A Study on the Inaccessibility Characteristics of the FDDI LAN

INESC Technical Report RT 25-92

J. Rufino, P. Veríssimo

March 1992

A Study on the Inaccessibility Characteristics of the FDDI LAN

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

March 1992

Abstract

Local area networks have long been established as the 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. Most previous studies have addressed this issue by computing worst-case access/transmission delays only for normal LAN operation.

However, LANs are subject to failures, namely partitions. Since most applications can live with temporary glitches in LAN operation, an alternative approach is to quantify all these glitches or temporary partitions, that we named inaccessibility, and derive a worst-case figure, to be added to the worst-case transmission delay in absence of faults. In these conditions, reliable real-time operation is possible on non-replicated LANs. This paper does an exhaustive study of the inaccessibility characteristics of the ISO 9314 FDDI LAN.

1 Introduction

Local area networks have long been established as the basis for distributed systems. The several variants of standardised LANs (ISO 8802 and FDDI) have different mechanisms to control access to the medium and recover from errors. Continuity of service and determinism in transmission delay are requirements of a number of applications, specially in the fault-tolerance and real-time area, which are unperfectly fulfilled by these LANs, if used without special measures. A number of authors have studied problems such as priority inversion [1], probability of meeting estimated access times [2, 3], extensions for medium failure resiliency through redundancy [4], potential lack of determinism [5].

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. When the requirement is very strict (eg. for life-critical applications), specialized space-redundant architectures are the solution: point-to-point graphs [6] or multiple LANs [7].

These solutions are however costly and complex. In spite of their limitations, standard LANs are a very important design component. It is worthwhile investigating if the real-time requirement can be reliably met in local area networks that are not replicated, except for eventual medium redundancy — at the electrical signalling level. To achieve reliable real-time communication, three fundamental conditions must be validated:

*Instituto de Engenharia de Sistemas e Computadores, R. Alves Redol, 9 - 6º - 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. bounded delay from request to transmission of a frame[1], given the worst case load conditions assumed;
2. message[2] delivery despite the occurrence of omission failures (eg. lost frames);
3. control of partitions.

Most of the existing studies with this regard have addressed point 1 [2, 8, 9, 10]. 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. Uncontrolled omissions and partitions are a source of asynchrony and inconsistency. This is unacceptable for most systems, let alone real-time ones. Point 2 is addressed under the scope of the LLC type 3 service for point to point interactions. However, it has been practically disregarded for broadcast or multicast interactions. One exception is a modified token-ring mechanism described in[11]. Point 3, when regarding non-replicated networks, and to the best of authors' knowledge, only recently has deserved some attention.

All three points have been addressed in [12] for LANs in general, while point 3 has been specifically addressed in [13] for Token-Bus LANs. A study on the behaviour of an ISO 9314 *Fiber Distributed Data Interface* (FDDI) with regard to partitions is the central issue of this paper. Although originally conceived for the computer center environment, the FDDI LAN presents characteristics that made it attractive for applications in the control and automation arena, where real-time, reliability and accessibility are a must.

This study contributes to a better understanding of the ISO 9314 FDDI LAN operation having those attributes in mind.

2 Controlling Partitions

A network is partitioned when there are subsets of the nodes which cannot communicate with each other[3]. In this sense, a single LAN displays a number of causes for partition, not all of them of physical nature, like bus failure (cable or tap defect): bus contention, ring disruption, transmitter or receiver defects; 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. A solution to the problem of controlling partitions was presented in [12]. It is based on a very simple idea: if one knows for how long a network is partitioned, and if those periods are acceptably short, real-time operation of the system is possible.

Let us call them periods of *inaccessibility*, to differentiate from classical partitions. The definition of inaccessibility in [14] 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.*

This is not hard to implement, as shown in [12]. To achieve it one must first assure that all conditions leading to partition are recovered from. For example, tolerance to one medium failure is directly assured in the dual FDDI ring [15].

[1] LAN level information packet.

[2] User level information packet.

[3] 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.

Then, one needs to show that all the inaccessibility periods are time-bounded and determine the upper bound. The study of ISO 9314 FDDI inaccessibility is presented next. The scenarios described in the following sections are essentially the inaccessibility periods foreseen in the standard specification. The figures presented illustrate the intervals in the network operation when the LAN does not provide service, although not being failed.

The study yields interesting conclusions, like: a figure for the worst case duration of inaccessibility (to be added to a worst-case access time...); the existence of some periods much longer than average; strategies to reduce the longest periods are possible.

3 The FDDI LAN

The *Fiber Distributed Data Interface* (FDDI) a high speed LAN [16] developed under the auspices of ANSI[4] within its X3T9.5 committee, and approved as international standard ISO 9314.

The FDDI was originally conceived to be used either as a backend network within the computer center environment or as a backbone network for LAN interconnection. The original FDDI was supported on multi-mode fiber, needed to accommodate a high data rate transfer. Along these last years the FDDI has evolve through the definition of extensions in order to support isochronous traffic (FDDI-II), and to expand their geographic scope for the metropolitan area, using either single-mode fiber [17] or dedicated lines [18]. On the other hand a serious effort has been placed in bring FDDI to the desktop environment through the use of either low-cost fiber or shielded/unshielded cooper cables [19, 20].

The FDDI uses a timed-token protocol access method. Two classes of service are provided: synchronous and asynchronous. Synchronous frames can be transmitted whenever a station receives the token. Asynchronous frames can only be transmitted using the bandwidth not consumed by synchronous class. Multiple levels of priorities can be assigned within the asynchronous class. The timeliness characteristics of the FDDI LAN presumably allows its utilization in control and automation, as frontend networks.

The performance of the FDDI access method has been widely studied [21, 22, 23, 24, 25]. Most of these studies assume that the network always operates normally, neglecting the influence of inaccessibility. However, in the presence of faults corrective actions must be performed and in consequence the ring operation is affected, usually severely, in the sense that it can no longer ensure the calculated frame transmission delay bound.

