Minimizing Token-Bus Inaccessibility Through Network Planning and Parameterizing*

José Rufino, Paulo Veríssimo
Technical University of Lisboa – INESC†
Lisboa – Portugal
e-mail:...ruf@inesc.pt, paulov@inesc.pt

Abstract

Local area networks have long been established as a basis for distributed systems. Continuity of service and bounded and known message delivery latency are requirements of a number of applications, which are imperfectly fulfilled by standard LANs.

One key issue with this regard is that LANs are subject to failures, namely partitions. Every single LAN displays a number of causes for partition, not all of them of physical nature: bus contention, ring colapse, token loss, etc. Since most applications can live with temporary glitches, reliable real-time operation is possible on non-replicated LANs, provided that these temporary partitions are time-bounded. We call these periods of inaccessibility, to differentiate from classical partitions.

This paper investigates how network planning and parameterizing policies can contribute for minimizing inaccessibility periods in ISO 8802/4 Token-Bus LANs.

1 Introduction

In reliable real-time systems, the fundamental requirement of communications is that there be a bounded and known message delivery latency, in the presence of disturbing factors such as overload or faults. For applications where this requirement is very strict (eg. life-critical ones), specialized space-redundant architectures, like point-to-point graphs [1] or multiple LANs [2], are clearly the solution. These architectures are however costly and complex.

When that requirement is not so strict, standardised local area networks (ISO 8802 and FDDI) can represent a very important design alternative since the utilization of existing technology allow to implement a cost-effective networking infra-structure. Nevertheless, LANs present shortcomings with regard to continuity of service and determinism in transmission delays, if used without special measures.

Methodologies and algorithms to reliably enforce real-time behaviour in non-replicated local area networks were discussed in a recent work [3]. Previous existing studies with this regard have essentially addressed network parameterizing for achieving bounded access delays in frame transmission, given the worst case load conditions [4, 5, 6, 7]. However, they are helpless at representing the LAN behaviour, when faults occur. In that case, it is necessary to study the patterns for omission failures (eg. number of consecutive omission failures) and for partitions.

In [8] we have presented an exhaustive study on the behaviour of standard ISO 8802/4 Token-Bus LAN with regard to partitions. Token-Bus, given its connection with MAP, assumes a very important role in control and automation[1], where real-time, reliability and accessibility are a must. This paper uses the results of [8] to investigate network planning and parameterizing policies, having in mind the control of network partitions. Furthermore, shortcomings of the standard specification with this regard, falling out from the scope of [8] but extremely relevant for the definition of minimizing strategies, will be analysed.

2 Network Partitions

A network is partitioned when there are subsets of the nodes which cannot communicate with each other[2]. In this sense, a single LAN displays a number

*Paper presented at *EFOC/LAN92 the Tenth Annual European Optical Communications and Networks Conference*, Paris, France, June 24-26 1992 ©1992 IGI Europe

†Instituto de Engenharia de Sistemas e Computadores, R. Alves Redol, 9 - 6^o - 1000 Lisboa - Portugal, Tel.+351-1-3100000. This work has been supported in part by Junta Nacional de Investigação Científica e Tecnológica (JNICT) through Programa Ciência.

[1] Manufacturing Automation Protocol.

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

of causes for partition, not all of them of physical nature, like bus failure (cable or tap defect) or transmitter/receiver defects: bus contention, ring disruption; token loss; etc. Some LANs have means of recovering from some of these situations, and can/should be enhanced to recover from the others, if reliable real-time operation is desired.

However, the recovery process takes time, so in the meantime the LAN is partitioned. Let us call them periods of *inaccessibility*, to differentiate from classical partitions. The definition of inaccessibility in [9] is summarised here:

> *Certain kinds of components may temporarily* refrain *from providing service, without that having to be necessarily considered a failure. That state is called* **inaccessibility***. It can be made known to the users of the component; limits are specified (duration, rate); violation of those limits implies permanent failure of the component.*

To achieve the control of partitions one must first assure that all conditions leading to partition are recovered from. Then, one needs to show that all the inaccessibility periods are time-bounded and determine the upper bound. This worst-case inaccessibility figure will then be added to the worst-case transmission delay in the absence of faults, to obtain a consolidated transmission delay bound.

