A METHOD FOR ANALYSING THE PERFORMANCE ASPECTS OF THE FAULT-TOLERANCE MECHANISMS IN FDDI

Aad P.A. van Moorsel Boudewijn R. Haverkort Ignas G. Niemegeers

University of Twente, Tele-Informatics and Open Systems group
P.O. Box 217, 7500 AE Enschede, The Netherlands
tel: +31 53 893767; fax: +31 53 333815; e-mail: moorsel@cs.utwente.nl

Abstract

The high speed Fibre Distributed Data Interface is becoming an accepted standard for many real-time communication networks. One of its real-time applications supporting aspects is that the fault tolerance mechanisms are fast and initiated automatically. Nevertheless, occurring errors can have great influence on performance measures such as the probability of missing a deadline. In this paper we analyse the ability of the error recovery mechanisms to make FDDI satisfy real-time performance constraints in the presence of errors. A complicating factor in these analyses is the rarity of the error occurrences, which makes direct simulation unattractive. Therefore, we have developed a fast simulation technique, called Injection Simulation, which makes it possible to analyse the performance of FDDI including its fault tolerance behaviour. In this paper we discuss the implementation of Injection Simulation for polling models of FDDI and show simulation results.

KEYWORDS: FDDI, Fault Tolerance, Performance Evaluation, Fast Simulation, Systems with Rare Events, Polling Models.

1 Introduction

The FDDI high speed token ring protocol [1] is a likely candidate for many local area communication networks which should support real-time applications. In this respect one can think of safety critical applications such as avionics or nuclear power plant control and of less critical applications like video transmission or factory automation. FDDI supports real-time traffic by different means. First of all it can offer high bandwidth, namely 100 Mbps, as it uses optical fibre as the medium. Secondly, when the ring operates error free, the timed token protocol guarantees a maximum access delay for the stations on the ring, thus enabling them to send some highest priority, or synchronous, traffic [2]. Finally, the fault tolerance mechanisms [3] contribute to the real-time capabilities of FDDI, because they are extremely fast and automatically performed.

In this paper we will discuss the influence of error occurrences and the fault tolerance mechanisms on the performance of the network. A first indication for this influence on the Quality of Service is the fact that a maximum access delay can no longer be guaranteed. For many applications though, this will not be the performance measure that is of most interest. Other measures like buffer overflow probability or the probability of missing a deadline, can be of interest to the user or network supplier. These probabilities will probably be very small in an error free operating FDDI ring, but might increase some orders of magnitude when error occurrences are taken into account. In this paper we will analyze the influence of the fault tolerance mechanisms on the probability that jobs miss their deadlines.

Considering deadline missing probabilities means that we want to derive results for the tail of the steady-state waiting time distribution. Even when we abstract from most of the detail in the protocol hardly any analytical results are available. Therefore we have to use simulation to obtain results for the measures of interest. A problem we face when we try to analyze the fault tolerance mechanisms of FDDI is that errors occur very infrequently. For this reason, direct simulation will take too much simulation time until enough information about the influence of the rarely occurring errors has been collected. The use of a fast simulation technique becomes therefore inevitable. We have developed Injection Simulation (IS), an intuitively attractive fast simulation technique that can be implemented very easily [4]. The method is based on decomposition of the original system in an error affected part and an error free part. For these two parts separate simulations are performed and the output of these simulations enables us to derive the required measure for the total system.

In this paper we first of all want to show *how* performance results can be derived despite of the inherent difficulty of analyzing systems with rarely occurring events. Secondly, we want to obtain insight in the generic class of systems that can be represented by polling models with server breakdown. So, it is not our main intention to obtain detailed conclusions regarding the suitability of some specific FDDI configuration for real-time applications. The presented models will therefore not have too much detail of the FDDI protocol included. As we are doing a simulation study it is in principal possible to add any desired detail to the model and thus to analyze any kind of practical FDDI application using the Injection Simulation technique.

This paper is organized as follows. In Section 2 we first describe the most important aspects of the FDDI fault tolerance mechanisms and discuss their incorporation in a polling model. In Section 3 we describe Injection Simulation for a ratio estimator and derive its variance reduction properties. In Section 4 we then demonstrate the use of Injection Simulation for the polling model of FDDI, by discussing simulation results in detail. Finally, Section 5 states conclusions and further research topics.

