INACCESSIBILITY IN PROFIBUS DUE TO TRANSIENT FAULTS

Saad H. Al-Tak*, Sufyan T. Faraj **,

*PhD researcher. Electronics & Comm. Dept., College of Eng., Univ. of Baghdad, Iraq
saadhazim2003@yahoo.com,

** Associate Prof., M IEEE, M ComSoc, Dean of College of Computers, Anbar University, Iraq
sufyantaih@yahoo.com

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 paper 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 suggested 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. This work is essential to assess the real-time behaviour of the protocol under the incidence of errors.

Keywords: *Fieldbus, Fault Model, Inaccessibility, PROFIBUS, Transient Faults, WCRT*

1. 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 [1]. 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 [21]. 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 [20].

Generally, faulty events may be in hardware, software, and communication subsystems [1]. Faults in first two types are mainly based on design weakness. This paper 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. PROFIBUS protocol [10] 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 [4].

The issue of network inaccessibility had been investigated in many studies through which main LAN protocols are covered like: token bus LAN [13], token ring LAN [14], and FDDI LAN [15]. In this context, Fieldbus protocols had been also investigated such as CAN [12], [9], and PROFIBUS [21], [20], [6]. Concerning the PROFIBUS, these studies had investigated inaccessibility results from ring management actions (like the insertion of new stations, station leaving), and erroneous token passing. None of these studies had introduced an analytical model for the inaccessibility of PROFIBUS message/token transmission in the presence of transient faults. Such a trend is to the author's 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 paper with the same meaning. Different fault models are introduced in various studies; Bounded fault models are widely used with worst-case response time calculations since they suggest a bounded separation between faults [2], [11]. Other models are based on a probabilistic framework [2], [3]. As shown later, a bounded fault model is adapted with customised specifications to accomplish this 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.

Fundamental aspects of PROFIBUS protocol are addressed in section 2. While section 3 introduces the fault model features. Different inaccessibility scenarios are analysed in section 4. In section 5, a novel *WCRT* formula is proposed including error effects. Finally, section 6 concludes the current work.

2. PROFIBUS PROTOCOL

The PROFIBUS is a well-known standard and widely used fieldbus [10],[17]. 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 section will give main concepts of this protocol [10], [19], [17], [8] and brief description of the relevant parameters within this paper.

2.1 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.

2.2 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 to 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 second trial and then in the third, (*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.

2.3 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 below:

- 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.

2.4 Ring Maintenance

First maintenance rule is defined for a station which is newly switched on. It needs to listen passively on the medium during two 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 ($T_{time-out} = 6T_{SL} + 2(TS) \times T_{SL}$). The second term ensures that each station timer expired in unique time. In this way, no two 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 to 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*).

2.5 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.

2.6 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} [18]. 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 [19] for the message queuing delay in a master k is:

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

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 [4] can be calculated by:

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

Note that C_m can be calculated in the same way of equation (4) with different c_{req} and c_{resp} values. Usually, worst-case message cycle (calculated by equation (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.

3. 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 literatures, fault influence can be a combination of single bit error and multiple bit error (burst error) like [16] and [7]. Others just rely on burst errors with special assumptions [11] and [5]. These fault consequences reflect the real behaviour of frame transmission between fieldbus nodes suffering from transient interference. While all the mentioned literatures 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 (section 4), they are integrated in the general WCRT (section 5). Including single bit error with burst errors in the general *WCRT* allows maneuvering in case of different error sources with different behaviours. 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 sections introduce various error scenarios with the analysis of maximum and minimum inaccessibility overheads.

4. 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.