A comprehensive analysis of the FDDI error handling mechanisms is provided in [26]. However, it is essentially qualitative. This work presents an exploratory quantitative analysis of the FDDI error handling mechanisms. Future evolution of the FDDI specification, particularly in issues concerning fault detection and fault isolation policies, as well as network reconfiguration, may lead to modifications of the model to enhance its adherence to the standard.

MAC Frame	Symbol	Duration (μs)
Token	t_{Tk}	0.88
Frame header/trailing	t_{HrTr}	2.24
Claim Token	t_{CLM}	2.56
Beacon	t_{BCN}	3.04

Table 1: Duration of MAC FDDI Protocol Data Units ($l_{add} = 48$)

[4] American National Standards Institute.

MAC Timers	Symbol	Default Values
Token Holding Timer	t_{THT}	
Transmission Valid Timer	t_{TVX}	2.5 ms
Token Rotation Timer	t_{TRT}	T_min=4.0 ms
		T_max=165.0 ms
SMT Timers	Symbol	Default Value
Entity Coordination	t_{TEC}	
Physical Connection	t_{TPC}	
Idle Detection	t_{TID}	
Noise Event	t_{TNE}	
Ring Management	t_{TRM}	
Timing Values	Symbol	Default Value
PCM signalling timeout	T_Out	100 ms
No operational ring	T_Non_Op	1.0 s
Stuck beacon time	T_Stuck	8.0 s
Direct beacon time	T_Direct	370 ms
Scrub time	t_{Scrub}	7.1 ms

Table 2: FDDI Protocol Timing References

Network Characterization

For our purposes, the following set of two parameters characterize the basic operation of any ISO 9314 FDDI network:

- ◇ **Data Rate** - The rate of data signalling, on the ring (100 Mbps). It gives a meaning to the *bit* ($t_{bit} = 0.01\mu s$) and *octect* times ($t_{oct} = 0.08\mu s$).
- ◇ **Network Ring Latency** - This parameter accounts the time required for a data unit to propagate once around the ring. It is an aggregate variable accounting for station delays and network size, as expressed by equation:

$$t_{rlat} = t_{PD} + N_{ST} \,.\, t_{st\_lat} \tag{1}$$

  - t_{PD} – End-to-end ring cable propagation delay. It is a ring length-dependent variable with a typical value of 5 $\mu s/km$.
  - t_{st_lat} – Station latency, i.e. the delay that each individual station introduces in the repetition of data units. We consider $t_{st\_lat} = 0.6\mu s$.
  - N_{ST} – Maximum number of stations in the network. Equation (1) defines therefore a worst-case bound for the ring latency instead of their current value. However, notice that its variation with this parameter is usually small.

- ◇ **Station Delay** (t_{SD}) - This parameter represents the overhead due to transmitter idle timer after token capture. We consider $t_{SD} = 3.5\mu s$ and in order to avoid the introduction of an additional parameter we also use the station delay to represent the overhead associated with the processing of any other MAC protocol frame.

MAC Protocol Data Units

Besides token passing only two more types of protocol data frames are exchanged between peer MAC[5] entities: *claim token* frames for the negotiation of the *operational target token rotation time* and *beacon* frames for the signaling of ring breaks. The transmission of *beacon* frames can be made autonomously by the MAC protocol or under *Station Management* request. These *beacon* frames only differ in their contents. The duration of these MAC protocol frames is presented in Table 1, for an address length (l_{add}) of 48 bits. The values were computed based on frame length and data rate.

FDDI Timers

A set of timers as well as some timeout values are defined in the standard to be used by the MAC and SMT protocols. These are listed in Table 2 together with the the corresponding default values.

4 Accessibility Constrains

A comprehensive set of scenarios leading to network inaccessibility will be analysed in this section. As a general rule, we start with very simple single error situations that progressively evolve to less restrictive – and thus more realistic – operating/fault assumptions. For most of the cases, the main analysis is completely general, being particularised for best and worst cases, afterwards.

Frame Stripping

Each FDDI station is responsible for the elimination, from the network ring, of every single frame that it has transmitted. This procedure – known as frame stripping – aims to avoid the drawbacks arising from the repetition of a frame beyond the originating station: managing of frame duplicates at the MAC sub-layer, resource wastefulness, etc.

Frame stripping is initiated when a frame with a source address matching the station individual address is detected. Upon this event, the frame is deliberately truncated by replacing all the remaining data fields with *idle* symbols. This procedure actually leaves a frame remnant on the ring since a set of fields, at the beginning of the frame, were already repeated. However, this does not represent a real disadvantage: no station receives such truncated frames and all them will be eliminated whenever encountering a transmitting station.

When the emitting station fails, before stripping from the network all its transmissions, these no-owner frames will tend to persist on circulating around the ring. Since a FDDI station immediately releases the token after its last transmission, the subsequent failure of that station does not directly lead to network inaccessibility. In a best-case scenario, ring operation is not disrupted: the token will be captured by a downstream station and transmissions originated in the failed station will be eliminated when they arrive to the then-current token user. In a worst-case scenario, the token may be lost as a consequence of the failure[6] and inaccessibility will then occur.

[5] Medium Access Control.

[6] This is particularly true in small networks when the token is not captured by any station, within the round immediately following station failure.

Frame Corruption

Unlike other ring architectures [27], *all* the frame fields of particular relevance for MAC protocol robustness are covered by a *frame check sequence* (FCS). This means that a frame error can be detected with the coverage provided by the *cyclic redundancy check* (CRC) polynomial already employed in the standard ISO 8802 protocols [28]. These frames will not be received by the corresponding MAC entity. Therefore, corruption of MAC protocol data units usually degenerates in other error scenarios.