2 FDDI Fault Tolerance Mechanisms and their inclusion in Polling Models

In this section we discuss FDDI's fault tolerance mechanisms and the way to incorporate them in a polling model. To be able to discuss this properly, we first very briefly describe the basic timed token protocol as well as a basic polling model in Section 2.1. In Section 2.2 we then discuss fault tolerance mechanisms in FDDI and their modelling in a polling model with server breakdown. We also briefly report on relevant analytical methods to evaluate performance of polling models.

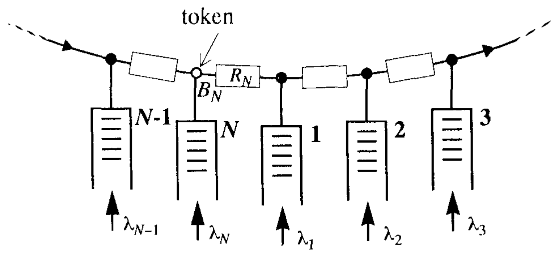
2.1 Basic concepts

Basic concepts of FDDI

The FDDI protocol is a timed token protocol for a high speed ring which guarantees a certain bandwidth for all stations (e.g. [5]). Two classes of traffic can be distinguished, synchronous and asynchronous traffic. The synchronous traffic is the highest priority traffic and is therefore especially suited for real-time applications. When a station receives the token it has the opportunity to transmit synchronous frames for some period. Only when there is time left the station is allowed to transmit asynchronous frames, which are ordered according to priority classes. In this way a maximum access delay is guaranteed, and thus the opportunity to transmit at least some synchronous traffic within two times the pre-negotiated Target Token Rotation Time (TTRT) [2].

Basic polling model

In our performance modelling study we do not have the intention to model the complete timed token protocol and we therefore will not discuss it in more detail. We will consider a simple polling model with only synchronous traffic, and with a simplified scheduling discipline, namely 1-limited service. The reason for only considering synchronous traffic is that it is likely that the highest priority traffic class will be used for real-time applications over the ring. The 1-limited scheduling discipline models the limited time a station may hold the token. 1-limited implies that at most one frame per station can be transmitted every time the token visits the station. The polling model we analyze for the system without error occurrences can thus be depicted as in Figure 1. We consider a polling model with N stations, which are polled by the server (token) in cyclic order. The switch over time of the token for station i is distributed as a random variable R_i. Later we will call the switching over of the token from station i to $i+1$ (if $i = N$ then from N to 1) a *token pass*. We assume that jobs (frames) arrive in station i according to a Poisson process with parameter λ_i and have a service time (transmission time) distributed as a random variable B_i.



N: number of nodes, R_i: switch over time,
B_i: transmission time, i: number of the node.
λ_i: arrival rate frames (Poisson),

Figure 1 Polling model of error free FDDI

2.2 Fault tolerance aspects

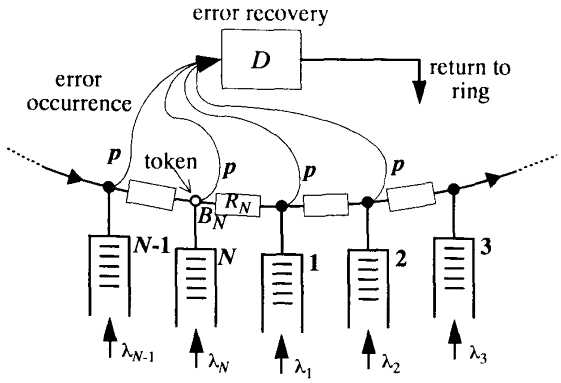
Fault tolerance mechanisms in FDDI

We can distinguish between two types of errors in a LAN: protocol related errors and errors caused by failure of a physical resource [3]. The protocol related errors lead to a recovery process, physical resource related errors lead to reconfiguration. In this paper we limit our attention to performance analysis of the recovery process. Application of Injection Simulation for a model with reconfiguration can be done in a similar, but slightly more intricate way.

Protocol related errors are detected by a time out of the Token Rotation Timer (TRT) or the Valid Transmission Timer (TVX) in some station. Upon such a time out a recovery process is automatically initiated. The station in which the time out first showed up, starts to transmit Claim Token Frames and offers some value for the Target Token Rotation Timer (TTRT). Every other station which receives a Claim Token Frame compares its own TTRT offer with the received one and transmits the lowest of the two to the downstream station. The station which receives its own Claim Token Frame knows that it has won the bidding and releases a new token. The first token rotation then is used to set the station parameters, such as timer values, the second rotation is only for synchronous traffic transmission. Then normal operation is resumed.