Exception made to physical nature failures, the standard Token-Bus is provided with built-in mechanisms for handling all the other inaccessibility causes. Tolerance against medium failures has to be supported through custom-implemented extensions to the standard. For example, in [10] it is described a glitch-free method for real-time switch-over between buses of a dual-media token-bus.

A qualitative analysis of the standard Token-Bus error handling mechanisms has been provided in [11]. Surprisingly, and to the best of the author's knowledge, a quantitative analysis of the 8802/4 error handling mechanisms had never been presented, until recently.

3 Inaccessibility Behaviour

In fact, a quantitative characterization of ISO 8802/4 Token-Bus inaccessibility behaviour was drawn in [8]. In this study we established a set of easy-to-use formulas that allow the prediction of the best and worst-case durations of inaccessibility periods, i.e. those intervals in the Token-Bus operation when the LAN does not provide service, although not being failed. The analysed scenarios were those foreseen in the standard specification. The study has been oriented for networks employed in an opened fashion, using *IEEE registered addresses* [12] and where stations are connected to the power supply lines in an "ad-hoc" way [3].

The results from the application of those formulas to a target Token-Bus network are presented in Table 1, where N_{st} represents the number of currently active stations. The network length is $C_l = 500m$ and the maximum number of stations $N_{ST} = 32$. We have additionally assumed that standard $l_{add} = 48$ bits long addresses are used. The data rate is $5Mbps$.

Scenario		t_{ina} (ms)	
		min.	*max.*
Station Join	*No Responses*	0.073	0.100
	No Contention	0.118	0.145
	Contention	0.382	4.612
Multiple Joins $(N_{st\rightarrow join} = 30)$		0.363	139.999
Station Leave		0.056	
Multiple Leaves $(N_{st} = 32)$		0.112	1.674
No Successor		0.306	
Token Loss		1.717	5.794
Multiple Fails $(N_{st} = 32)$		0.612	4.896
Station Group Fail		0.521	5.176
Multiple Group Fails $(N_{st} = 32)$		5.697	51.762

Table 1: Original Inaccessibility Times ($5Mbps$, $l_{add} = 48$, $t_{SD} = 11\mu s$, $t_{Slot} = 27\mu s$)

The obtained figures evidence the three-fold nature of inaccessibility with sets of values either much higher or much lower than average. The recognition that a longest inaccessibility period occurs whenever there is contention between stations (see Table 1: station joins, token loss and particularly multiple joins and multi-group failure scenarios) provides the basis for an investigation of inaccessibility minimizing strategies. Let us therefore review our 8802/4 error-handing performance model having this goal in mind[4].

Station Joins

Station joins are initiated by the then-current token holding station, through the opening of admission windows, and its operation varies slightly depending

[3] Meaning that the full address string may be required for station discrimination and that unrestricted failure modes can occur.

[4] The following discussion will assume familiarity with the 8802/4 Token-Bus operation. For further reading, please see [13].

on whether or not this station is the one having the lowest address. Accordingly, the duration of the associated inaccessibility period is, as derived in [8], given either by equation (1) or (2).

$$t_{ina \leftarrow join1} = t_{SD} + t_{SS1} + t_{Slot} + t_{rcp} \tag{1}$$

$$t_{ina \leftarrow join2} = t_{SD} + t_{SS2} + 2 \, . \, t_{Slot} + t_{rcp} \tag{2}$$

where:

