Asian Journal of Information Technology 6 (7): 806-816, 2007
ISSN: 1682-3915

PROFIBUS Timing Analysis under Transient Faults

[1]Saad Hazim Al-Tak and [2]Sufyan T. Faraj
[1]College of Engineering, University of Baghdad, Iraq
[2]College of Computers, University of Anbar, Iraq

Abstract: Fieldbus applications suffer from severe environmental conditions. These conditions may affect the communication process among different nodes. Because of the real-time nature of such applications, timing behaviour must be well designed and studied. PROFIBUS as one of the widely applied fieldbus protocols is considered here. This study evaluates inaccessibility overheads in PROFIBUS protocol in the presence of transient faults. It introduces a novel analytical model for the inaccessibility of PROFIBUS message/token transmission in the presence of transient faults. Different error scenarios are investigated to produce Best-Case (BC) and Worst-Case (WC) error overhead evaluation that are based on the integration of single bit errors together with burst errors into a bounded fault arrival model. The introduced error components are included in the Worst-Case Response Time (WCRT) of PROFIBUS message cycles. Such a work is essential to assess the real-time behaviour of the protocol under the incidence of errors.

Key words: Bounded fault model, fieldbus, inaccessibility, PROFIBUS, transient faults

INTRODUCTION

Fieldbus networks, as a part of distributed control applications, are subjected to harsh environment that may cause different faulty events. Examples of fault causes are temperature changes, vibrations, aging and Electromagnetic Interferences (EMI). Such faults could generate potential changes producing transient or permanent component failures. The occurrence of such events in fieldbus networks may produce a subtle form of inaccessibility (virtual rather than physical partition). A network is said to be inaccessible when it temporarily ceases providing services and subset (or all) of nodes are unable to communicate with each others (Verissimo *et al*., 1997). Standard fieldbusses have means of recovering from such situations in time-consuming manner. Most of non-critical applications can live with such temporary glitches in network operation, provided these temporary periods of inaccessibility are bounded in time (Verissimo *et al*., 1991).

Generally, faulty events may be in hardware, software and communication subsystems. Faults in first two types are mainly based on design weakness. This study deals with the faults affecting the communication process in a harsh environment. The consequent failures for such faults may be performance or omission failures. During performance failures (or timing failures), the action may be too late or too early. However, the data will be correct. While omission failures are the no response case (Hansson *et al*., 2002). PROFIBUS protocol (Profibus, 2002) is the selected fieldbus for extending many research works to prove its timing behaviour. In addition, PROFIBUS is intended to extend its functionality to cover industrial wireless communication besides supporting industrial multimedia traffic (Ferreira *et al*., 2002).

The issue of network inaccessibility is investigated in many studies through which main LAN protocols are covered like: Token bus LAN (Rufino and Varissimo, 1992) token ring LAN (Rufino and Varissimo, 1992a) and FDDI LAN (Rufino and Varissimo, 1992b). In this context, Fieldbus protocols are also investigated such as CAN (Pinho *et al*., 2000) and PROFIBUS (Verissimo *et al*., 1997; Verissimo *et al*., 1991; Li, 1996). Concerning the PROFIBUS, these studies have investigated inaccessibility results from ring management actions (like the insertion of new stations, station leaving) and erroneous token passing. None of these studies have introduced an analytical model for the inaccessibility of PROFIBUS message/token transmission in the presence of transient faults. Such a trend is to the authors' best knowledge not covered in the published literature. This paper handles the evaluation of worst-case and best-case inaccessibility overhead resulting from transient faults hitting data transmission among PROFIBUS stations. Inaccessibility and error overhead are interchangeably used in this study with the same meaning. Different fault models were introduced in various studies. Bounded fault models are widely used with Worst-Case Response Time (WCRT) calculations since they suggest a bounded separation between faults (Hansson *et al*., 2002; Broster,

Corresponding Author: Saad Hazim Al-Tak, College of Engineering, University of Baghdad, Iraq

2003; Punnekkat *et al*., 2000). Other models are based on a probabilistic framework (Broster, 2003; Burns *et al*., 2003). In this research, a bounded fault model is adapted with customised specifications to accomplish analysis in which single bit errors and burst errors are introduced. Both of them are assumed to be bounded in their arrival time. Different scenarios are proposed to evaluate the worst-case and best-case error overhead. Both cases can help in system design by outlining overhead margins introduced to the message response time.

PROFIBUS PROTOCOL

The PROFIBUS is a well-known standard and widely used fieldbus. It is goaled to be simple, rugged and reliable, can be expanded online and can be used in both standard environments and hazardous areas. This study will give main concepts of this protocol and brief description of the relevant parameters (Profibus, 2002; Tovar and Vasques, 1999; Tovar and Vasques, 1999a; Nieuwenhuyse and Behaeghel, 2003).

Message cycle: PROFIBUS has a multi-master architecture in which a message cycle (or a transaction) is initiated by master stations, while slave stations only transmit upon master request. In this vision, the station that sends the first frame (action frame) is said to be the initiator of that transaction, while the destined station is the responder. The transaction is said to be complete whenever the related acknowledge or response frame is received correctly by the initiator. The responder must reply before the expiration of the slot time timer (T_{SL}) at the initiator. Otherwise, the initiator repeats its frame a number of times depending on the protocol parameter; max_retry_limit. At the network setup phase, the maximum number of retries (max_retry_limit) must be defined uniquely in all master stations.

The PROFIBUS uses a broadcast medium in which all stations watch all transmitted frames to distinguish the addressed ones and to maintain their knowledge lists (as will be shown later). Generally, a master station inserts an idle time (T_{ID}) which is a period of physical medium inactivity, after an acknowledgement, response, unacknowledged request, or token frame. The (T_{ID}) parameter can be set individually for each master station.

Token passing mechanism: Bus mastership is served through token passing (or rotation) mechanism. A logical ring is formed by arranging master stations based on ascending addresses. A station address can be ranged from 0-126 per segment. Address 127 is reserved for broadcast and multicast messages. HSA is the Highest Station Address installed and not allowed to be a master station. If a master station receives a valid token frame in which the destination address is equivalent to This Station (TS) address, it checks whether the token is sent by its Previous Station (PS) to accept the token and own the mastership. Otherwise, the frame is discarded. If the same token frame is received again, it is accepted and the related source address is considered the new (PS).