Polling model with server breakdown

From a performance modelling point of view, the main issue is that the recovery procedure does not allow data frames to be send for some period of time. After this period of server breakdown the ring resumes its operation as before, first



p: error probability,
D: error recovery time,
N: number of nodes,
λ_i: arrival rate frames (Poisson),
B_i: transmission time,
R_i: switch over time,
i: number of the node.

Figure 2 Polling model of FDDI with error recovery mechanism

having to deal with the backlog of jobs. Figure 2 gives the polling model. When the token goes from station i to $i + 1$ (if $i = N$, then from N to 1), an error occurs with probability p. Upon error occurrence the bidding and reinitialization of the ring is modelled by a delay with random length D. Typically p is very small. This error occurrence directly models token losses. Another aspect of modelling error recovery is the return strategy of the server (i.e. token) to the ring. After the Claim Token process the station that wins the bidding issues a new token and restarts normal ring operation. Depending on the application one can model the return of the token by different strategies. Appropriate seem in many cases a so-called Fixed Station strategy in which always the same station wins the bidding, or a Random Station strategy in which stations win the bidding at random.

Analytical performance results for polling models

We state some general analytical results for cyclic server or polling models which are of interest to our study. For a detailed and up-to-date survey of results for these models, see Takagi [6].

Our performance measure of interest is the probability that a job misses some predefined deadline. This implies that we have to look at the tail of the steady-state waiting time distribution of jobs in the system. For this measure hardly any analytical results are available. Known analytical results mainly concern mean values of the waiting time distribution. Results for the tail of the waiting time distribution have only been obtained for models that further abstract from the timed token protocol and are often Laplace Stieltjes Transforms that are difficult to invert. An exception is Genter [7], who analyses a polling model with a gated service discipline. Assuming that the cycle time is independent of the number of jobs in the stations, results about the tail of the waiting time distribution can be derived relatively easily.

Adding the recovery mechanism in the model complicates the derivation of analytical results, as the order in which the stations are visited by the server (i.e. the token) is no longer fixed. Only for a return strategy in which the token returns to the first downstream station the results for the original model can directly be extended [8]. For our measure of interest approximations like the independence assumption in [7] do not seem to be appropriate because the influence of the occurring errors is also visible in the prolonged cycle time. To get a good insight in the influence of the errors we therefore have to use simulation. The problem of simulating models with rare events, such as small server breakdown probabilities, will be discussed in the following section.

3 Injection Simulation

When we try to investigate for FDDI the influence of error occurrences on the waiting time distribution we face the problem of the rarity of these errors. To derive accurate simulation results in a reasonable time span for these systems we have to use a fast simulation technique (see e.g. [9], [10], [11] for discussions on simulation of systems with rarely occurring events). In Section 3.1 we first briefly discuss the Injection Simulation (IS) method as presented in [4]. Then we extend it in Section 3.2 for the ratio estimator which comes up when we try to estimate the probability of missing a deadline for an FDDI ring with token losses. In Section 3.2 also results about unbiasedness and variance reduction for the ratio estimator are given.

3.1 Injection Simulation

The basic idea behind Injection Simulation[1] is that an occurred error only influences the performance for a limited period of time, i.e. the influence of the error on the performance *fades out.* After the influence has faded out the performance is considered to be identical to the performance of the error free system, until the next error occurs. This idea is justified by the very small error probabilities p we consider. The waiting times of the jobs which are considered to be affected by the occurred error are called *affected* observations, while the other observations that are considered to behave as in an error free system are called *non-affected* observations (see upper part Figure 3). Performance results for the two kinds of observations are derived by two separate simulations (see lower part Figure 3). The results for the non-affected observations are obtained by simulation of the error free system S_0. Results for the affected observations are obtained by a simulation of a system S_f in which errors are *injected* (i.e. forced to occur). Injection is carried out at moments that the system is assumed to be in steady-state of the error free system. In our implementation of IS the affected system is a system with an error injected every n_f token passes, with n_f chosen