No Valid Transmissions

The FDDI LAN presents a decentralised control strategy for monitoring ring functionality. A specific timer — *Valid Transmission Timer* (TVX) — is re-started in each station, every time a non-restricted token[7] or valid frame is received. In the absence of such data units, the TVX timer is not re-started and consequently it will expire in some station. This signals a ring error situation, whose recovery will be tried through the execution of a *token recovery procedure* that includes: a *token claim process* for the election of the new token generator, with a duration given by t_{tcp}, as well as a *token restoration process*, of duration t_{trp}. Assuming that this process succeeds in providing service restoration, the corresponding inaccessibility period will present a duration given by the sum of the time required to detect the error – $t_{dect \leftarrow noVtx}$ – plus the time required for its recovery, as described by equation:

$$t_{ina \leftarrow noVtx} = t_{dect \leftarrow noVtx} + t_{tcp} + t_{trp} \tag{2}$$

The time elapsed between the occurrence of the fault and expiration of TVX represents the ring error detection latency, for this scenario. In the worst-case the fault just occurs after TVX re-start, while in the best-case the data unit triggering TVX re-start is able to complete almost an entire ring round before the occurrence of the fault. Under such conditions the error detection latency is described by equation (3). The value of TVX timer should be made large enough, in order to allow its expiration only in the presence of long-term ring errors, such as token loss. Short-term random noise bursts shall never cause TVX timeout.

$$t_{dect \leftarrow noVtx} = \begin{cases} t_{TVX} - t_{rlat} & \text{best-case scenario} \\ \\ t_{TVX} & \text{worst-case scenario} \end{cases} \tag{3}$$

Upon error detection, a recovery process is entered. Execution of the aforementioned *token claim process* is completely distributed and aims two goals: negotiate among all the network stations the next *operational token rotation time* (T_Opr) and bid for token regeneration [29]. The process is started when the station where TVX has expired, initiates the continuous transmission of *claim_token* frames, carrying its proposal for the *target token rotation time* (TTRT), as well as its individual address. Downstream stations that receive this frame can either accept or reject this proposal.

The proposed *target token rotation time* is accepted by a downstream station whenever its own *target token rotation time* is higher than the proposed one, i.e. this station requests a slower token round time. In such a case, it stores the received *target token rotation time* as the current bidding

[7]The FDDI LAN allows a restricted token utilization, previously negotiated among stations comited with asynchronous data transfers. Normal operation is performed using non-restricted tokens. Further details can be found in [29].

value for the token rotation time[8], starts to repeat the received *claim_token* frames and, eventually, stops a *token claim process* formerly initiated.

Otherwise, the proposal will be rejected. A downstream station which requests a *target token rotation time* lower than the carried by the *claim_token* frame will either continue to leader a *token claim process* formerly initiated or starts, at this time, their execution, with the transmission of its own *claim_token* frames. An eventual tie between contenting stations will be resolved by station addresses, with the station with the highest address taking precedence.

The *token claim process* successfully terminates when a given station receives its own *claim_token* frames. The exact duration of the *token claim process* depends on whether or not there is a process leadership replacement and on how quickly this leadership propagates to the winning station. In the best-case, token claiming is initiated by the winning station, i.e. by the station with the highest address among those requesting the fastest token rotation. The duration of the *token claim process* is, in this case, essentially defined by the time required to receive back from the ring their own *claim_token* frames and is given by equation:

$$t_{tcp}^{bc} = t_{SD} + t_{CLM} + t_{rlat} \tag{4}$$

A different situation occurs when token claiming is initiated by the lowest precedence station, i.e. by the station with the lowest address among those requesting the slowest token rotation. The previously described algorithm for *target token rotation time* negotiation requires the replacement of the process initiator in the token claiming leadership. Despite the simplicity of this operation, it consumes time and therefore increases the duration of the *token claim process*. So, a worst-case scenario is obtained when this process of leadership replacement successively continuous throughout all the network stations, with each station taking the place of its nearest upstream neighbor, until token claiming control reaches the highest precedence station.

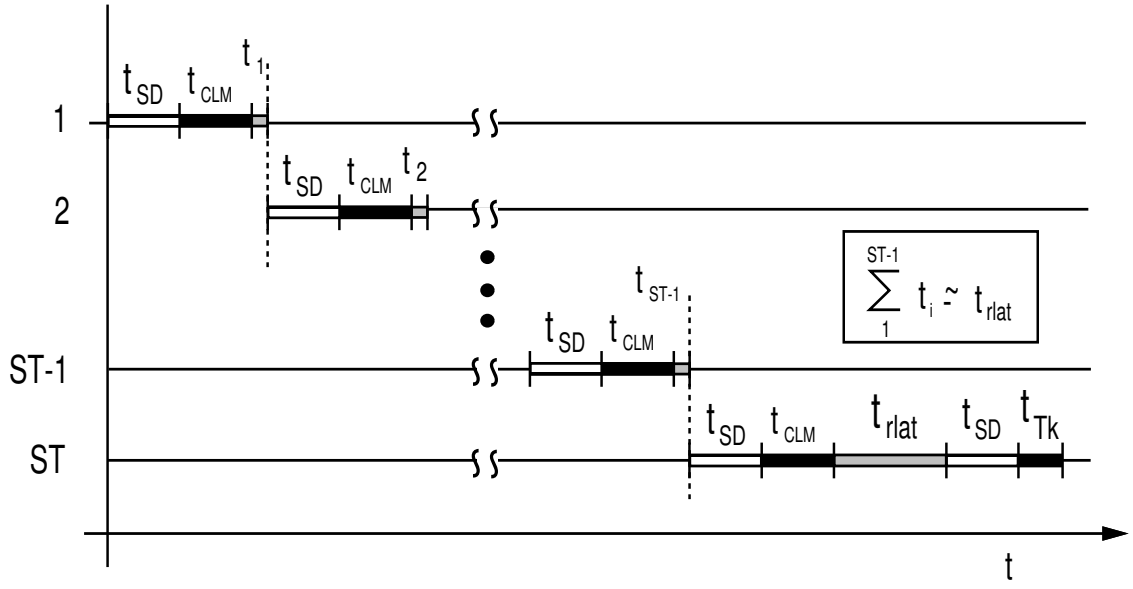

Figure 1: Contention resolution on token claim process (worst-case scenario)

A representation of this process is provided in Figure 1. Such situation only occurs if attachment of individual stations to the network ring is ordered by increased precedence and assumes that the

[8]The MAC receiver stores this value in the *T_Bid_Rc* register and passes the final negotiated value (*T_Neg*) to the MAC transmitter, where it becomes the *operational target token rotation time* (*T_Opr*) upon successful ring initialization. Further details can be found in [29].