In this manner, when a token holder decides to leave the mastership, it will send a token frame destined to its Next Station (NS). If the (NS) does not exhibit any action within (T_{SL}), the token transmission is repeated. If there is no activity in the 2nd trial and then in the 3rd, (NS) is assumed to be quiet and (TS) starts token passing to the successor of (NS) in the logical ring and so on until a master station accepts being the token holder.

Bandwidth allocation: Measurement of the token rotation time is started after receiving the token and ended after the next token arrival, resulting the real token rotation time (T_{RR}). Another parameter; target rotation time (T_{TR}) is assigned equally to all masters in the network. After receiving the token, the token holding time (T_{TH}) timer counts down with a starting value of the difference between (T_{TR}) and (T_{RR}). PROFIBUS uses two types of messages: high priority and low priority. (T_{TH}) timer is always checked before any message execution as briefed:

- After token arrival: $T_{TH} = T_{TR}-T_{RR}$
- Regardless (T_{TH}), one high priority message is performed.
- While (T_{TH}) timer does not expire, subsequent high priority message cycles are executed.
- After completing all pended high priority messages and if (T_{TH}) does not have been expired yet, the execution of low priority message cycles may be started.
- A message cycle execution includes any necessary retransmissions.

After all high priority messages have been executed; poll list message cycles are started. When the poll cycle is completed within (T_{TH}), the requested low priority non-cyclical messages are then carried out. If a poll cycle takes several token visits, the poll list is handled in segments.

Ring maintenance: First maintenance rule is defined for a station which is newly switched on. It needs to listen passively on the medium during 2 successive token cycles. Meanwhile, a valid view on the entire logical ring is established by the new station which is not allowed to send or receive any data or token frame. Every station

address found in a token frame during this interval is included into the List of Active Stations (LAS) table. After building (LAS) during these two cycles, the station can enter the ring if it is invited by another station token. Second maintenance rule is to update the (LAS) by inspecting address fields of the transmitted token frames after joining the ring.

A special rule is used for the first ring initialization or after a token loose. Each master station has a time-out timer which is used to monitor bus activity. The timer value is related to the station address by ($T_{time\text{-}out} = 6T_{SL} + 2(TS)\times T_{SL}$).The second term ensures that each station timer expired in unique time. In this way, no 2 master stations claim the token simultaneously at initialization or after a token loose.

In order to track changes in the logical ring, every master station included in the ring maintains a Gap List (GAPL) table which contains all address ranged between (TS) and (NS). Every time the Gap Update Timer (T_{GUD}) for a master station expires, it must check all addresses in its (GAPL) by sending a Request-FDL-Status frame to a single address and waiting for a response not more than (T_{SL}). (T_{GUD}) is calculated by ($T_{GUD} = G\times T_{TR}$) where G is the Gap Update Factor (between 1 and 100) which is specified by the Data Link Layer (DLL).

Another rule may optionally be maintained is the Live List (LL). Performing this rule requires an explicit demand by a Request-FDL-Status frame which is sent cyclically for each destination address (ranged from 0-126) except to the master stations because they are already included in the (LAS). By including (LAS) and positively responded slave stations in (LL), a list of all active (master and slave) stations is obtained.

In order to detect a defective transceiver and resolve any possible collisions, a special rule enables the token sender to read back (hearback) from the medium every transmitted bit. If the token holder (TS) detects a difference for the first time, it completes the transmission and waits for a bus activity within (T_{SL}). If no activity is encountered, (TS) starts sending the token for the second time with the same rule for hearback. However, any other mismatch is detected, results in discarding the token transmission immediately and (TS) removes itself from the ring, behaving as a newly switched on with an empty (LAS) and (LL).

Frame format: Each Protocol Data Unit (PDU) is coded in UART character, in which 11 bits are used to encode 8 data bit. The remaining three bits are the start, stop and parity bits. Each Action PDU, the first PDU transmitted in all transactions, must be preceded by a synchronization period of at least 33 idle bit periods (T_{SYN}). Every PDU starts with a Start Delimiter (SD) that characterizes its type. Token PDU consists of three UART characters, in addition to (SD), the Source Address (SA) and the destination address (DA). Other PDUs are those with variable data field or with fixed data field, besides fixed length PDU without data field. All message PDUs other than the token PDU, have Frame Check Sequence (FCS) of 8 bits. It is a simple checksum for all PDUs except token and short acknowledgement frames.

Message worst-case response time: As a master station is able to transmit, at least, one high priority message per received token (no matter if there is enough token holding time left), a maximum queuing delay can be guaranteed for PROFIBUS messages. Defining T_{cycle} as the upper bound between two consecutive token arrivals to a particular master, the maximum queuing delay of a single message request (Q) is equal to T_{cycle}. Note that this only guarantees a maximum transmission delay for the first high priority message in the outgoing queue. If there are m pending messages in the outgoing queue it will take, in the worst-case, m token visits to execute all those high priority messages.

PROFIBUS implements First-Come-First-Served (FCFS) outgoing queues. Consequently, if nh_i^k represents the number of high priority message streams in a master k waiting transmission before message cycle i, then the maximum number of pending messages will be nh_i^k. Thus, an upper bound for the message queuing delay in a master k is (Tovar and Vasques, 1999a):

$$Q = nh_i^k \times T_{cycle}^k \quad (1)$$

And:

$$T_{cycle}^k = T_{TR} + n \times C_m \quad (2)$$

Where n is the number of masters and C_m is the longest message cycle in the network. Worst-case response time for a message cycle is given by:

$$R_i^k = nh_i^k \times T_{cycle}^k + Ch_i^k \quad (3)$$

Where Ch_i^k is the worst-case duration of a message cycle i in master k. In this way, when C_{req} and C_{resp} are, respectively the duration of the request and response frame, Ch_i^k can be calculated by (Ferreira *et al*, 2002):

$$Ch_i^k = T_{ID} + C_{req} + T_{SL} + C_{resp} \quad (4)$$

Note that C_m can be calculated in the same way of Eq. 4 with different C_{req} and C_{resp} values. Usually, worst-case

message cycle (calculated by Eq. 3) is defined by considering message duration with its maximum number of retries is exhausted. The above argument is very pessimistic in that it supposes the use of all possible retries. In addition, it does not consider errors that may hit token frames. Our proposed analysis relies on how errors occur without the necessity of exploiting all the allowed retries for message transmissions. On the other hand, the proposed analysis suggests failure semantics concerning token transmissions.