1. In [4] Injection Simulation was introduced as Fault Injection Simulation.

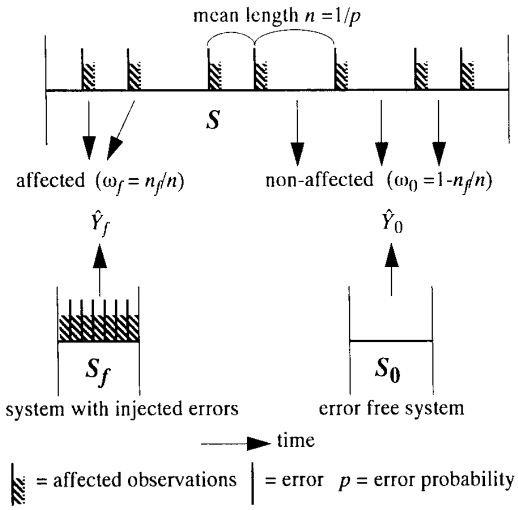


Figure 3 Injection Simulation - Separation of simulations

such that the influence of the occurred error can be considered to be faded out after n_f token passes. The estimator in the affected system is denoted by $\hat{Y}_f$, the estimator in the error free system by $\hat{Y}_0$. These estimators will be multiplied with the so-called weighting factors ω_f and ω_0. These weighting factors represent the fraction of token passes during which the observations are considered to be affected and non-affected respectively. With $n = 1/p$ we have:

$$\omega_f = \frac{n_f}{n}, \text{ and consequently, } \omega_0 = 1 - \omega_f = 1 - \frac{n_f}{n}. \quad (1)$$

The IS estimator $\hat{Y}_{IS}$ then is:

$$\hat{Y}_{IS} = \omega_f \hat{Y}_f + \omega_0 \hat{Y}_0. \quad (2)$$

Notice that from this definition it follows that the IS estimator $\hat{Y}_{IS}$ is a function in p (or $1/n$), the error probability. So, IS has an extrapolative character, i.e. it gives results for a range of error probabilities by only one simulation study.

3.2 IS for the ratio estimator of the probability of missing a deadline

When we want to estimate $\bar{F}(\alpha)$, the probability a job (frame) has to wait longer in the queue than some predefined value α, we consider an estimator which actually is a ratio of two estimators. The results in [4] will therefore be extended in this section for the case of a ratio estimator.

To estimate $\bar{F}(\alpha)$ we use the following expression:

$$\bar{F}(\alpha) = \frac{M(\alpha)}{T},$$

with:

$M(\alpha)$ = the mean number of jobs missing the deadline α per token pass;

T = the mean number of jobs served per token pass.

The reason to take measures which are defined *per token pass* is that the probability of an error occurrence is defined per token pass. It is essentially for this reason that it is inevitable to use a ratio estimator.

Using ^ to indicate estimators and the subscripts f and 0 for respectively the affected and error free system, we obtain from (2) the following expression for the IS estimator $\hat{\bar{F}}_{IS}(\alpha)$ of the deadline missing probability:

$$\hat{\bar{F}}_{IS}(\alpha) = \frac{\hat{M}_{IS}(\alpha)}{\hat{T}_{IS}} = \frac{\omega_f \hat{M}_f(\alpha) + \omega_0 \hat{M}_0(\alpha)}{\omega_f \hat{T}_f + \omega_0 \hat{T}_0}. \quad (3)$$

For this estimator we will list some unbiasedness and variance reduction properties. It extends the results in [4] and [12] to the case of a ratio estimator. More detailed derivations of the results can be found in [13]. For obtaining results about the variance we use the method of batch means (see e.g. [14]). A batch consists of the observations during a number of consecutive token passes. This number is chosen large enough to make it possible to consider different batches as independent. We take as batch size n_f token passes, as we think that the influence of the begin state of the batch is faded out after n_f token passes. In the following we assume that we simulate a total of m batches of which βm are dedicated to the affected simulation and $(1-\beta)m$ to the error free simulation. We call β the *allocation fraction.*

