06
Token Loss	1.74	5.79
Multiple Group Failure	5.70	51.76
ISO 8002/5 Token-Ring $(4Mbps)$		
Stripping Errors	4.6	4.7
Token Loss	14.6	14.7
Active Monitor Loss	1044.9	15068.4
Station Join/Leave	14.6	14.7
Streaming Receiver	28255.2	28278.3
ISO 9314 FDDI $(100Mbps)$		
No Valid Transmissions	2.53	2.76
No Valid Tokens	15.03	15.26
Station Join	30.03	30.24
Station Leave	20.03	20.24
Streaming MAC Receiver	9457.18	9457.33

Table 4: LAN Inaccessibility Times

6.1 PROFIBUS Analysis

Now, we evaluate the inaccessibility time bounds for a typical PROFIBUS network, according to the expressions developed in section 5. The configuration parameters are as follows:

- Network length: 500 m
- Data rate: 500 Kbps.
- HST (Highest Station Address): 64
- The maximum retry number N_{retry} is equal to one

The results of the evaluation, for each one of the studied scenarios, are summarized in Table 5. The second column in the table represents the worst-case figures. From the analysis made and from the table we observe the following points:

- the time spent for token recovery depends on the lowest address in the network. In the worst case, the lowest address is equal to HSA - N_{st}. For our example (HSA = 64; $N_{st} = 32$), the worst-case inaccessibility time is $70T_{SLOT}$ (15.89 ms);

- station entry is simple (compared to the token-bus), because one station enters at a time and thus there is no contention;
- recovery from station failures is very simple because of the use of the LAS (cf.§ 3.2);
- **in PROFIBUS, the worst-case inaccessibility time is bounded by** $75\ ms$.

Scenario	t_{ina}^{wc} (ms)
Single Station Join ($N_{st} = 32$)	32.5
Multiple Station Joins ($N_{st} = 2$, $K = 30$)	74.8
Token Loss ($N_{st} = 32$)	15.89
Single Failed Station	1.54
Multiple Failures ($N_{st} = 32$; $N_{fail} = 16$)	24.64
Station Group Failures ($N_{st} = 32$; $N_{fail} = 16$)	18.86
Multiple Group Failures ($N_{st} = 32$)	29.6

Table 5: Profibus Inaccessibility Times

This contrasts with the Token-bus, where these procedures are more complex and lengthier. Whereas that can be observed to some extent by comparing Tables 4 and 5, the difference is even higher if we evaluate the two networks with a harmonized bit rate. PROFIBUS figures harmonized for 5Mbps, the rate of Token-bus, can be found in [5].

6.2 CAN Analysis

We now evaluate inaccessibility time bounds, for a given CAN network, for example, a 32 station CAN field-bus, in an industrial environment. The data rate is $1Mbps$ and we assume a moderate value ($n = 3$) for the *omission degree* bound due to medium errors. The results of the evaluation, for each one of the studied scenarios, are summarized in Table 6. We see that inaccessibility is on average limited to $500\mu s$, but can reach a high $2.5\ ms$, on account of transmitter or receiver failures.

The results of CAN evaluation are summarized in Table 6. The second column in the table represents the worst-case figures. From the analysis made and from the table we observe the following points:

Data Rate - $1Mbps$	
Scenario	t_{ina}^{wc} (μs)
Bit Errors	155.0
Bit-Stuffing Errors	145.0
CRC Errors	148.0
Form Errors	154.0
Acknowledge Errors	147.0
Overload Errors	40.0
Reactive Overload Errors	23.0
Overload Form Errors	60.0
Inconsistent Overload Errors	194.0
Multiple Consecutive Errors ($n = 3$)	195.0
Multiple Successive Errors ($n = 3$)	465.0
Transmitter Failure	2480.0
Receiver Failure	2325.0

Table 6: CAN Inaccessibility Times

- CAN exhibits the shortest inaccessibility of the real-time LAN or field-buses analyzed;
- this can be attributed to the simplicity of the protocol, and to the cooperation of transmitter and receiver to detect errors immediately that they occur, during the transmission of the affected frame;
- **in CAN, the worst-case inaccessibility time is bounded by** $2.5\ ms$.

6.3 Achieving hard real-time communication

In order to achieve reliable hard real-time communication, the worst-case inaccessibility figures found above must be included in the timeliness equations of the systems under design. That is, be compounded with other sources of delay, such as queuing delays. The mechanisms and techniques for achieving hard real-time communication from application to application are outside the scope of this paper, and are discussed with detail in [16, 15]. For completeness, we just provide a few remarks on the subject.

The crux of the problem is to control inaccessibility or, in other words: to ensure that the number of inaccessibility periods and their duration have a bound; to verify that the bound is suitably low for the service requirements; to accommodate inaccessibility in protocol execution and timeliness calculations.

In our earlier work on LANs we have observed that some inaccessibility situations can be minimized right from the start, by careful tuning of some parameters of the network. We had some success for the Token-bus [10], where we managed a five-fold decrease in the worst-case inaccessibility figure, and more recently for the PROFIBUS, one of the studied field-buses, where the decrease was two-fold [5]. We have not attempted at reducing CAN inaccessibility this way. Recent studies have shown that CAN inaccessibility has very little sensivity to network parameter tuning.