TVX timer does not expire in any other station, but in token claiming initiator. This allows the establishment of an upper bound for *token claim process* duration, as defined by equation:

$$t_{tcp}^{wc} = N_{ST} \,.\, (t_{SD} + t_{CLM}) + 2 \,.\, t_{rlat} \tag{5}$$

Upon termination of the token claim process, the winning station issues a token, that circulating around the ring, performs a very important role in the completion of its recovery. In fact, this initial token is not captured by any station[9], since its purpose is not provide service to transmission queues but rather trigger, in each station, the update of specific MAC operational variables and align the *Token Rotation Timer* (TRT) throughout all network stations. The exact set of actions include the assertion of the *Ring_Operational* flag, the setting to one of the *Late_Ct* counter, as well as, the set of the *operational target token rotation time* (T_Opr) to the newly negotiated T_Neg value[10]. The TRT timer is then set to T_Opr. The time required to perform these actions, in all network stations, an operation that we designate by *token restoration process*, is given by equation:

$$t_{trp} = t_{SD} + t_{Tk} + t_{rlat} \tag{6}$$

Service of transmission queues is started thereafter, but since $Late\_Ct \neq 0$, this service is restricted to the synchronous access class. During the second token rotation, each station accounts the current synchronous bandwidth utilization, and resets the *Late_Ct* counter. This enables service of asynchronous latency classes[11] on the third and subsequent token rotations. Therefore, the inaccessibility period is not, for those access classes, merely represented by the time time required to recover the token. Its worst-case value, must be consolidated with the addition of the network access delay upper-bound.

As a matter of fact, such correction is required even in a more restrictive scenario, where only the existence of synchronous traffic is considered. This arises from some side effects that token recovery may have on the normal token circulation. To illustrate this, consider the FDDI network, of Fig. 2. Let us assume that station $\mathcal{S}_1$ has the highest precedence and that the token is garbled, due to random noise, at the output of station $\mathcal{S}_{n-1}$. Ring recovery is initiated upon TVX timeout in some station and the subsequent *token claim process* is wined by the highest precedence $\mathcal{S}_1$ station. Station $\mathcal{S}_n$, which was about to receive the token when the fault occurs, is passed over. So, in addition to the token recovery time, station $\mathcal{S}_n$ must wait an extra token rotation to get access to the network and, therefore, it sees an inaccessibility time added by that delay.

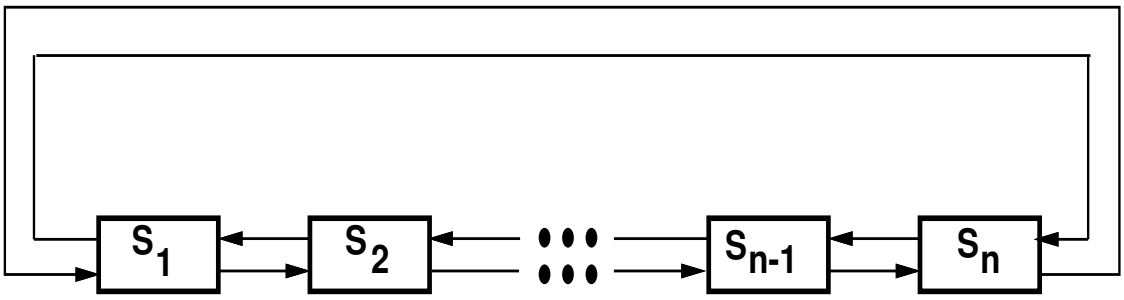

Figure 2: Example of a FDDI network

[9] The absence of token capture is due to the non-assertion of *Ring_Operational* flag.
[10] A detailed description of these protocol elements can be found in [29] or, for some of them, later in the paper.
[11] A detailed description of the FDDI latency classes can be found in [29].

No Valid Tokens

The mechanism described in the previous scenario has been designed to cope with an abnormal absence of valid frames or with the erroneous persistence of a *restricted* token in the ring [26]. Although that mechanism may represent, for most of the cases, an efficient speed-up alternative to the method herein described for the detecting of incorrect ring activity, it does provide full coverage.

Let us analyse a particular example: a station fails, after having transmitted a frame but before token release. Assuming that the transmitted frame is not garbled, due to an eventual ring interruption, it will persist on the ring. Incorrect ring operation is not detected by TVX timeout, since the persistent frame will cyclely reset that timer.

A decentralised mechanism is provided by the FDDI protocol to cope with such problems. It is based on the timeliness characteristics of token circulation. The affected resources, that remain active in each network station, are the following:

- **Token Rotation Timer** (TRT) – under normal ring operation it is re-started with the current value of the *operational target token rotation time* (T_Opr), each time a valid token (restricted or non-restricted) is received with $Late\_Ct = 0$ or when it expires.
- **Late Counter** (Late_Ct) – under normal ring operation this counter holds the number of TRT timeouts since the last received token (restricted or non-restricted). This counter is reset on the arrival of a valid token (restricted or non-restricted) and is incremented on each expiration of TRT. It is set to one upon station initialize or ring recovery.

These MAC elements, together with the *Token Holding Timer*[12], support, within each station, the execution of the timed-token access protocol, scheduling ring transmissions accordingly with the following rules:

- If the token is on time, i.e. if $Late\_Ct = 0$, then THT is set to the current value of TRT, before its re-start with the value stored in T_Opr. Both synchronous and asynchronous frames may be transmitted.
- If the token is late, i.e. if $Late\_Ct \neq 0$, then $Late\_Ct$ is reset but not TRT, in order to retain the accumulated lateness. Only synchronous frames may be transmitted, in this case.
- The time spent in the transmission of synchronous frames may not exceed the station's synchronous bandwidth allocation[13] The time taken by transmission of asynchronous frames is upper-bounded by THT. Service of both classes is prevented after TRT expiration.