Unbiasedness

Let the observations in the affected system all be distributed identically to a random variable (r.v.) Y_f, and all observations in the error free simulation be identically distributed to the r.v. Y_0. When the observations in the original system S are all distributed identically to the r.v. Y such that:

$$Y = \begin{cases} Y_f, & \text{with probability } \omega_f \\ Y_0, & \text{with probability } \omega_0, \end{cases}$$

then the IS estimator $\hat{Y}_{IS}$, as defined in (2), is unbiased, i.e. $E\hat{Y}_{IS} = EY$. This has been shown in [4][2]. For the ratio estimator in (3) both the numerator $\hat{M}_{IS}(\alpha)$ and the denominator $\hat{T}_{IS}$ in are thus unbiased when this assumption holds. For our simulation examples in Section 4 we discuss to what extent the above assumption is satisfied, as it is the correctness of this assumption which decides on the correctness of applying IS. For the ratio estimator $\hat{\bar{F}}_{IS}(\alpha)$ it is known that the bias is of the order $1/m$, with m the number of batches (see e.g. [15]). So for m large, as will be the case in the simulations in Section 4, this bias becomes negligible.

Variance reduction

An important aspect is the variance reduction capability of IS as this is directly related to the obtained time saving by the simulations. We base our deduction of the confidence interval on work for regenerative simulation in [16] and for Stratified Sampling in [17]. In our simulation study we will use the

2. In [4] this is described more precisely, taking into account the periodicity of the affected system with its injected errors every n_f token passes.

standard estimators (see e.g. [14]) to compute confidence intervals with these formulas. Let for the numerator $M_{IS}(\alpha)$ in (3), the results for the batches in the affected simulation be distributed identically to the r.v. M_f, with mean EM_f and variance $\sigma^2(M_f)$, and let the results for the batches in the non-affected simulation be distributed identically to the r.v. M_0 (mean EM_0, variance $\sigma^2(M_0)$). Similar definitions will be used for the denominator T_{IS} in (3). For the IS ratio estimator $\hat{F}_{IS}(\alpha)$ in (3) batches are distributed identically to a r.v. F_{IS} with mean EF_{IS} and variance σ^2_{IS}.

Now define:

$$\sigma_f^2 \equiv \sigma^2(M_f) - 2EF_{IS}Cov(M_f, T_f) + (EF_{IS})^2\sigma^2(T_f) \quad (4)$$

and define σ_0^2 in a similar way. Further defining:

$$\sigma^2_{IS}(\beta) \equiv \frac{\omega_f^2\sigma_f^2}{\beta} + \frac{\omega_0^2\sigma_0^2}{(1-\beta)}, \quad (5)$$

it can be shown (see [13] for more details) that the confidence interval is of the following form (for large m):

$$[EF_{IS} - \frac{z_\gamma\sigma_{IS}(\beta)}{ET_{IS}\sqrt{m}}, EF_{IS} + \frac{z_\gamma\sigma_{IS}(\beta)}{ET_{IS}\sqrt{m}}], \quad (6)$$

with z_γ the appropriate point in the standard normal distribution Φ ($z_\gamma = \Phi^{-1}(\frac{\gamma}{2})$, with γ the desired accuracy).

It follows directly from (5) and (6) that the length of the confidence interval depends on the allocation fraction β. The optimal value for the allocation fraction, the so-called *optimal allocation fraction* β_{opt} equals [13]:

$$\beta_{opt} = \frac{\omega_f\sigma_f}{\omega_f\sigma_f + \omega_0\sigma_0}. \quad (7)$$

The variance of the IS estimator then equals:

$$\sigma^2_{IS}(\beta_{opt}) = (\omega_f\sigma_f + \omega_0\sigma_0)^2. \quad (8)$$

To compare the length of the confidence interval of the IS estimator with the variance of the direct estimator we use the following assumption (as in [4]). Batches in the direct simulation are distributed identically to the r.v. M_d, with variance σ_d^2 and with the following property:

$$M_d = \begin{cases} M_f, & \text{with probability } \omega_f, \\ M_0, & \text{with probability } \omega_0, \end{cases}$$

A similar assumption can be made for the denominator T. This assumption states that we can identify batches from the two separate simulations in IS, in a direct simulation of the original system. Then it holds that the variance of the direct estimator can be estimated by σ_d^2 / ET_{IS} (see [13]), with:

$$\sigma_d^2 = (\omega_f\sigma_f + \omega_0\sigma_0)^2 + \omega_f\omega_0(\sigma_f - \sigma_0)^2 + $$
$$+ \omega_f\omega_0(EM_f - EF_{IS}ET_f - (EM_0 - EF_{IS}ET_0))^2. \quad (9)$$

From the expressions above the obtained time saving factor can be derived by dividing (9) by (5). The maximally obtainable speed up factor is reached when using the optimal allocation fraction in the IS simulation. The speed up factor is then given by the quotient of (9) and (8). Notice that (8) equals the first right hand side term of (9).