Table 1: Summary of the applied symbols

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
Ch_i^k	worst-case duration of a message cycle to transfer k its bytes	T_{bit}	Duration of a token frame (in msec)
nh_i^k	number of high priority message instances starting transmission before message cycle i in a master k	T_{cycle}	maximum queuing delay of a single message request (in msec)
T_{TR}	Slot time (in msec)	$T_{token-out}$	Token-out time
T_{ID}	Idle time (in msec)	HSA	Highest Station Address
N	No. of masters	N_m	No of active master stations
C_m	longest message cycle in the network (in msec)	N_{lowest_add}	Lowest address in master stations
C_T	shortest message cycle in the network (in msec)	T_d	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 bit)
E_i^k	Overhead due to single bit errors (in msec)	$\lfloor x \rfloor$	Floor function
BE_i^k	Overhead due to burst errors (in msec)	$\lceil x \rceil$	Ceiling function

Recall equation (3):

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

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:

$$O_i^k = \max\_retry\_limit \times (nh_i^k \times n \times C_m + Ch_i^k) \qquad (5)$$

4.1 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 equation (5), a specific condition is checked out; if there is any possibility that two (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[1]:

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

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:

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

[1] The following functions are used within the paper context. The floor function ($\lfloor x \rfloor$) which is the greatest integer not greater than x. The ceiling function ($\lceil x \rceil$) which is the smallest integer not smaller than x.

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

In terms of worst and best-cases, equation (6) is rewritten as:

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

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

4.2 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 can be greater than one, the error overhead can be expressed by the same way of equation (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 previous section, 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:

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

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

Where

$m = 1, 2, ...,$ *max retry limit*

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

* 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, ...,x_m), is reflecting the effect of the burst length in the network overhead as shown in case (a) of Figure 1. 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 as shown in case (b) of the same figure. Case (b) 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. Cases (a) and (b) covers all possible effects of burst errors to hit the message transmission.

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

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

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

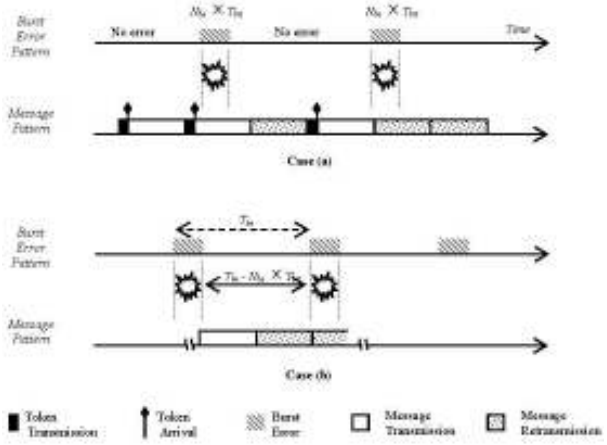

Figure 1: Illustration of possible burst effects on message transmission: Case (a) The effect of burst length, and Case (b) The effect of separation between successive bursts.

4.3 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. By observing Figure .2 (a), the worst-case error overhead due to such failures exhausts both retries and is given by:

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

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 $(nh_i^k \times n+1)$ 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:

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

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:

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

where:

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

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

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:

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

$$= (nh_i^k \times n+1)\times(9T_{SL}+T_{bit})$$

since:

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

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

4.4 Burst errors in the token frame

Illustration of typical burst errors hitting a token frame and its retransmissions is given in Figure 2(b). 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 as in section 4.2. The two conditions used in the following equations are illustrated in Figure 2, where case (a) deals with the burst length effect while case (b) deals with error-free separation between successive burst errors. To calculate the omission failure overhead results from burst errors, the same formula introduced in section 4.3 is used but with different conditions:

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

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:

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

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:

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

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:

$$BE^{(bc)k}_{i\ tkn\rightarrow hearback} = (nh_i^k \times n+1)\times(T_{SL}+T_{bit}+T^{(bc)}_{time-out})$$

$$if\ any\ of\ \{\left\lfloor \frac{T_{tkn}+T_{SL}-T_{bit}}{T_{be}-(N_{be}\times T_{bit})} \right\rfloor, \left\lfloor \frac{N_{be}\times T_{bit}}{2T_{bit}+T_{SL}} \right\rfloor\} \geq 1 \qquad (20)$$

Where $T^{(bc)}_{time-out}$ is calculated as in equation (16).