- ◇ t_{SS1} and t_{SS2} are, respectively, the durations of the *solicit_successor_1* and *solicit_successor_2* MAC[5] frames.
- ◇ t_{Slot} — represents an aggregate variable accounting for MAC sub-layer intrinsic performance and network size. The *slot time* is usually expressed in octect times, due to its formal definition, provided in [13]. In this work, for simplicity, we use its value expressed in plain time, as given by equation:

  $$t_{Slot} = 2 \, . \, (t_{SD} + t_{PD}) \tag{3}$$

  - t_{SD} - is the station delay, as defined in [13]. This variable accounts for the intrinsic performance of each particular MAC VLSI implementation.
  - t_{PD} - is the worst case end-to-end propagation delay of the physical layer. This variable accounts for the network cable propagation delay, plus the modem delays (at both transmit and receiver ends) and the regenerative repeater delay, whenever used. For the length-dependent cable propagation delay a typical value of 5 $\mu s/km$ is usually assumed.
- ◇ t_{rcp} — accounts for the duration of the *resolve contention process* and its exact value depends on whether or not a contention between stations occurs and on how quickly this contention is resolved. All the possible situations are described in equation (4). In the definition of (4) we have considered that responses from potential successors are only received, at the process initiator, near the end of each window opening period and that, only 1/4 of the contention rounds occur at the fourth slot time. The durations of the MAC *set successor* and *resolve contention* frames, taking part in the process, are represented by t_{SSF} and t_{RC}, respectively.

$$t_{rcp} = \begin{cases} 0 & \text{no response} \\ t_{SSF} & \text{no contention} \\ \displaystyle\sum_{nb_{rounds}} (t_{RC} + 4 \, . \, t_{Slot} + \frac{t_{SSF}}{4}) & \text{contention} \end{cases} \tag{4}$$

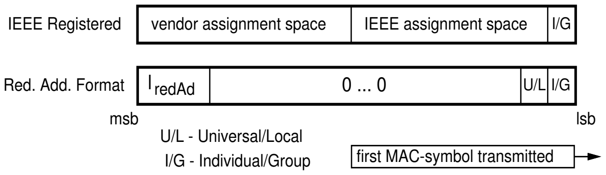

Figure 1: Structure of station address

Equation (4) evidences why contention, on ring entry demand, may seriously affect token-bus accessibility. In first place, the duration of each individual contention round is always larger than any other value taken by t_{rcp} when contention does not occur. Secondly, a large number of rounds, up to a maximum of 24, may be required for the discrimination of stations using the full 48 bit address string, in the definition of the station's unique identifier. This is a situation commonly found in installations using *IEEE registered addresses* [12].

However, let us assume that the station's unique identifier may be represented by a string using only the high-order l_{redAd} bits of the full 48 bit address string (see Figure 1). Since a station starts to use their high-order address bits in the resolution of collisions [13], the maximum number of rounds required for the execution of the *resolve contention process* is now given by expression:

$$nb_{rounds} = \left\lceil \frac{l_{redAd}}{2} \right\rceil \tag{5}$$

where $\lceil \;\; \rceil$ represents the *ceil* function[6].

For a single station join, the worst-case inaccessibility time, that we signal with superscript ^{wc}, is under these circumstances, given by equation (6), derived from equation equations (2) and (4), with the assumption that competing stations systematically answer in the fourth window of a *resolve contention* round.

$$\begin{aligned} t^{wc}_{ina \leftarrow join} \quad = \quad & t_{SD} + t_{SS2} + 2 \, . \, t_{Slot} + \\ & \left\lceil \frac{l_{redAd}}{2} \right\rceil \, (t_{RC} + 4 \, . \, t_{Slot} + t_{SSF}) \end{aligned} \tag{6}$$

Contention arises when multiple potential successors reply to a successor query, in the same response window. The raw *solicit successor procedure* [13] always adds new stations to the logical ring in a one by one basis. Station configuration options allows to dither these entries, at least by a given number of token rotations, or cluster them, into the same token

[5] Medium Access Control.

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

rotation, through successive executions of the *solicit successor procedure*. If this scenario is analysed "per si", then clearly, the most suitable solution is the one that transforms multiple join requests into a sequence of independent join actions. However, such a solution presents serious drawbacks when stations demanding ring entry are intermixed with failed stations. This is a complex scenario, that calls for a more thorough analysis, to be performed later on this section, after having analysed failure scenarios.