This methodology of accessing network medium presents timeliness characteristics guaranting that, under normal operation, a token always return to any given station in a time upper-bounded by $2 \times T\_Opr$. This claim has been formally proofed in [3] and furnishes the foundations for a method aiming the detection of incorrect ring activity:

> Ring recovery is initiated upon TRT has expired twice without any token has been received, i.e. recovery is started when TRT expires and $Late\_Ct > 0$.

Establishment of an expression for inaccessibility duration is straightforward. The ring error detection latency is essentially defined by the value of the TRT timer, as expressed by equation (7). The value associated with TRT corresponds to the previously negotiated *target token rotation time*, which is lower and upper bounded by technological and functional constrains associated either with interoperability issues or with network size [29].

[12]The *Token Holding Timer* (THT) controls how long a station may transmit asynchronous frames. A station may start an asynchronous transmission as long as THT has not expired and it is lower than the associated priority threshold value. Further details can be found in [29].

[13]Not checked by MAC. Supported by a dedicated protocol, inside SMT [30].

$$t_{dect \leftarrow noTk} = \begin{cases} 2 \,.\, t_{TRT} - t_{rlat} & \text{best-case scenario} \\ \\ 2 \,.\, t_{TRT} & \text{worst-case scenario} \end{cases} \tag{7}$$

Ring recovery is based on the aforementioned *token recovery procedure* and therefore, the corresponding inaccessibility period is given by equation:

$$t_{ina \leftarrow noTk} = t_{dect \leftarrow noTk} + t_{tcp} + t_{trp} \tag{8}$$

Ring Interruption

The previously described mechanisms are able to recover ring operation, whenever its faulty behaviour is not due to a physical cause, like cable break or defective transmitter/receiver. However they are helpless when dealing with such problems.

The impact of physical ring interruption on network operation depends on its nature (physical or logical interruption) and duration. Ring interruption errors are essentially recovered by *Station Management* (SMT) protocols, with the eventual cooperation of dedicated MAC sub-layer processes and station self-test capabilities.

Special mechanisms are provided, within SMT, for detecting defective link connections and breaks in the logical ring. The methodology dealing with the first problem is based on the continuous monitoring of inbound line states, whereas the second one requires the intervention of MAC sub-layer. This latter process runs several orders of magnitude slower than line state monitoring [31].

In the next scenarios we perform an analysis of these error detection methods. Further details on the protocols where they are built-in, can be found in the standard documents [30, 29]. Since the corresponding recovery actions are not fully covered by the standard specification we made the reasonable assumption of considering algorithms that can be generically implemented, without violating FDDI constrains and inter-operability issues. Evolution of the standard specification may, nevertheless, originate the revision of this model in order to ensure a more accurate description of network inaccessibility behaviour.

Line State Monitoring

A dedicated protocol inside SMT is responsible for establishing and managing the connection between neighboring ports. This protocol – known as *Physical Connection Management* (PCM) – initializes such a connection and manages the required signalling.

Due to their inherent functions, the PCM is particularly suitable to support the detection of abnormal operating conditions, originated by the failure of PHY[14] and/or PMD[15] layer components. Under correct operation, a stream of *idle* symbols should be received within a period defined by T_Out, from each inbound active link. Should this not occur or otherwise, should *quiet* symbols, *halt* symbols or an extended noise condition be received, and the PCM will attempt re-initialize port connection, since these can be either symptoms of a defective/broken connection or an indication that the other end has just started a connection procedure, probably due to the joining of a new station.

However, in the presence of a failure such an attempt to restore port connection does not succeed and the PCM will either stuck in its entry-point or will cyclely perform successive attempts to re-establish port connection. A high-level SMT entity[16] can monitor these situations, based on

[14]The *Physical* layer of FDDI [32].
[15]The *Physical Medium Dependent* layer of FDDI [33].
[16]Currently not specified.

indications furnished through the CMT-SMT interface[17] and take a decision on subsequent actions, eventually after having performed some sort of self-testing.

Together with the PCM, another *Connection Management* entity, the *Link Error Monitor* (LEM), performs a role of outstanding importance in the observation of link quality. In fact, the LEM examines the *link error rate* (LER) of an active link, aiming the detection of an inadequate *binary error rate* (BER) due to a marginal link quality, link degradation or connector unplugging. These tests complement any off-line *link confidence test* [30] and their results can be used by the aforementioned SMT entity, namely for fault diagnostics and fault isolation purposes.

Physical Ring Recovery

Upon detection of a failure, the network should be reconfigured in order to remove the failed component, from the data path. The reconfiguration of each local station is executed by a CMT entity, the *Configuration Management* (CFM), which is responsible for the interconnection of PHY and MAC components.

Any legal station configuration can be selected. This selection is triggered by relevant changes in the PCM status and is performed accordingly with well defined criteria, taking as a base the description of component's state, held the *Management Information Base* (MIB) [30]. The present standard specification does not clearly define the way how these MIB objects should be managed, i.e. does not define the criteria that rule component's state update, from the indications furnished upon PCM status change. This means that a fault detection and fault isolation policy should be formally defined, hopefully, in the near future. Meanwhile, and in order to complete our study of networking inaccessibility, we postulate a set of reasonable assumptions, to which any fault detection and isolation policy should yield.

- Networking components failures can be reliably detected through SMT protocols, eventually assisted by self-test procedures.
- Fault detection procedures are time bounded.
- Error recovery, to be performed either by station removal or through network physical reconfiguration, is also time bounded.

In order to define an expression for the inaccessibility period due to the previously described reconfiguration procedure, let us define the following parameters:

- $t_{PCM-Dect}$ – The PCM error detection latency.
- $t_{st \leftarrow join}$ – The interruption on ring data flow, due to the joining of a given station.
- $t_{st \leftarrow leave}$ – The duration of ring interruption, due to the withdraw of a station.
- $t_{self-test}$ – The period required for a station reliably test their network components.
- t_{scrub} – The time required for the execution of the *scrub function*. The *scrub* function is an error prevention measure that aims to purge from the ring all the data units, issued before reconfiguration takes place.