FAULT MODEL

The proposed fault model assumes that fault arrival rate is bounded (i.e., there is a minimum interval between two consecutive faults). As mentioned previously, such a model is an appropriate choice to be used in conjunction with the WCRT formulation because of its bounded nature. On the contrary to the bounded model, probability based models which interpret the stochastic nature of fault arrival, have a complex nature and they can not set extremes for the fault behaviour.

In the literature, fault influence can be a combination of single bit error and multiple bit error (or burst error) like (Tindell and Burns, 1994; Navet *et al*., 2000). Others just rely on burst errors with special assumptions (Punnekkat *et al*., 2000; Hansson *et al*., 2000). These fault consequences reflect the real behaviour of frame transmission between fieldbus nodes suffering from transient interference. While all the mentioned studies postulate the number of error overheads resulting from burst error effects, this analysis deduces the number of error overheads according the burst and protocol characteristics. In the proposed model, the expected fault may cause a single bit error or/and burst error. The error rate, which is the minimum time between single bit errors, is bounded by T_e(milliseconds). In the same manner, burst error rate is bounded by T_{be}(milliseconds), while the burst length is bounded by N_{be} (measured in bits). Faults either hit the transmission of the queued messages during waiting the turn of the related message, or hit the transmission of the current message. In other words, fault propagation is a key feature of the proposed model. Since errors may occur during a token transmission, erroneous tokens participate in the overhead added to a message response time.

After discussing each scenario separately, they are integrated in the general WCRT. Including single bit error with burst errors in the general WCRT allows maneuvering in case of different error sources with different behaviors. Also, it enables analyzing the effect of these sources on the WCRT simultaneously (for example, to find the threshold value beyond which the message deadline is violated). One can simply neglect any of them by equaling its component to zero.

This analysis contains some explicit assumptions that need clarification. For example, it assumes each source of arrivals is independent of each other, as a consequent, errors are uncorrelated but their effects may interfere. The existence of correlations would complicate the analysis-but pessimistic assumptions may be relatively straightforward to incorporate. The life time of the fault is assumed to be either one bit duration in case of single bit error or multiple bit duration in case of burst errors. Also, the occurrence of faults is assumed to be synchronised with the transmitted bits. Such assumption is practically accepted since the fault effect on transmitted bits is one of two states; corruption or not.

The proposed analysis deals with maximum (worst-case) and minimum (best-case) overhead resulting from transient errors. In this approach, best-case overhead does not mean extinction of errors but the lowest effect of them. The knowledge of both cases illuminates the inaccessibility boundaries under each of the following scenarios and yields more understanding for the system behaviour. The following section introduces various error scenarios with the analysis of maximum and minimum inaccessibility overheads.

INACCESSIBILITY SCENARIOS

Individual analysis for the case of non-token frame and token frame errors are discussed in the following sub-sections, regarding errors to be either single bit errors or burst errors as defined in the fault model. Table1 summarizes various symbols used later on.

Recall Eq. 3:

$$\begin{aligned} R_i^k &= nh_i^k \times T_{cycle}^k + Ch_i^k \\ &= nh_i^k \times (T_{TR} + n \times C_m) + Ch_i^k \\ &= nh_i^k \times T_{TR} + nh_i^k \times n \times C_m + Ch_i^k \end{aligned}$$

The first item in this equation ($nh_i^k \times T_{TR}$) is error independent. Contrarily, both other items ($nh_i^k \times n \times C_m$ and Ch_i^k) are error dependent since errors can cause retries for them. In such a case, the maximum faulty message overhead that may result is:

$$\begin{aligned} O_i^k &= max\_retry\_limit \\ &\times (nh_i^k \times n \times C_m + Ch_i^k) \end{aligned} \quad (5)$$

Single bit errors in the message frame: The response time of a message frame is delayed by the consequence of retransmitting the corrupted frames. The corruption may be in the considered message frame or/and the queued message frames that are transmitted before it. By using the same concept in both situations and the concept producing Eq. 5, a specific condition is checked out; if

Table 1: Summary of the applied symbols

Symbol	Description	Symbol	Description
Ch_i^k	Worst-case duration of a message cycle *i* in master *k* (in msec)	T_{tkn}	Duration of a token frame (in msec)
nh_i^k	Number of high priority message streams waiting transmission before message cycle i in a master *k*	T_{cycle}	Maximum queuing delay of a single message request (in msec)
T_{SL}	Slot time (in msec)	$T_{time\text{-}out}$	Time-out timer
T_{ID}	Idle time (in msec)	HSA	Highest Station Address
N	No. of masters	N_{st}	No of active master stations
C_m	Longest message cycle in the network (in msec)	N_{lowest_add}	Lowest address in master stations
C_s	Shortest message cycle in the network (in msec)	T_e	Minimum time between single bit errors
max_retry_limit	Maximum No. of retries	T_{be}	Minimum time between burst error arrivals
T_{bit}	duration of one bit (in msec)	N_{be}	Maximum length of an burst error (in bits)
E_i^k	Overhead due to single bit errors i(in msec)	$\lfloor x \rfloor$	Floor function
BE_i^k	Overhead due to burst errors (in msec)	$\lceil x \rceil$	Ceiling function

there is any possibility that 2 (or more) successive errors may occur during the transmission period of a message frame and its retry, the maximum inaccessibility overhead will contain all the allowed retry attempts. Otherwise, the overhead is restricted to a single retry. This is can be generalised as follows:

$$E_i^k = A_e \times nh_i^k \times n \times C_m + B_e \times Ch_i^k \quad (6)$$

A_e represents the number of retries that may occur during the transmission of queued messages due the single bit error pattern, while B_e is the number of retries results during the specified message transmission. Each of A_e and B_e can not be more than the max_retry _limit value. The following conditions govern the overhead amount (The following functions are used within the paper context. The floor function ($\lfloor x \rfloor$) is the greatest integer not greater than *x*. The ceiling function ($\lceil x \rceil$) is the smallest integer not smaller than x):

$$A_e = \begin{cases} \text{max\_retry\_limit if } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_e} \right\rfloor \geq 1, \text{and} \\ 1 \qquad \text{elsewhere} \end{cases}$$

$$B_e = \begin{cases} \text{max\_retry\_limit if } \left\lfloor \dfrac{2(Ch^k{}_i - T_{bit}) - T_{ID}}{T_e} \right\rfloor \geq 1, \text{and} \\ 1 \qquad \text{elsewhere} \end{cases}$$