Token Loss

The inaccessibility period concerning the recovery of a lost token has two main contributions, as described by equation:

$$t_{ina \leftarrow tkloss} = t_{BIdle} + t_{tcp} \tag{7}$$

- The first term on this expression concerns the error detection latency and accounts for the time elapsed between token loss and the beginning of the token claim process. It corresponds to the value loaded into the *Bus Idle Timer*. Usually this timer assumes the value of seven slot times. The exception is the station having the lowest address in the network, where the *Bus Idle Timer* is loaded with only six slot times.

$$t_{BIdle} = \begin{cases} 6 \,.\, t_{Slot} & \text{lowest station present} \\ 7 \,.\, t_{Slot} & \text{lowest station failed} \end{cases} \tag{8}$$

- The second term represents the duration of the *token claim process*. Any station where the *Bus Idle Timer* expires, initiates a recovery procedure by issuing a *claim_token* frame whose length depends on the two most significant bits of the station address [13]. Stations passing each claim token contention round, wait during one slot time, before issuing a new *claim_token* frame, whose duration depends on the next two unused address bits. The process ends after the winning station has used all the bits of its address, plus two randomly chosen bits.

$$t_{tcp} = \sum_{i=1}^{(l_{add}/2)+1} (t_{CF}(i) + t_{Slot}) \tag{9}$$

  - $t_{CF}(i)$, represents the duration of a *claim_token* frame issued in round i, and is given by:

    $$t_{CF}(i) = t_{HdTr} + 2 \times TwoBit_{add}(i) \times t_{Slot}$$

    where t_{HdTr} stands for the duration of the MAC header/trailing sequence. The $TwoBit_{add}$ value depends of each particular address bit pair, and can range from zero to three.

The previously assumed reduction in the number of significant bits in the station's unique identifier allows reducing the duration of the token claim process. Due to protocol design, token claiming still needs to be performed throughout the same, fixed number of rounds [13]. However, the duration of those rounds that do not participate in station's discrimination can be shortened, by forcing *claim token* frames transmitted in such rounds to have the minimum duration. This is accomplished by clearing the non-significant bits in the address string. The two low-order address bits define the address type (individual/group) and its administration (universal/local). For the computation of the corresponding worst-case token claim round duration, we assume that both these bits are set. Therefore, the worst-case inaccessibility duration for token loss recovery is given by:

$$\begin{aligned} t^{wc}_{ina \leftarrow tkloss} = \; & 7 \,.\, t_{Slot} + \\ & \left(\left\lceil \tfrac{l_{redAd}}{2} \right\rceil + 2 \right) \; (t_{HrTr} + 7 \,.\, t_{Slot}) + \\ & \left\lfloor \tfrac{l_{add} - l_{redAd} - 2}{2} \right\rfloor \; (t_{HrTr} + t_{Slot}) \end{aligned} \tag{10}$$

where $\lfloor \;\; \rfloor$ represents the *floor* function[7].

Station Failures

In this section we will consider a set of scenarios where a whole set of stations fail, at the same time, due to some common cause. Common mode failures are realistic enough to be considered. For instance, in a network where several stations are connected to the same power supply line, a failure or a shutdown in this line will bring all these stations down. Three hypothesis are taken into account:

i) *All the failed stations are grouped in a single cluster of adjacent stations, within the token-passing order* – recovery from this error situation uses two variants of the basic *solicit successor procedure*. The first mechanism, known as *who follows query*, specifically looks for the presumable successor of the failed station. Since by assumption this station is also failed, the method does not succeed and a procedure where all the active stations are solicited as potential successors (*solicit any*) is entered. The time spent in all these recovery actions was derived in [8]. It is given by equation (11), where t_{rcp} represents the duration of an eventual resolve contention process, as given by equation (4).

$$\begin{aligned} t_{ina \leftarrow gfail} \; = \; & t_{SD} + 2 \,.\, (t_{TK} + t_{WF}) + \\ & 10 \,.\, t_{Slot} + t_{SS2} + t_{rcp} \end{aligned} \tag{11}$$

[7] The *floor* function $\lfloor x \rfloor$ is formally defined as the greatest integer not greater than x.

ii) *There are multiple clusters of failed stations* – bypass of failed stations is achieved through successive executions of the *solicit any procedure*. Clearly, the inaccessibility figures obtained in this case are always worst than the ones displayed by the single cluster scenario.

iii) *None of the failed stations are adjacent in the token-passing order* – bypass of failed stations is simply achieved through successive executions of the *who follows* query. The corresponding inaccessibility time grows linearly with the number of failed stations [8] but, in certain settings, it displays worst-case inaccessibility figures lower than the single cluster scenario. However, no guarantees can be given that failed stations are always intermixed with correct stations.