7 Conclusion

In order to achieve reliable real-time operation on a given field-bus network, a bounded delay requirement must be met, in the presence of faults. Inaccessibility faults are a performability problem that is often disregarded by designers expecting hard real-time behavior from a field-bus. We have made a thorough analytical study of CAN and PROFIBUS inaccessibility situations. Furthermore, we derived worst-case figures for the several inaccessibility situations on both field-buses.

The figures showed that previous studies on the access delays of these field-buses were too optimistic, since some inaccessibility situations last far more than commonly accepted transmission delays in industrial applications. Translating our claim into figures, this means that under a reliable communication perspective, there is no trustworthy worst-case frame delivery delay bound below 2.5 ms for CAN, or 75 ms for PROFIBUS, even in idle network situations.

Finally, we compared our previous results for the token-bus, token-ring and FDDI LANs, with those for CAN and PROFIBUS. The better results for field-buses should be evaluated under the light of the expected transmission and round-trip delays, which are also shorter for field-buses. We also found out that CAN exhibits the shortest inaccessibility figures of the networks studied.

As future work, it would be interesting to evaluate, for example with the help of adequate tools, the general performability of CAN and PROFIBUS, under complex fault scenarios where several inaccessibility events can take place [3].

References

[1] Raj Jain. Performance analysis of FDDI token ring networks: effect of parameters and guidelines for setting TTRT. In *Proceedings of the ACM-SIGCOM'90 Symposium*, Philadelphia-USA, September 1990.

[2] D. Janetzky and K. Watson. Token bus performance in MAP and PROWAY. In *Proceedings of the IFAC Workshop on Distributed Computer Protocol System*, 1986.

[3] L. Mahlis, W. Sanders, and R. Schlichting. Analytic performability evaluation of a group-oriented multicast protocol. Technical report, University of Arizona, 1992.

[4] John Meyer. Performability: a retrospective and some pointers to the future. *Performance Evaluation*, 14(3-4):139–156, 1992.

[5] Li Ming. *Real-time communication in an industrial network — Profibus*. PhD Thesis, Dept. of Computer Science, Swiss Federal Institute of Technology, 1996.

[6] D. Powell, editor. *Delta-4 - A Generic Architecture for Dependable Distributed Computing*. ESPRIT Research Reports. Springer Verlag, November 1991.

[7] K.H. Prodromides and W.H. Sanders. Performability evaluation of csma/cd and csma/dcr protocols under transient faults. *IEEE Transactions on Reliability*, 42(1):166–127, March 1993.

[8] J. Rufino and P. Veríssimo. A study on the inaccessibility characteristics of the controller area network. In *Proceedings of the 2nd International CAN Conference*, London, England, October 1995. CiA.

[9] J. Rufino and P. Veríssimo. A study on the inaccessibility characteristics of ISO 8802/4 Token-Bus LANs. In *Proceedings of the IEEE INFOCOM'92 Conference on Computer Communications*, Florence, Italy, May 1992. IEEE. also INESC AR 16-92.

[10] J. Rufino and P. Veríssimo. Minimizing token-bus inaccessibility through network planning and parameterizing. In *Proceedings of the EFOC/LAN92 Conference*, Paris, France, June 1992. IGI. also INESC AR 17-92.

[11] José Rufino and Paulo Veríssimo. A study on the inaccessibility characteristics of ISO 8802/5 Token-Ring LANs. Technical Report RT/24-92, INESC, Lisboa, Portugal, February 1992.

[12] José Rufino and Paulo Veríssimo. A study on the inaccessibility characteristics of the FDDI LAN. Technical Report RT/25-92, INESC, Lisboa, Portugal, March 1992.

[13] K. Tindell and A. Burns. Guaranteeing message latencies on Controler Area Network. In *Proceedings of the 1st International CAN Conference*, Mainz, Germany, September 1994. CiA.

[14] K. Tindell, A. Burns, and A. Wellings. Calculating Controller Area Network (CAN) message response times. In *Proceedings of the IFAC Workshop on Distributed Computer Control Systems*, Toledo, Spain, September 1994. IFAC.

[15] Paulo Veríssimo. Real-time Communication. In S.J. Mullender, editor, *Distributed Systems, 2nd Edition*, ACM-Press, chapter 17, pages 447–490. Addison-Wesley, 1993.

[16] P. Veríssimo, J. Rufino, and L. Rodrigues. Enforcing real-time behaviour of LAN-based protocols. In *Proceedings of the 10th IFAC Workshop on Distributed Computer Control Systems*, Semmering, Austria, September 1991. IFAC.

[17] ISO. *ISO International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network (CAN) for high-speed communication*, November 1993.

[18] Deutsche Industrie Norm, Profibus Standard DIN 19245 Part 1, Beuth-Verlag, Berlin, 1989.

[19] Deutsche Industrie Norm, Profibus Standard DIN 19245 Part 2, Beuth-Verlag, Berlin, 1989.