In terms of worst and best-cases, Eq. 6 is rewritten as:

$$E_i^{(wc)k} = \text{max\_retry\_limit} \times (nh_i^k \times n \times C_m + Ch_i^k) \quad (7)$$

$$E_i^{(bc)k} = nh_i^k \times n \times C_m + Ch_i^k \quad (8)$$

Burst errors in the message frame: As burst errors are bounded by a minimum arrival time (T_{be}), their effect on message frames can be introduced in the same manner of the single bit error. Moreover, if the burst error has enough extension to hit a message frame and its retry(s), this will be considered in the error overhead. If so, then the burst length-at least-must be equal to the sum of minimum bits to sense the error in both successive frames and the minimum value of (T_{SL} and T_{ID}). Either (T_{ID}) may be intermediate between two message frames or (T_{SL}) between a message frame and it's retry. Assuming max_retry_limit is greater than one, the error overhead can be expressed by the same way of Eq. 6:

$$BE_i^k = A_{be} \times nh_i^k \times n \times C_m + B_{be} \times Ch_i^k \quad (9)$$

In the same manner of the study, A_{be} and B_{be} represent the number of retries that may occur during the transmission of the queued messages and the specified message respectively under a defined fault model. A_{be} and B_{be} are calculated according to the following formulas:

$$A_{be} \begin{cases} 1 \quad \text{if } \max(x_1, x_2, \ldots x_m) = 1, \\ \quad \text{and } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 0 \\ 2 \quad \text{if } \max(x_1, x_2, \ldots x_m) = x_2, \\ \text{and } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 0 \\ \ldots\ldots \\ \text{max\_retry\_limit if } \max(x_1, x_2, \ldots x_m) \\ = x_m, \text{ or } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor \geq 1 \end{cases}$$

$$B_{be} \begin{cases} 1 \quad \text{if } \max(x_1, x_2, \ldots x_m) = 1, \\ \quad \text{and } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 0 \\ 2 \quad \text{if } \max(x_1, x_2, \ldots x_m) = x_2, \\ \text{and } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 0 \\ \ldots\ldots \\ \text{max\_retry\_limit if } \max(x_1, x_2, \ldots x_m) \\ = x_m, \text{ or } \left\lfloor \dfrac{2(C_m - T_{bit}) - T_{ID}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor \geq 1 \end{cases} \quad (10)$$

Where

$m = 1, 2, \ldots,$ . . . . . .

$$x_m = \begin{cases} 1 & \text{if } \ldots \\ m \times \overbrace{\left\lfloor \dfrac{N_{be} \times T_{bit}}{2T_{bit} + (m-2) \times C_s + (m-1) \times \min(T_{SL}, T_{ID})} \right\rfloor}^{*} & \text{if } \ldots \end{cases}$$

* If it is greater than 1 , considered to be equal to 1.

As can be noticed, values of A_{be} and B_{be} are always bounded by the max_retry_limit value. The value of A_{be} or B_{be} depends on checking two conditions. The first one; $x(x_1, x_2, \ldots, x_m)$, is reflecting the effect of the burst length in the network overhead. The C_s (shortest message cycle) is used in the calculation of x_m to adapt the worst-case situation where C_s maximize the possibility of hitting successive message frames. The burst may hit a single message or a message with its retry(s) depending on the burst length in addition to the message length. While the second condition (the ceiling function in A_{be} or B_{be} formulas), stands for the effect of separation of successive bursts. This latter case accounts for the situation where the span between a burst end and its successive burst start hits the transmission of a message and its retry. If such a case occurs, then all the retries are exhausted.

The worst-case and best-case overheads are calculated by taking the highest extremes and the lowest extremes, respectively. These can be expressed as:

$$\overset{(wc)}{BE}_i^k = \text{max\_retry\_limit} \times (nh_i^k \times n \times C_m + Ch_i^k) \quad (11)$$

$$\overset{(bc)}{BE}_i^k = nh_i^k \times n \times C_m + Ch_i^k \quad (12)$$

Single bit error in the token frame: Two different error consequences are possible to occur in the case of faults hitting the token frame; omission failure or a hear-back removal. Omission failures occur if the first token frame is corrupted or failed to be received correctly and no bit confliction is recognised by the source node, but the next trial(s) of token transmission succeeds and a bus activity from the next station, is recognised. The worst-case error overhead due to such failures exhausts both retries and is given by:

$$\overset{(wc)}{E}_{i\ tkn \to omission}^k = (nh_i^k \times n + 1) \times 2(T_{SL} + T_{tkn}) \quad if \left\lfloor \frac{2T_{tkn} - T_{bit} + T_{SL}}{T_e} \right\rfloor = 1 \quad (13)$$

Note that upper condition does not include 'greater' sign because it will lead to the corruption possibility of each transmitted token, which is practically not expectable. Also note that the term ($nh^k i \times n+1$) represents pessimistically the number of token rotations until finally reaching master 'k'.