This means that a policy aiming the minimization of inaccessibility times shall prevent multiple occurrences of failed stations/groups within the token path, by transforming them into a single group of failed stations. Implementation of such policy will be discussed in next section.

Execution of the *solicit any* procedure is typically affected by collisions among responders. Their resolution is supported by the aforementioned *resolve contention process*. Because of this, single group failure recovery benefits from the consideration of the reduced addressing assumption, in the sense that duration of this process is shortened. The value of t_{rcp} that should be considered, in equation (11), for the obtainment of the corresponding worst-case inaccessibility duration is, therefore, given by:

$$t^{wc}_{rcp} = \left\lceil \frac{l_{redAd}}{2} \right\rceil . (t_{RC} + 4 . t_{Slot} + t_{SSF}) \qquad (12)$$

Station Failures and Joins

Let us now analyse how the two aforementioned variants of the raw solicit successor procedure — essentially oriented to cope with station failures — can be affected in their execution, by the presence, in the network, of stations demanding logical ring membership.

The *who follows query* is a procedure intended to restore logical ring integrity upon the failure of isolated stations, within the token passing order. It simultaneously inquires all the active ring members in an attempt to locate the successor of the failed station. Since this query is not extended to non-ring members its result will always be the same, whether or not there are stations wanting to join the logical ring.

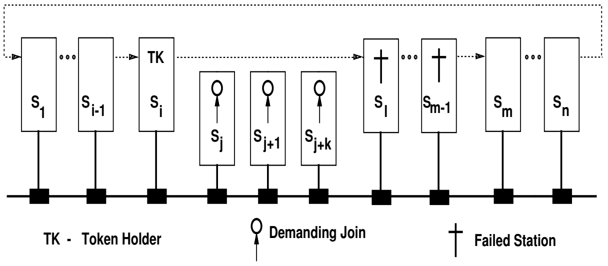

Figure 2: Station Failures intermixed with joins

Conversely, the *solicit any* procedure performs an extensive query in their looking for new successors. All the active stations are allowed to respond, non-ring members included. Responses from these last stations can only interfere with logical ring recovery if the set of failed stations and candidates to logical ring membership are adjacent in the token-passing order, as depicted in Figure 2. In order to thoroughly analyse such a scenario, let us recall from [13] the set of actions to be performed:

- After initiating the process, with the transmission of a *solicit_successor_2* MAC frame addressed to itself, the token holder waits for possible responses during two consecutive windows.
- Stations with addresses below the token holder respond in the first window.
- Station with addresses above the token holder delay possible responses by one slot time. If these stations hear no activity in the bus during this period, they issue their responses during the second window. Otherwise, they abandon the process.
- Stations that belong to the logical ring, leave untouched the *next station address* variable[8].
- Stations not belonging to the logical ring, store the address of the token holder station, as their potential successor, in the *next station address* variable. This event will be of great importance to our analysis.

Eventual contentions, occurring within the same response window, will be resolved through a *resolve responders algorithm* common to the raw *solicit successor procedure* and to the aforementioned variants [13]. Usually, the *resolve responder algorithm* is won by the contending station with the highest address. In the scenario presented in Figure 2 station $\mathcal{S}_j$ will, probably, be elected as the successor of station $\mathcal{S}_i$ and will receive the token afterwards.

Token reception signals that station it has won the contention. Should it be a just-arrived station, two important actions decisive for subsequent ring management operations, are performed upon this event:

[8] This procedure aims to preserve the remaining structure of the former logical ring.