The physical interruption of the network ring is therefore detected by *line state monitoring* and recovered through network reconfiguration. The duration of the ring interruption is given by equation (9), where we have specifically considered the time taken by station withdraw, for self-testing, and by the correct station join or, alternatively, the establishment of a connection between their upstream and downstream neighbors, if both stations have failed.

[17] The CMT - *Connection Management* - is a group of *Station Management* protocols that aims to assure port insertion/removal and the connection of port PHY entities to MAC entities. Besides PCM, it includes the *Entity Coordination Management* (ECM) and the *Configuration Management* (CFM).

$$t_{PhyBrk} = t_{PCM-Dect} + t_{st\leftarrow leave} + t_{self-test} + t_{st\leftarrow join} + t_{scrub} \quad (9)$$

Additionally, notice that interruption of physical ring directly leads to the collapse of logical ring, whose systematic analysis is performed in the next scenario. The inaccessibility period due to the need of network physical reconfiguration can then be easily obtained from the expressions of next scenario, by considering that physical and logical ring interruption present the same duration.

Logical Ring Recovery

A set of faults may lead to data flow interruption, although the physical ring remains intact. Let us to group such errors under the generic designation of *logical ring interruption*, whether or not they may have a physical cause. The two methodologies earlier described for restoration of token circulation can be view as particular cases of a logical ring collapse, where the interruption is essentially due to transient operational conditions rather than to severe failures.

In those scenarios, error recovery is started immediately after its detection, but that will not be the case when the error condition persists beyond their detection. Let us denote by $t_{RingBrk}$ the duration of ring interruption. The possible error scenarios are as follows:

i) $t_{Dect} \leq t_{RingBrk} < t_{MacBeacon}$ — with $t_{MacBeacon} \simeq t_{Dect} + T\_max$, where t_{Dect} represents the ring error detection latency bounds, as given by equations (3) and (7), and T_max is the maximum value for the *target token rotation time* [29]. This last parameter is several times the maximum ring initialization time, in order to allow a stable ring recovery, and places a deadline for termination of token claiming recovery actions. Should the network ring data path be restored in a time that allows the successful execution of token claiming and its operation is able to be recovered strictly through the set of actions associated with the aforementioned *token recovery procedure*. The corresponding inaccessibility period is directly proportional to the logical ring interruption duration, as given by equation:

$$t_{ina\leftarrow noRing}(noBeacon) = t_{RingBrk} + t_{tcp} + t_{trp} \quad (10)$$

ii) $t_{MacBeacon} \leq t_{RingBrk} < t_{DirBeacon}$ — where $t_{DirBeacon}$ is defined ahead. Termination of the *token claim process* previously used to recover the logical ring is monitored through the TRT timer. Upon that process initiation, the TRT timer is always loaded with the T_max value. Should the error condition persist beyond this timeout value and TRT timer will expire. This usually means that a more severe fault has occurred on the ring and that a stronger method for its recovery is required. Such recovery procedure, known as *MAC beacon process*, is started with the continuous transmission of *MAC beacon* frames and ends when a station receives back their own *MAC beacon* transmissions. The *MAC beacon process* aims to provide a simple logical ring recovery method for medium-term breaks [34], like for example the withdrawal of active stations from the network ring. Provided that the ring break duration lies within the aforementioned limits, the *MAC beacon process* successfully terminates and the corresponding inaccessibility time is given by equation:

$$t_{ina\leftarrow noRing}(MacBeacon) = t_{RingBrk} + t_{brp} \quad (11)$$

where t_{brp} – *beacon recovery process* – represents the time required to complete the restoration of network service. It includes the propagation of a *beacon* frame around the ring, the bidding of a new *operational target token rotation time* and the corresponding token issuing.

$$t_{brp} = t_{BCN} + t_{rlat} + t_{tcp} + t_{trp} \quad (12)$$

iii) $t_{DirBeacon} \leq t_{RingBrk}$ — A dedicated protocol within SMT monitors ring activity. This protocol, known as *Ring Management* (RMT), tracks MAC activity in order to detect when the ring is

not operational (claiming or MAC beaconing). Among other functions [30], RMT aims to detect "stuck beacon" conditions, i.e. situations of logical ring interruption that are unable to be recovered by the *MAC beacon process*, presumably because a reconfiguration that cannot be performed by the methods early described is required. Under such scenario we may find the failure of components located above the PHY level. Execution of *MAC beacon process* is allowed during a time given by:

$$t_{DirBeacon} = T\_Non\_Op + T\_Stuck \tag{13}$$

where T_Non_Op is the time allowed for ring recovery, before examining any other error condition and T_Stuck is the time granted, after that, for *MAC Beacon* transmissions [30]. Execution of a *direct beacon process* is entered after this period. This process is used to notify the ring that a "stuck beaconing" condition have been detected through the transmission of *direct beacon* frames, usually addressed to that station upstream neighbor. These frames are transmitted only for a short period of time, before the *PC-Trace* function is invoked. The *PC-Trace* function uses the PHY level signalling to identify the fault domain, which is defined between the MAC of the beaconing station and its nearest upstream MAC. Execution of the *PC-Trace* function requires the cooperation of the PCMs and ECMs within the fault domain. The recovery actions are actually only triggered upon execution of the *PC-Trace* function. After the *trace request* has been propagated to nearest upstream station with an inserted MAC, this function will be acknowledged downstream to initiating station. Testing of *data paths* [30] will be performed, at this point, by stations located at the edges of the fault domain. Failed components and/or stations should then be removed in order to restore network operation.