In case that one retry is sufficient (i.e., only single error hits the original token frame) to stimulate the next station reaction, the best-case inaccessibility can be expressed by:

$$\overset{(bc)}{E}_{i\ tkn \to omission}^k = (nh_i^k \times n + 1) \times (2T_{SL} + T_{tkn}) \quad if \left\lfloor \frac{2T_{tkn} - T_{bit} + T_{SL}}{T_e} \right\rfloor = 0 \quad (14)$$

While in the hearback removal, a more pessimistic situation can take place. If the source node (or This Station (TS)) senses error(s) in the transmitted token frame, it completes transmission and waits for the response. In case of no response is received, it will retransmit the token frame and if any error is sensed, the node stops transmission and remove itself from the ring. In such a case, the worst-case error overhead occurs when the node senses a confliction in the last bit of the retransmitted token frame. In addition, $T_{time\text{-}out}$ maximum (worst-case) expiration time occurs whenever N_{lowest_add} =(HSA - N_{st}), where N_{st} is the number of active master stations. The worst-case error overhead can be written as:

$$\overset{(wc)}{E}_{i\ tkn \to hearback}^k = (nh_i^k \times n + 1) \times (T_{SL} + T_{tkn} + \overset{(wc)}{T}_{time-out}) \quad if \left\lfloor \frac{2T_{tkn} - T_{bit} + T_{SL}}{T_e} \right\rfloor \geq 1 \quad (15)$$

where :

$T_{time-out} = 6T_{SL} + 2N_{lowest\_add} \times T_{SL}$,

$\overset{(wc)}{T}_{time-out} = 6T_{SL} + 2(HSA - N_{st}) \times T_{SL}$

To calculate the best-case error overhead, the error and the sense of bit confliction is expected to take place as soon as possible, i.e. in the first bit from the retransmitted token frame. The $T_{time\text{-}out}$ timer expires with minimum (best-case) time if N_{lowest_add} = 1 (as the minimum allowed address for a master station is 'one'), as shown in the following:

$$E^{(bc)k}_{i\ tkn\rightarrow hearback} = (nh_i^k \times n+1)\times (T_{SL} + T_{bit} + T^{(bc)}_{time-out})$$
$$if \left\lfloor \frac{T_{tkn} - T_{bit} + T_{SL}}{T_e} \right\rfloor \geq 1$$
$$= (nh_i^k \times n+1)\times (9T_{SL} + T_{bit}) \quad (16)$$

since:

$T_{time-out} = 6T_{SL} + 2N_{lowest\_add} \times T_{SL}$,

$T^{(bc)}_{time-out} = 8T_{SL}$

Burst errors in the token frame: Burst errors with token transmission have similar conditions to that relating the burst errors in message frames by taking into account the effect of the burst length mutually with the effect of adjacent burst errors. The first condition used in formulating the following equations deals with the burst length effect, while the second condition deals with error-free separation between successive burst errors. To calculate the omission failure overhead results from burst errors, the same formula introduced in study is used but with different conditions:

$$BE^{(wc)k}_{i\ tkn\rightarrow omission} = (nh_i^k \times n+1)\times 2(T_{SL} + T_{tkn})$$
$$if \left\lfloor \frac{2(T_{tkn} - T_{bit}) + T_{SL}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 1 \quad or \left\lfloor \frac{N_{be} \times T_{bit}}{2T_{bit} + T_{SL}} \right\rfloor \geq 1 \quad (17)$$

If neither the arrival time of a burst error nor its length may hit more than a single token frame, a best-case error overhead is written as:

$$BE^{(bc)k}_{i\ tkn\rightarrow omission} = (nh_i^k \times n+1)\times (2T_{SL} + T_{tkn})$$
$$if \left\lfloor \frac{2(T_{tkn} - T_{bit}) + T_{SL}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor = 0 \quad and \left\lfloor \frac{N_{be} \times T_{bit}}{2T_{bit} + T_{SL}} \right\rfloor = 0 \quad (18)$$

While concerning hearback removal, the worst-case scenario assumes the sense of bits contradiction occurs with the last bit in the retransmitted frame where (TS) decides to remove itself from the ring. This assumption is satisfied whenever the error-free separation between two successive bursts is not larger than the duration of the token and its retransmission besides T_{SL} that splits them. The resultant WC overhead is given by:

$$BE^{(wc)k}_{i\ tkn\rightarrow hearback} = (nh_i^k \times n+1)\times (T_{SL} + T_{tkn} + T^{(wc)}_{time-out})$$
$$if \left\lfloor \frac{2(T_{tkn} - T_{bit}) + T_{SL}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor \geq 1 \quad (19)$$

Where

$T^{(wc)}_{time-out}$

is calculated as in (15). The burst length component must be included in the best-case scenario. The burst length together with the minimum separation between consecutive bursts is considered. Any of them fulfils hitting the first bit in the retransmitted token frame with - at least - the last bit from the first frame will lead to the fastest hear-back removal:

$$BE^{(bc)k}_{i\ tkn\rightarrow hearback} = (nh_i^k \times n+1)\times (T_{SL} + T_{bit} + T^{(bc)}_{time-out})$$
$$if\ any\ of \left\{ \left\lfloor \frac{T_{tkn} + T_{SL} - T_{bit}}{T_{be} - (N_{be} \times T_{bit})} \right\rfloor, \left\lfloor \frac{N_{be} \times T_{bit}}{2T_{bit} + T_{SL}} \right\rfloor \right\} \geq 1 \quad (20)$$

Where

$T^{(bc)}_{time-out}$

is calculated as in (16). As can be noticed that for the worst case scenarios, Eq. 7, 13 and 15 outline the inaccessibility overhead as well as Eq. 8, 14 and 16 do for the best case scenarios.

THE GENERAL WCRT INCLUDING ERROR OVERHEADS

The analysis introduced above, details the protocol behaviour when different error scenarios can take place. The worst-case response time (defined by Eq. 3) can be re-written with the addition of worst-case error components in abstract manner as shown in Eq. 21. As the erroneous token and message frames may participate in the error overhead, each of them has its own component in the general WCRT. Each component includes both the single bit error and burst error effects. This will help in precise mapping for any fault behaviour concerning the investigated network. In this formalisation, token frames are assumed to be subjected to omission faults only, since they occur more frequently than the hearback faults. Such assumption results in a simplified analysis and overcome the situation where an extraordinary overhead results in insensible values of WCRT that breakdown the timing analysis and system design.