- The *inter_solicit_counter* is set to zero, which enables the station to immediately open admission windows, provided that the *token rotation timer for ring maintenance* has not expired [13].
- The *token rotation timer for ring maintenance* is set to a special value, furnished by station management entities, known as *ring maintenance timer initial value* [13].

This means that stations just-arrived to the logical ring can either immediately start a new *solicit successor procedure* or postpone it until a forthcoming opportunity, depending on the value initially assigned to the ring maintenance timer. Let us assume that the initial value of the ring maintenance timer is cleared to zero, meaning that no admission window will be opened. The *inter_solicit_counter* is (re)initialised with a parameter furnished by station management entities, known as *max_inter_solicit_count*, defining the number of token rotations that, at least, must elapse before soliciting new ring members. The token is then passed to the station that was previously set as this station's successor, i.e. the *solicit any* initiator. In the example of Figure 2 the token will be exchanged between stations $\mathcal{S}_j$ and $\mathcal{S}_i$, and all the remaining former ring members (i.e. from $\mathcal{S}_m$ till $\mathcal{S}_{i-1}$, in the token passing order) will be skipped. This situation lasts until a new admission window is opened. Successive solicitation of successors, at least interleaved by *max_inter_solicit_count* token circulations, will promote the entering of all the remaining $\mathcal{S}_{j+1}$ to $\mathcal{S}_{j+k}$ stations, in a one by one basis. The recovery process ends when a final window opening brings back to the ring all the skipped stations.

Although logical ring integrity is restored, we must stress that such a situation is unacceptable for real-time systems, since it leads to uncontrolled partitions. First, an acceptable upper bound cannot reliably be placed in the duration of those chained recovery procedures. Secondly, there are stations in the network with a view substantially different of inaccessibility: while the station initially leading the recovery process ($\mathcal{S}_i$ in the given example) gets a regular access to the network, all the skipped stations stay inaccessible, during an unacceptable and hardly bounded number of token rotations.

Conversely, if the station is configured to allow the immediate opening of admission windows, all the previously recovery actions are performed within a single token rotation and its duration can be reliably upper bounded. The corresponding inaccessibility time is given by the sum of two contributions. The first one accounts the time required to elect a successor for the station that initiates the recovery procedure, as given by equation (11). The other accounts the time spent in all the successive contending $N_{st\rightarrow join}$ station joins, whose individual durations are given by equation (6). The overall time is thus given by:

$$t^{wc}_{ina\leftarrow jgfail} = t^{wc}_{ina\leftarrow gfail} + N_{st\rightarrow join} \,.\, t^{wc}_{ina\leftarrow join} \quad (13)$$

Multiple Station Joins

Assuming that admission windows opening can be performed by just arrived stations, it has been shown in [8] that the worst-case inaccessibility time due to the join of $N_{st\rightarrow join}$ stations is given by:

$$\begin{aligned} t^{wc}_{ina\leftarrow mjoin} \;=\; & (N_{st\rightarrow join} - 1) \,.\, t^{wc}_{ina\leftarrow join} + \\ & 2 \,.\, (t_{SD} + t_{SS2}) + 4 \,.\, t_{Slot} + t_{SSF} \end{aligned} \quad (14)$$

4 Minimizing Strategies

The ISO 8802/4 Token-Passing Bus is a local area network (LAN) which has gathered a growing attention after it has been selected as the communication infra-structure of MAP, the Manufacturing Automation Protocol, becoming then a standard for interconnection and interworking in the factory floor [14]. Taking into account the hierarchical structure of MAP we can devise the existence of small subnetworks where openness can be restricted, but which are demanding in real-time requirements. Could openness be traded with shorter inaccessibility times and we will have a network with improved performability. Notice that we need not loose connectivity; communication with other subnetworks, for example for cell programming, can be assured by gateways above MAC.

For a network with a small number of nodes, only a few bits in the address rather than the full 48 bit string are needed for station's unique representation and, in consequence, a policy favoring a fast discrimination between contending stations can be implemented.