Let us to denote by $t_{PC-Trace}$ the time required for the execution of the *PC-Trace* function and by t_{path_test} the time needed to reliable test the data path local to a given station, as suggested in [30]. The duration of the inaccessibility period for this scenario is given by:

$$\begin{aligned} t_{ina \leftarrow noRing}(DirBeacon) \quad = \quad & t_{DirBeacon} + T\_Direct + \\ & t_{PC-Trace} + t_{st \leftarrow leave} + t_{path\_test} + \\ & t_{st \leftarrow join} + t_{scrub} + t_{brp} \end{aligned} \tag{14}$$

where T_Direct is the period of time during which *direct beacon* frames are transmitted, before the *PC-Trace* function is invoked. Equation (14) accounts the time required for station withdrawal before its testing, the time needed for the join of the station that remains correct or, alternatively, for the establishment of a connection between their upstream and downstream neighbors, if both stations have failed. Naturally it also accounts the time required to purge the ring of all the data units issued before reconfiguration.

Dumb Transmitter

Having generically studied FDDI inaccessibility, let us now consider some particular failure scenarios. We start with a scenario where a station is not able to transmit a signal to the medium. This may include failures in the transmit section of the PHY layer, as well as, the failure of optical transmitter, at the PMD sub-layer.

This station downstream neighbor detects this situation upon reception of *quiet* symbols originated by the absence of medium signalling. This starts the aforementioned PCM actions for fault detection on both sides of the connection, which allows the station with the dumb transmitter to detect its fault. Let us assume that detection of the abnormal signalling condition is performed by the PCM in a time given by $t_{PCM-Dect}(noTx)$. Known a numeric value to this parameter, the corresponding inaccessibility period can be easily evaluated.

Deaf Receiver

Let us assume that the optical receiver or the associated section of the PHY layer fail by starting to report only *quiet* symbols. This situation is essentially equivalent to the previous scenario, so

we made $t_{PCM-Dect}(noRx) = t_{PCM-Dect}(noTx)$.

Broken Cable

The symptoms of this failure are equivalent to those reported in the previous scenarios. Recovery from a broken cable is achieved by wrapping the data path into the secondary ring. Therefore, we assume that fault detection is more complex and thus $t_{PCM-Dect}(noLink) > t_{PCM-Dect}(noTx)$.

Jabbering Transmitter

Let us assume that the transmitter of a given station fails by stucking in the send of a continuous stream of data. Clearly, the *token claim process* does not succeed, since that station is not able to repeat *claim token* frames. The problem is detect by PCM timeout, after a period of T_out without any *idle* symbol. The diagnose and recovery procedures are similar to those of previous scenarios. The PCM latency must now obey to relation $t_{PCM-Dect}(JabTx) > T\_Out$.

Streaming Receiver

In this scenario we consider that the receiver section fails in the decoding of network control signal, thus feeding a continuous stream of data to both PCM and MAC inputs. The scenario is quite similar to the jabbering transmitter and therefore, we consider $t_{PCM-Dect}(StrRx) = t_{PCM-Dect}(JabTx)$.

Streaming MAC Receiver

In this scenario, we consider that the receiver section of a given station, although feeding a correct signal to PCM inputs, fails provision of correctly decoded data to the MAC interface. In this case, the PCM will not detect any abnormal condition and therefore does not initiates any fault isolation procedure.

From MAC operation viewpoint, the absence of correct data reception leads the station to a "stuck beacon" condition. The RMT will then start *direct beacon* and *PC-Trace* executions. The corresponding inaccessibility duration is given by equation (14).

Station Join

Stations are physically inserted in the network ring through the switching of a dedicated optical *bypass* switch. The PCM of the just arrived station, as well as, of their neighboring ports will then initiate the connection procedure. We consider that execution of a station join takes a time given by $t_{st \leftarrow join}$. The resulting inaccessibility period is obtained by considering a ring interruption with that duration.

Multiple Joins

In the best-case all the stations will simultaneously join to the network ring and the situation will not be distinguished from the insertion of a single station. In the worst-case, station joins are serialised. An upper-bound for ring interruption duration can be obtained assuming a massive join to a network previously formed by only two stations.

$$t_{RingBrk \leftarrow mjoin} = (N_{ST} - 2) \,.\, t_{st \leftarrow join} \tag{15}$$

Station Leave

A station leaves the ring by switching their optical relay to the *bypass* state. Afterwards, the new adjacent ports located upstream and downstream of the missing station must complete a connection procedure. The corresponding inaccessibility period is obtained by considering a ring interruption duration given by $t_{st \leftarrow leave}$.

Multiple Leaves

In the best-case, all the stations simultaneously abandon the network ring and the situation cannot be distinguished from the withdraw of a single station. In the worst-case, these events are serialised and upon a massive withdraw of $(N_{ST} - 2)$ stations, the duration of ring interruption is given by:

$$t_{RingBrk \leftarrow mleave} = (N_{ST} - 2) \, . \, t_{st \leftarrow leave} \tag{16}$$

Analytic Results

In order to complete our study of networking accessibility we now evaluate the inaccessibility time bounds, for a FDDI installation, for example, an industrial environment with a small cell network for real-time manufacturing control. The network length $C_l = 500m$ and the total number of stations $N_{ST} = 32$. Estimates for other network parameters are defined: $t_{st \leftarrow join} = 30ms$, $t_{st \leftarrow leave} = 20ms$. The implementation dependent parameters associated with self-test procedures are also estimated: $t_{self\_test} = t_{path\_test} = 5ms$. Finally we also estimate a set of parameters associated with PCM behaviour under different operation conditions: the parameters associated with the detection of faults triggered by reception of *quiet* symbols $t_{PCM\_Dect}(noTx) = t_{PCM\_Dect}(noRx) = 15ms$, $t_{PCM\_Dect}(noLink) = 25ms$; and the parameters associated with the detection of faults by timeout in the reception of *idle* symbols $t_{PCM\_Dect}(JabTx) = t_{PCM\_Dect}(StrRx) = 115ms$. The results of the evaluation, for each one of the studied scenarios, are summarized in Table 3.