The single bit error effect is included by regarding the maximum number of error events (the first ceiling function of both faulty components in Eq. 21) hitting the overall interval $(0, R_i^k]$. By considering burst error component, the ceiling function of both faulty (message and token) components represents the maximum number of burst

errors that may occur during the same interval. O_{msg} and O_{tkn} include the burst error effects on message and token frame, respectively where the burst length or error-free separation between bursts can contribute in these effects. By observing Eq. 22 and 23, first component, max (x_1,x_2, x_m), gives the possibility of the same burst to hit two or more successive frames (a message and its retries), assuming pessimistic situation that the burst phasing results in such effect. While the floor function (the second component) introduces the possibility of error-free separation between burst errors (separation between a last burst bit and the first bit in the next burst) not to hit a specified frame only but to outreach its effect to exhaust all the retries. As in Eq. 22, the first component in Eq. 23, Y_{tkn}, covers the possibility of the burst length to hit more than the token and its retries (2 retries). While the second component has the same rule of the second component in Eq. 22.

C_m represents single error overhead on a message frame, while the item: (T_{SL}+T_{tkn}) represents the token error overhead. The net value of the items included in the faulty message component (multiplicand of C_m) must not be greater than: max_retrey_limit×(nh_i^k×n+1). Otherwise, it exceeds the maximum worst-case value that represents exploiting all the retries in all transmitted message frames. Exceeding this limit may be explained as an existence of extreme noisy environment that prevents a guaranteed real-time performance. While the net value of the (T_{SL}+T_{tkn})'s multiplicand must not exceed, 2×(nh^k_i×n+1), the allowed token retries during R_i^k. In the same way, exceeding (or even reaching) such limit is due to severe disturbance in the environment. Remember that Eq. 21 is not exact but sufficient, since faults not always induce a maximum overhead load.

$$R_i^k = nh_i^k \times T_{cycle}^k + Ch_i^k \quad \leftarrow \text{fault free component}$$

$$\leq \text{max\_retery\_limited} \times (nh_i^k \times n+1)$$

$$+\left\{\left\lceil \frac{R_i^k}{T_e} \right\rceil \times O_{msg}\right\} \times C_m \quad \leftarrow \text{faulty massege component}$$

Single bit error component; Error brust component

$$\leq 2\times(nh_i^k \times n+1)$$

$$\left\lceil \frac{R_i^k}{T_e} \right\rceil \times O_{tkn}\} \times (T_{SL} + T_{tkn}) \quad \leftarrow \text{faulty token component}$$

Single bit error component; Error brust component

(21)

Where,

$$O_{msg} = \max(x_1, x_2, \ldots x_m) + \left\lceil \frac{2(C_m - T_{bit}) - T_{ID}}{T_{eb} - N_{eb} \times T_{eb}} \right\rceil^{**} \quad (22)$$

$$O_{tkn} = Y_{tkn} + \left\lceil \frac{2(T_{tkn} - T_{bit}) - T_{SL}}{Teb - N_{eb} \times T_{eb}} \right\rceil^{***} \quad (23)$$

$$m = 1, 2 \ldots, max\_retery\_limit$$

To solve an equation like (21), a recurrent relation (Burns *et al*., 2003) is produced:

$$x_m = \begin{cases} 1 & \text{if } m.\ 1 \\ m \times \left\lceil \frac{N_{be} \times T_{bit}}{2T_{bit} + (m-2) \times C_s + (m-1) \times \min(T_{SL}, T_{ID})} \right\rceil^{*} & \text{if } m\ 1 \end{cases}$$

$$y_{tkn} = \begin{cases} 1 & \text{if } \left\lfloor \frac{N_{be} \times T_{bit}}{2T_{bit} + T_{SL}} \right\rfloor = 0 \\ 2 & \text{if } \left\lfloor \frac{N_{be} \times T_{bit}}{2T_{bit} + T_{SL}} \right\rfloor \geq 1 \end{cases}$$

* If it is greater than 1 , considered to be equal to 1.
** If not equal to 0, it must force the 'faulty message component' to be equal to its upper bound.
*** If not equal to 0, it must force the 'faulty token component' to be equal to its upper bound.

$$r_i^{(n+1)k} = nh_i^k \times T_{cycle}^k + Ch_i^k$$

$$\leq \ldots\ldots\ldots\ldots \times (nh_i^k \times n+1)$$

$$+ \left\{ \left\lceil \frac{r_i^{(n)k}}{T_e} \right\rceil + \left\lceil \frac{r_i^{(n)k}}{T_{eb}} \right\rceil \times O_{msg} \right\} \times \max(Ch_i^k, C_m) \quad (24)$$

$$\leq 2\times(nh_i^k \times n+1)$$

$$+ \left\{ \left\lceil \frac{r_i^{(n)k}}{T_e} \right\rceil + \left\lceil \frac{r_i^{(n)k}}{T_{eb}} \right\rceil \times O_{tkn} \right\} \times (T_{SL} + T_{tkn})$$

Where

$r_i^{(0)k}$

is considered an initial value (usually the value of C_m). The recurrence procedure will be stopped whenever

$$r_i^{(n+1)}$$

is equal

$$r_i^{(n)}$$

to and this will be the value of the worst-case response time R_i^k.

CASE STUDY

A numerical example is presented in order to analyze the proposed model and to elaborate some conclusions on them. An assumption is made of a PROFIBUS network composed by four masters and four slaves. Table 2 presents the network parameters, where T_{TR} is calculated using unconstrained low priority traffic profile introduced in Tovar *et al*. (1999a) while other parameters are suggested on the base of applicable ranges.

Masters (M1 to M4) are assumed to exchange messages with slaves (S1 to S4). For each master (k), a message cycle (i) is assumed to have a deadline (Dh_i^k) which is equal to th e period of message. The worst case duration(Ch_i^k) is calculated according to Eq. (4) and the given network parameters.