Reduced Addressing Policy: *The station's unique identifier must be defined within the most significant l_{redAdd} bits of the address string; the two least significant bits of the address string are set as required; all the remaining bits must be cleared to zero.*

No constraints are placed in the definition of address qualifying bits, thus allowing the assignment of individual or group addresses, universally or locally administered. This policy is not hard to implement. It merely requires the modification of the station's LAN physical address. Station addresses are usually held in some sort of non-volatile storage element that can be easily changed by network engineers[9]. The major consequence of such a modification is that, whether or not a *IEEE registered address* was formerly assigned to the station, address uniqueness is no longer globally ensured[10].

Planning of station connection to the power supply network is also a mandatory measure for minimizing inaccessibility time bounds. It aims at avoiding multi-failure scenarios. Should a set of stations be able to fail simultaneously, they must be clustered with adjacent addresses.

Address Ordering Policy: *Stations plugged within the same or interconnected power supply line must have adjacent addresses in the token-passing ordering.*

This policy is based on the assumption that failures or shutdowns in power supply lines are the main cause of common mode failures. Circuit breakers and line interconnection points are thus considered single-points of failure for all the stations connected on their downstream links. A *target token-passing path* shall be imposed over this topology of power lines. The assignment of station addresses shall be performed in order to guarantee that the network logical ring is coincident with the aforementioned target path. This means that the assignment of an address to a station depends on the location of the corresponding connection with the power supply lines and on the addresses previous assigned to both nearest neighbors.

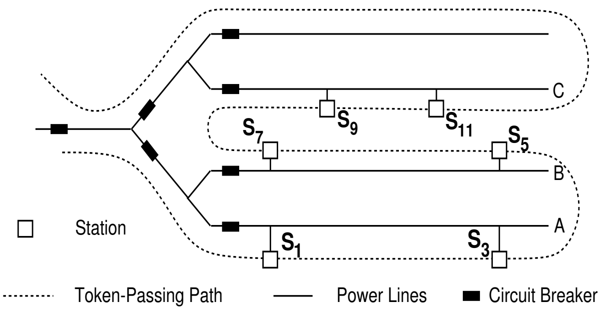

Figure 3: Methodology for multi-failures control

For example, a station with address $\mathcal{S}_2$ could be plugged on branch A, of the power supply network of Fig. 3, between stations $\mathcal{S}_1$ and $\mathcal{S}_3$. Likewise, for the plugging of a station to branch B, between $\mathcal{S}_5$ and $\mathcal{S}_7$, it should be assigned address $\mathcal{S}_6$. A station with address $\mathcal{S}_4$ can be plugged either to branch A after $\mathcal{S}_3$ or to branch B, before $\mathcal{S}_5$.

Finally, a policy aiming at the prevention of ill effects in window opening and the reduction of its influence on ring operation is defined.

Window Opening Policy: *Stations must be parameterized to perform window opening, upon ring entry, through the definition of the* ring maintenance timer initial value. *Stations must be parameterized, through the definition of the* max_inter_solicit_count, *to interleave further soliciting queries with a period much higher than network message delivery latency.*

Scenario		t_{ina} *(ms)*	
		min.	*max.*
Station Join	*No Responses*	0.073	0.100
	No Contention	0.118	0.145
	Contention	0.382	0.852
Multiple Joins ($N_{st\rightarrow join}$ = 30)		0.363	25.811
Station Leave		0.056	
Multiple Leaves (N_{st} = 32)		0.112	1.674
No Successor		0.306	
Token Loss		1.717	2.716
Station Group Fail		0.521	1.228
Group Fail & Joins ($N_{st\rightarrow join}$ = 28)		—	25.084

Table 2: Effect of Minimizing Strategies on Inaccessibility Times ($5Mbps$, $l_{redAd} = 8$, $t_{SD} = 11\mu s$, $t_{Slot} = 27\mu s$)

5 Analytical Results

Let us consider the application of the aforementioned policies to a network in an industrial environment used in a closed fashion, for example, a small cell for real-time manufacturing control.