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

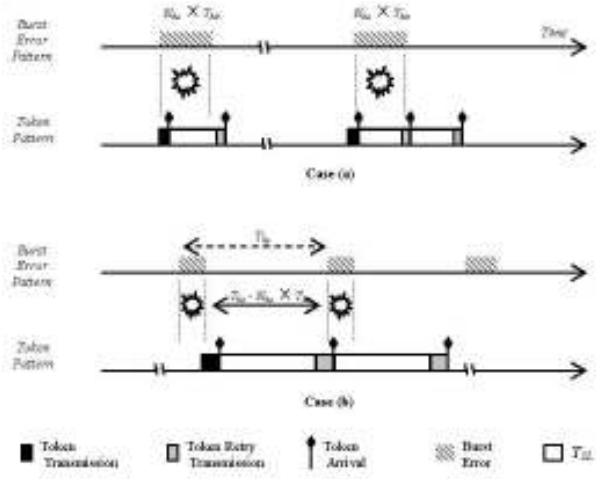

Figure 2: Illustration of possible burst effects on token transmission: Case (a) The effect of burst length, and Case (b) The effect of separation between successive bursts.

5. 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 equation (3)) can be re-written with the addition of worst-case error

components in abstract manner as shown in equation (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.

The single bit error effect is included by regarding the maximum number of error events (the first ceiling function of both faulty components in equation (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 equations (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 equation (22), the first component in equation (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 equation (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: $\overline{max\ retry\ limit} \times (nh_i^k \times n + 1)$. 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 2 (T_{SL}+T_{tkn})'s multiplicand must not exceed, $\times (nh_i^k \times n + 1)$, 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 equation (21) is not exact but sufficient, since faults not always induce a maximum overhead load.

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

$$+ \overbrace{\left\{ \underbrace{\left\lceil \frac{R_i^k}{T_e} \right\rceil}_{\text{single bit error component}} + \underbrace{\left\lceil \frac{R_i^k}{T_{eb}} \right\rceil \times O_{msg}}_{\text{error burst component}} \right\}}^{\leq \overline{max\ retry\ limit} \times (nh_i^k \times n+1)} \times C_m \quad \leftarrow \text{faulty message component}$$

$$+ \overbrace{\left\{ \underbrace{\left\lceil \frac{R_i^k}{T_e} \right\rceil}_{\text{single bit error component}} + \underbrace{\left\lceil \frac{R_i^k}{T_{eb}} \right\rceil \times O_{tkn}}_{\text{error burst component}} \right\}}^{\leq 2 \times (nh_i^k \times n+1)} \times (T_{SL} + T_{tkn}) \quad \leftarrow \text{faulty token component}$$

(21)

Where

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

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

$m = 1, 2, ..., \text{max\_retry\_limit}$

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

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

* 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.

To solve an equation like (21), a recurrent relation [3] is produced:

$$r_i^{k(n+1)} = nh_i^k \times T_{cycle}^k + Ch_i^k + \overbrace{\left\{ \left\lceil \frac{r_i^{k(n)}}{T_e} \right\rceil + \left\lceil \frac{r_i^{k(n)}}{T_{eb}} \right\rceil \times O_{msg} \right\}}^{\leq \overline{max\ retry\ limit} \times (nh_i^k \times n+1)} \times C_m + \overbrace{\left\{ \left\lceil \frac{r_i^{k(n)}}{T_e} \right\rceil + \left\lceil \frac{r_i^{k(n)}}{T_{eb}} \right\rceil \times O_{tkn} \right\}}^{\leq 2 \times (nh_i^k \times n+1)} \times (T_{SL} + T_{tkn}) \quad (24)$$

Where $r_i^{k(0)}$ is considered an initial value (usually the value of C_m). The recurrence procedure will be stopped whenever $r_i^{k(n+1)}$ is equal to $r_i^{k(n)}$ and this will be the value of the worst-case response time R_i^k.

6. Conclusions

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. Single bit and burst errors with bounded features are considered the consequence of the transient faults. 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 behaviours. Also, it enables analyzing the effect of these sources on the WCRT simultaneously. This work 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 always adapted without taking into account the fault characteristics.