Table 2: Network parameters

Parameter	Value
Bit rate	1.5Mbps
T_{SL}	225isec
T_{ID}	100isec
T_{TR}	600isec
Max_retry_limit	1

Table 3 presents the WCRT for the message streams under study using different error parameters for each message cycle. The WCRT under error free condition is calculated on the base of Eq. (3), while WCRT under different error scenarios is calculated using recurrence relation of Eq. (21). The highlighted cells represent the failed WCRT to fulfil deadline constraints. The more severe error scenario, the more failed message cycles to guarantee real-time performance. Even though, the system may be still considered as working within the designed margins. Such concept is based on the flexible design resulting in that violating the real-time constraints occurs only in case of crossing over the allowed limit of failed deadlines (Broster , 2003).

For more investigation of Eq. (21), each of the three variables T_e, T_{be} and N_{be} is varied while the others are fixed. The investigation is performed with the assumption that the variables are independent. Practically, this is an accepted assumption since faults are random and independent in nature. Although any message cycle can be chosen, but the last message cycle of M3 (with Stream No. =11) is selected which has a moderate deadline. From now on, this message cycle will be mentioned in short form as 'z-message'. The parameter 'max_retry_limit' is set to 2 in order to clarify the effect of N_{be}.

By varying burst length and using the same equation, Fig. 1(a) illustrates the response time of z-message with different T_{be}. The higher T_{be}, the lower variance in response time is observed. For low values of N_{be} and high T_{be}, response times are very close in case of T_{be} =6 and 10 msec. This behaviour can be understood since for high separation between burst arrival times and relatively small burst lengths, the consequent error overhead will be

Table 3: The WCRT under different error scenarios

				WCRT under the fault model		
From	Ch_i^k (msec)	Dh_i^k (msec)	error free WCRT	T_e=15 msec T_{be}=15 msec N_{be}=300 bit	T_{be}=10 msec T_e=10 msec N_{be}=600 bit	T_e=5 msec T_{be}=5 msec N_{be}=900 bit
M1	0.435	25	11.50	13.87	16.25	23.38
	0.530	50	11.59	13.97	16.34	23.47
	0.530	50	11.59	13.97	16.34	23.47
	0.405	12	11.47	13.84	16.22	23.35
M2	0.435	20	8.73	11.11	13.48	18.23
	0.405	25	8.70	11.08	13.45	18.20
	0.545	12	8.84	11.22	13.59	18.34
M3	0.530	25	11.59	13.97	16.34	23.47
	0.545	12	11.61	13.98	16.36	23.49
	0.435	50	11.50	13.87	16.25	23.38
	0.405	20	11.47	13.84	16.22	23.35
M4	0.545	20	6.08	8.45	8.45	13.20
	0.435	12	5.97	8.34	8.34	13.09

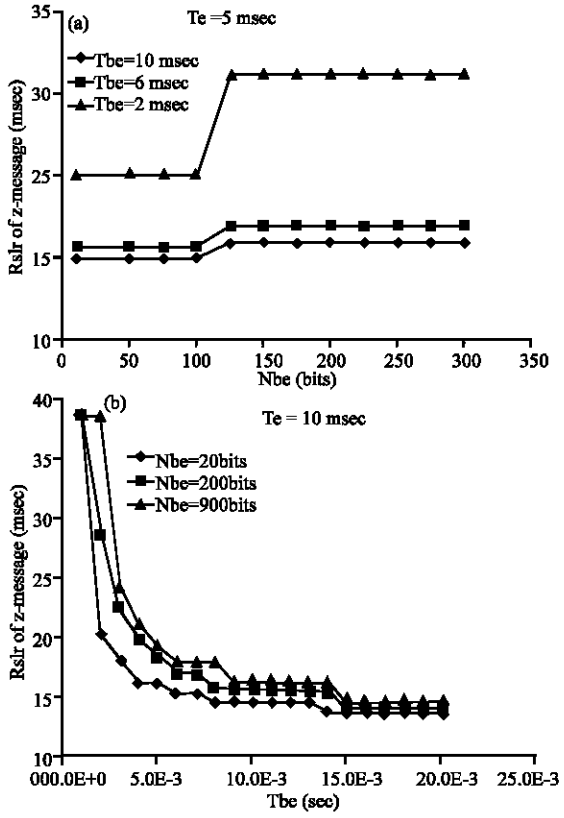

Fig. 1: Variation of z-message response time versus: (a) N_{be} and (b) T_{be}

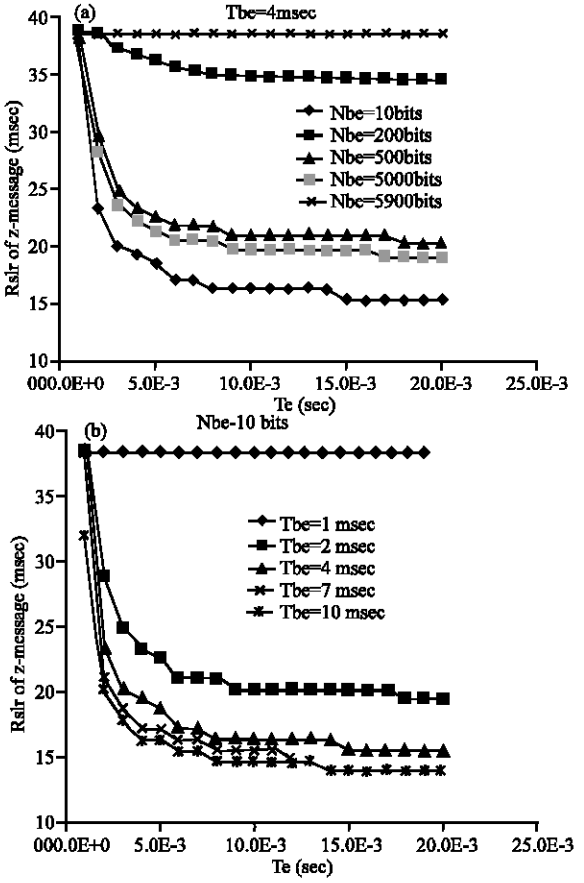

Fig. 2: Variation of z-message response time versus T_e with different

comparable. The independence behaviour for the response time (except one change) against N_{be} yields from the bounds of O_{msg} and O_{tkn}. The higher saturation level represents the WCRT where all more retries are exhausted.