We consider $l_{redAdd} = 8$, which yields at most 256 stations (more than enough) in the cell network. The figures obtained are presented in Table 2. The multiple failure scenarios are omitted, as results from the application of the *Address Ordering Policy*. On the other hand, *Reduced Addressing Policy* brings down drastically the worst-case bounds of network inaccessibility. Compare for instance the $26ms$ for the multiple joins scenario against the original $140ms$, that are obtained for a massive join of $N_{st\rightarrow join} = 30$.

[9] With an eventual assistance from equipment manufacturer.

[10] Interconnection via MAC bridges should take the necessary precautions with this regard.

This study is interesting, on the grounds that it provides guidelines, to network managers, on where to act to reduce the inaccessibility time bounds and justifiably expect to achieve better performability, and thus better respect hard requirements for bounded delay.

6 Conclusions

To achieve reliable real-time operation of a local area network, a bounded delay requirement must be met. Most previous studies have addressed this issue by computing worst-case access/transmission delays only for normal LAN operation.

However, achieving the bounded delay requirement means, amongst other factors, ensuring continuity of service. LANs are subject to failures, namely partitions: if these are not controlled, the above mentioned requirement is not met. A single LAN displays a number of causes for partition, not all of them of physical nature: bus contention, token loss, etc. Recovering from these situations takes time and we have named those periods *inaccessibility*. Some noncritical applications can live with temporary glitches in LAN operation, provided that they are acceptable short. In these conditions, reliable real-time operation is possible on non-replicated LANs, through appropriate techniques[3].

In [8] we have presented an exhaustive quantitative study of the ISO 8802/4 Token-Bus inaccessibility. The figures presented illustrate the intervals on Token-Bus operation when the LAN does not provide service, although not being failed. This paper uses those results to investigate inaccessibility minimization strategies aiming the reduction of the longest periods. This study is interesting since it provides network managers with rules to improve the situation.

Our previous results, recapitulated in Table 1, outlined a serious problem often disregarded by designers when expecting hard real-time performance from token-bus: it can be inaccessible for periods in excess of $140ms$, and this figure is to be added to the worst-case transmission delay expected in the absence of faults.

The proposed minimization policies trade LAN openness with performability. The results are quite effective: an acceptably short figure, not exceeding $30ms$, for maximum inaccessibility time.

References

[1] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Prog. Lang. and Systems*, 4(3), July 1982.

[2] Flaviu Cristian. Synchronous atomic broadcast for redundant broadcast channels. Technical report, IBM Almaden Research Center, 1989.

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

[4] Dittmar Janetzky and Kym S. Watson. Token bus performance in MAP and Proway. In *Proceedings of the IFAC Workshop on Distributed Computer Protocol System*, 1986.

[5] R.Mangala Gorur and Alfred C. Weaver. Setting target rotation times in an IEEE Token Bus network. *IEEE Transactions on Industrial Electronics*, 35(3), August 1988.

[6] D. Dykeman and W. Bux. An investigation of the FDDI media-access control protocol. In *Proceedings of the EFOC/LAN Conference*, Basel, Switzerland, June 1987.

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

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

[9] P. Veríssimo and José A. Marques. Reliable broadcast for fault-tolerance on local computer networks. In *Proceedings of the Ninth Symposium on Reliable Distributed Systems*, Huntsville, Alabama-USA, October 1990. IEEE. Also as INESC AR/24-90.

[10] Paulo Veríssimo. Redundant media mechanisms for dependable communication in token-bus LANs. In *Proceedings of the 13th Local Computer Network Conference*, Minneapolis-USA, October 1988. IEEE.

[11] Thomas L. Phinney and George D. Jelatis. Error Handling in the IEEE 802 Token-Passing Bus LAN. *IEEE Journal on Selected Areas in Communications*, 1(5), November 1983.

[12] ISO. *Registration of MAC Addresses - IEEE Standards Office*, 1990.

[13] *ISO DIS 8802/4-85, Token Passing Bus Access Method*, 1985.

[14] *Manufacturing Automation Protocol Specification V2.1*, March 1985.