REFERENCES

[1] kerdal Ö., Claesson V., Fredriksson L., "Error detection and Handling," *PÅLBUS* Task 10.5. Revision 4. June 2000.

[2] Broster I., "Flexibility in Dependable Communication," *PhD thesis*, Department of Computer Science, University of York, York, YO10 5DD, UK, Aug 2003.

[3] Burns, G. Bernat and I. Broster. "A Probabilistic Framework for Schedulability Analysis," *In the Proceedings of the 3rd International Embedded Software Conference* (EMSOFT03), pp. 1-15, 2003.

[4] Ferreira L., Alves M., and Tovar E., "Hybrid wired/wireless Profibus networks supported by bridges/routers," *in Proc. of WFCS*, Vasteras, Sweden, 2002.

[5] Hansson, H., Norström, C., and Punnekkat, S., "Integrating reliability and timing analysis of CAN-based systems," i*n Proc. 2000 IEEE Int. Workshop Factory Communication Systems* (WFCS'2000), Porto, Portugal, pp. 165–172, Sept. 2000.

[6] Li M. "Real-Time Communication in an Industrial Network – PROFIBUS," *PhD Thesis* nº1586. École Polytechnique Fédérale de Lausanne (EPFL).1996.

[7] Navet, N., Song, Y.-Q., and Simonot, F., "Worst-case deadline failure probability in real-time applications distributed over controller area network," *J. Syst. Architect.*, vol. 7, no. 46, pp. 607–617, Sept. 2000.

[8] Nieuwenhuyse K. V., Behaeghel S. "Timing performance of a hybrid wired/wireless," *a dissertation in industrial engineering*, Polytechnic Institute of Porto (ISEP/IPP)-Portugal. June 2003.

[9] Pinho L. M., Vasques F., and Tovar E., "Integrating inaccessibility in response time analysis of can networks," *in Proc. 2000 IEEE Int. Workshop Factory Communication Systems* (WFCS'2000) Porto, Portugal, Sept. 2000, pp. 77–84.

[10] Profibus–PROFIBUS Technology and Application - System Description. http://www.profibus.com. 2002

[11] Punnekkat S., Hansson H., and Norström C., "Response time analysis under errors for CAN," *in Proc. IEEE Real-Time Technology and Applications Symp.* (RTAS 2000), June 2000, pp. 258–265.

[12] Rufino J., "Computational System for Real-time Distributed Contro," *PhD thesis*, Universidade T´ecnica de Lisboa Instito Superior T´ecnico, July 2002.

[13] Rufino J. And Verissimo P., "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.

[14] Rufino J. And Verissimo P., "A study on the inaccessibility characteristics of ISO 8802/5 Token-Ring LANs," *Technical Report RT/24-92*, INESC, Lisboa, Portugal, February 1992.

[15] Rufino J. And Verissimo P., "A study on the inaccessibility characteristics of the FDDI LANs," *Technical Report RT/25-92*, INESC, Lisboa, Portugal, March 1992.

[16] Tindell, K. W. and Burns, A., "Guaranteed message latencies for distributed safety-critical hard real-time control networks," *Tech. Rep. YCS229*, Dept. Comput. Sci., Univ. York, York, U.K., June 1994.

[17] Tovar E., "Supporting Real-Time Communications with Standard Factory-Floor Networks," P*hD Thesis in Electrical and Computer Engineering*. 1999

[18] Tovar, E. and Vasques, F., "From Task Scheduling in Single Processor Environments to Message Scheduling in a Profibus Fieldbus Network," *In Lecture Notes in Computer Science*, No. 1586, pp. 339-352, WPDRTS'99, April 1999.

[19] Tovar, E. and Vasques, F., "Real-Time Fieldbus Communications Using PROFIBUS Networks," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 6, pp. 1241-1251, December 1999.

[20] Verissimo P., Rufino J., and Rodrigues L., "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.

[21] Verissimo P., Rufino J., Ming L., "How hard is hard real-time communication on field-buses?" *In Digest of Papers, the 27th International Symposium on Fault-Tolerant Computing Systems*, pages 112–121, Seatle, Washington, USA,June,1997