Data Rate - $100Mbps$		
Scenario	t_{ina} (ms)	
	min.	*max.*
No Valid Transmissions	2.53	2.76
No Valid Tokens	15.03	15.26
Dumb Transmitter	77.15	77.36
Deaf Receiver	77.15	77.36
Broken Cable	87.15	87.36
Jabbering Transmitter	177.18	177.39
Streaming Receiver	177.18	177.39
Streaming MAC Receiver	9457.18	9457.33
Station Join	30.03	30.24
Multiple Joins ($N_{ST} = 32$)	30.03	900.29
Station Leave	20.03	20.24
Multiple Leaves ($N_{ST} = 32$)	20.03	600.29

Table 3: FDDI Inaccessibility Times ($l_{add} = 48$)

The worst case figures are rather high, like for example the $9.5s$ required for the removal of a streaming MAC. However, note three points: the parameter estimates are quite conservative, the network parameters are based on the default values and in consequence inadequate for a network with at most 32 stations and finally the probability of occurrence of some worst-case scenarios is very small. As a minimizing strategy we envisage the tuning of timing references with network size. This can bring some of the figures drastically down.

The study is interesting on the grounds that it provides a basis to know what to expect from FDDI performance in the presence of failures, and provides guidance on how to improve the situation and *justifiably* achieve better performability , and thus better respect any bounded delay requirements.

5 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. While LAN replication is a solution, it is costly and complex. Some applications can live with temporary glitches in LAN operation, so an alternative approach is to quantify all these glitches or temporary partitions, which we have named *inaccessibility* periods, and derive a worst-case figure, to be added to the worst-case transmission delay expected in the absence of faults.

In these conditions, reliable real-time operation is possible on non-replicated LANs, through appropriate techniques[12].

This paper does an exploratory study of the inaccessibility characteristics of the ISO 9314 FDDI LAN, addressing the problem of temporary LAN partitioning in a comprehensive way. The derived error-handling performance model allows the evaluation of corrective terms for computing transmission delays in various situations.

References

[1] Jeffery H. Peden and Alfred C. Weaver. The utilization of priorities on token ring networks. In *Proceedings of the 13th Conference on Local Computer Networks*, Minneapolis, USA, October 1988.

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

[3] Marjory J. Johnson. Proof that timing requirements of the FDDI token ring protocol are satisfied. *IEEE Transactions on Communications*, 35(6), June 1987.

[4] 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.

[5] Gerard LeLann. Critical issues in distributed real-time computing. In *Proceedings of the Workshop on communication networks and distributed operating systems within the space environment*. European Space Research and Technology Centre, October 1989.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] C. Guerin, H. Raison, and P. Martin. Procedé de diffusion sûre de messages dans un anneau et dispositif permettant la mise en oeuvre du procedé. *French Patent*, (85.002.02), January 1985.

[12] 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.

[13] 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.

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

[15] X3T9.5 FDDI. *FDDI documents: Media Access Layer, Physical and Medium Dependent Layer, Station Mgt.*, 1986.

[16] Floyd Ross. An Overview of FDDI: The Fiber Distributed Data Interface. *IEEE J. on Selected Areas in Comm.*, 7(7), 1989.

[17] FDDI. *Single-Mode Physical Layer Medium Dependent (SMF-PMD).* ANSI X3T9.5/ 88-155 Draft Proposal Rev 3, 1989.

[18] Lawrence J. Lang and James Watson. Connecting remote FDDI installations with single-mode fiber, dedicated lines, or SMDs. *Computer Communication Review (ACM Sigcomm)*, 20(3):83–94, July 1990.

[19] FDDI. *Shielded Twisted Pair Physical Medium Dependent (STP-PMD).* ANSI X3T9.5/ 91-166 IBM Initial Draft Proposal, 1991.

[20] FDDI. *Unshielded Twisted Pair Physical Medium Dependent (UTP-PMD).* ANSI X3T9.5/ 91-170 Draf Proposal Rev 0, 1991.

[21] A. Schill and M. Zieher. Performance analysis of the fddi 100Mbit/s optical token ring. In *Proceedings of the IFIP TC6/WG 6.4 Workshop on HSLAN*, pages 53–74, Aachen, February 1987. IFIP.

[22] K. C. Sevcik and M. J. Johnson. Cycle time properties of the FDDI token ring protocol. *IEEE Transactions on Software Engineering*, 13(3), March 1987.

[23] D. Dykeman and W. Bux. Analysis and tuning of the fddi media access control protocol. *IEEE Journal on Selected Areas in Communications*, 6(6):997–1010, July 1988.

[24] Marjory J. Johnson. Analysis of FDDI synchronous traffic delays. In *Proceedings of Systems Design and Networks Conference: Patting Local Area networks t Work*. IEEE, January 1988.

[25] Marjory J. Johnson. Performance analysis of FDDI. In *Proceedings of EFOC/LAN'88 Conference*, Amsterdam, April 1988.

[26] Marjory Johnson. Reliability mechanisms of the FDDI high bandwidth Token-ring protocol. *Computer Network and ISDN Systems*, 11(2), 1986.

[27] *ISO DP 8802/5-85, Token Ring Access Method*, 1985.

[28] R. Jain. Error characteristics of Fiber Distributed Data Interface. *IEEE Transactions on Communications*, 38(8):1244–1252, August 1990.

[29] FDDI. *FDDI Token-Ring Media Access Control (MAC).* ANSI X3.139, 1987.

[30] FDDI. *Station Management (SMT).* Draft Proposal ANSI X3T9.5/91 SMT-LBC-161, January 1992.

[31] J. Hamstra. *FDDI Ring Reconfiguration.* SMT 40 - X3T9.5 working document, January 1987.

[32] FDDI. *Physical Layer Protocol (PHY).* ANSI X3.148, 1988.

[33] FDDI. *Physical Layer Medium Dependent (PMD).* ANSI X3.166, 1990.

[34] My T. Le. *FDDI Fault Detection and Recovery Mechanisms.* SMT 287 - X3T9.5 working document, February 1989.