Figure 1(b) illustrates the variation of response time against T_{be}. Three values for N_{be} are used in this investigation. With large values for T_{be}, response times for z-message become closer in spite of having various burst length. On the other hand, response time variations become more sensible as T_{be} is being smaller. This behaviour is expected since for higher T_{be}, the dominant effect is just for the burst length whether it hits message and its reply or not. The response time saturates to the highest level for high N_{be} and small T_{be}.

In the same manner, the effect of T_e variation on the response time can be observed in Fig. 2 a and b. The effect of N_{be} and T_{be} on such investigation is shown respectively. The same outline for response time can be observed with small N_{be} and large T_{be} values and vice versa. Generally, the same behaviour for the response time can be noticed in both figures since N_{be} and T_{be} variation are inversely related in their effect to the response time.

When the burst error is dominant, the response time saturates to the level where all retries are depleted.

CONCLUSION

The inaccessibility behaviour of PROFIBUS protocol is investigated against different error scenarios. In addition, each error scenario is related to a proposed formula that evaluates the inaccessibility overhead in worst and best cases conditions. The investigation focuses on the inaccessibility results from faulty transmission of message and token frames individually. The resultant overhead depends on the phasing and length of errors. The worst- and best-case scenarios give the extremes of such overhead. It can be seen that the same formulas representing overhead are introduced in either single bit errors or burst errors but with different error constraints. Finally, the WCRT equation is rehabilitated to include the formulated error loads by combining message and token overheads. Including single bit error with burst errors in the general WCRT allows maneuvering in case of different error sources with

different behaviors. Also, it enables analyzing the effect of these sources on the WCRT simultaneously.

Due to the unavailability of similar studies to compare with, the proposed formulas in this paper are tested using a suggested case study that clarifies the behaviour of PROFIBUS under various error scenarios. Although the introduced analysis assumes high pessimistic situations, it omits the cases of corrupted or lost token which lead to an extraordinary overhead. Such overhead results in insensible values of WCRT that breakdown the timing performance and needs more investigation in the future. Using the proposed formulas would enable the system designer, who has an outline of fault characteristics in the environment, to sketch out whether the network may achieve the targeted performance or not.

This study proposes a foundation for studying the protocol reliability and its ability to guarantee real-time requirements under faulty conditions. The proposed concepts and analyses are pioneer in the field of PROFIBUS timing analysis where the extreme WCRT analysis is usually adapted without taking into account the fault characteristics.

REFERENCES

Broster, I., 2003. Flexibility in Dependable Communication, PhD thesis, Department of Computer Science, University of York, York, YO10 5DD, UK.

Burns, G., G. Bernat and I. Broster, 2003. A Probabilistic Framework for Schedulability Analysis, Proceedings of the 3rd International Embedded Software Conference (EMSOFT)., pp: 1-15.

Ferreira, L., M. Alves and E. Tovar, 2002. Hybrid wired/wireless Profibus networks supported by bridges/routers, Proceedings of WFCS, Vasteras, Sweden.

Hansson, H., T. Nolte, C. Norström and S. Punnekkat, 2002. Integrating Reliability and Timing Analysis of CAN-based Systems, IEEE. Trans.Indus. Elec., 49: 6.

Hansson, H., C. Norström and S. Punnekkat, 2000. Integrating reliability and timing analysis of CAN-based systems, Proceedings of IEEE Workshop Factory Communication Systems (WFCS'2000), Porto, Portugal, pp: 165-172.

Li, M., 1996. Real-Time Communication in an Industrial Network-PROFIBUS", PhD Thesis No.1586. École Polytechnique Fédérale de Lausanne (EPFL).

Navet, N., Y.Q. Song and F. Simonot, 2000. Worst-case deadline failure probability in real-time applications distributed over controller area network, J. Sys. Architecture, 7: 607-617.

Nieuwenhuyse, K. and S. Behaeghel, 2003. Timing performance of a hybrid wired/wireless fieldbus, A Dissertation in Industrial Engineering, Polytechnic Institute of Porto (ISEP/IPP)-Portugal.

Pinho, L., F. Vasques and E. Tovar, 2000. Integrating inaccessibility in response time analysis of can networks, Proceedings of the IEEE Workshop Factory Communication Systems (WFCS'2000), Porto, Portugal, pp: 77-84.

Profibus-PROFIBUS Technology and Application-System Description, 2002. http://www.profibus.com.

Punnekkat, S., H. Hansson and C. Norström, 2000. Response time analysis under errors for CAN, Proceedings of IEEE Real-Time Technology and Applications Symposium (RTAS)., pp: 258-265.

Rufino, J. and P. Verissimo, 1992. A study on the inaccessibility characteristics of ISO 8802/4 Token-Bus LANs, Proceedings of the IEEE INFOCOM'92 Conference on Computer Communications, Florence, Italy (also INESC AR 16-92).

Rufino, J. and P. Verissimo, 1992a. A study on the inaccessibility characteristics of ISO 8802/5 Token-Ring LANs, Technical Report RT/24-92, INESC., Lisboa, Portugal, February.

Rufino, J. and P. Verissimo, 1992b. A study on the inaccessibility characteristics of the FDDI LANs, Technical Report RT/25-92, INESC, Lisboa, Portugal.

Tindell, K. and A. Burns, 1994. Guaranteed message latencies for distributed safety-critical hard real-time control networks, Department of Computer Science, University of York, York, U.K., Tech. Rep. YCS229.

Tovar, E. and F. Vasques, 1999. From Task Scheduling in Single Processor Environments to Message Scheduling in a Profibus Fieldbus Network, Lecture Notes in Computer Science, WPDRTS., 1586: 339-352.

Tovar, E. and F. Vasques, 1999a. Real-Time Fieldbus Communications Using PROFIBUS Networks, IEEE. Trans. Indus. Elec., 46: 1241-1251.

Verissimo, P., J. Rufino and L. Ming, 1997. How hard is hard real-time communication on field-buses?, IEEE 27th International Symposium on Fault-Tolerant Computing Systems, Seatle, Washington, USA., pp: 112-121.

Verissimo, P., J. Rufino and L. Rodrigues, 1991. Enforcing real-time behaviour of LAN-based protocols, Proceedings of the 10th IFAC Workshop on Distributed Computer Control Systems, Semmering, Austria.