4 Application of Injection Simulation

In this section we discuss the application of Injection Simulation for the performance analysis of the FDDI fault tolerance mechanisms. In Section 4.1 we first discuss two implementation issues of IS: the choice of the length n_f of the affected period and the use of the optimal allocation fraction. Then we show simulation results of IS in Section 4.2. In Section 4.3 we discuss the validity of applying IS for the FDDI model.

4.1 Implementation of Injection Simulation

Determining the length of the affected period

An important aspect in the implementation of IS is the choice of n_f, the length of the affected period. The problem of choosing an appropriate value for n_f closely resembles the problem of deciding on the length of the initial transient period of a steady-state simulation (see Pawlikowski [19] for a survey of the results in this area). We decide on a suitable length by doing a pilot run and validate the chosen n_f by doing a check after the simulation. In the pilot run some value for n_f is chosen and the observations during the last 20% of the n_f affected token passes in all affected batches are compared with results of a short error free simulation. The result for this part of the affected observations should be close to the result for the error free system, as the effect of an occurred error should have become almost negligible. The results of the pilot run will not be very accurate yet but give an idea of an appropriate value for n_f. In the check after the simulation again the observations in all affected batches during the last 20% of the n_f token passes are compared with the error free observations. This results in a more accurate idea of the validity of the chosen value for n_f.

Dynamically adjusting the allocation fraction

To make use of the variance reduction of IS as much as possible, we would like to do the IS simulation with the optimal allocation fraction (7). This optimal allocation fraction depends on the error probability. So, we choose a reasonable error probability and derive the correct value for β_{opt} as follows. We first simulate 100 affected as well as 100 non-affected batches, and then estimate β_{opt} by (7). Then we simulate new affected and non-affected batches such that the resulting fraction of affected batches is closest to β_{opt}. To limit the risk of having a substantial error in the estimate of the optimal allocation fraction, we compute a new and more accurate β_{opt} every 1000 batches. The simulation is stopped when the missing deadline probability is with 90 percent certainty within 10 percent of the estimate, i.e., the length of the 90 percent confidence interval is two tenths of the point estimate.

In our implementation of IS we have chosen to let the allocation fraction be an element of the interval [0.1,0.9]. In this way stable estimation of the confidence intervals is achieved.

4.2 Simulation results

To set the stage for a discussion about simulation results for FDDI using IS we first list the chosen parameter values in Section 4.2.1. Then we present the obtained results in two subsections. In Section 4.2.2 we first discuss results for a fixed error probability $p = 5\times10^{-5}$ and varying load and packet length. In Section 4.2.3 we extrapolate the results of the system with fixed load $\rho = 0.5$ and packet length 2560 bits to a range of error probabilities. In both cases we will look at the probability $\bar{F}(\alpha)$ of missing the deadline α, the optimal allocation fraction β_{opt}, and the speed up factor.

4.2.1 Parameter values

In Table 1 we list the parameter choices for the polling model of the FDDI ring with token loss. We study a symmetric system, i.e., the parameter values are independent of the station number. We want to see how the performance of the ring behaves under different traffic conditions when errors, such as token losses, occur. Therefore, we will vary the arrival rate λ and the average service or transmission time $1/\mu$. We assume the packet lengths to be constant 1280, 2560 (based on [20]) or 3840 bits. The deadline we consider is $\alpha = 2$ TTRT, with TTRT equal to N/μ, the number of stations times the bandwidth per station (1-limited service discipline). Note that, when no errors occur, the FDDI protocol guarantees station access within this period and that this deadline α changes with the required service time per packet. The recovery time is chosen to be uniformly distributed between 0.5 and 1.5 times TTRT. This recovery time approximates the sum of three periods: the time until error discovery (by time outs), the sending of Claim Token Frames (between 1 and 2 cycles) and the cycle of the token for the parameter setting [3].

Special care has to be taken when we model the token return strategy (see Section 2). We think that in many applications the natural return strategy to assume is the fixed station strategy. For example in systems with a fixed configuration the same station might always win the bidding in the Claim Token process. When we use Injection Simulation with the error injected every n_f token passes, letting the token return to the same station after every token loss would lead to periodic behavior of the model. In our example with $N = 20$ and $n_f = 1000$, the token would return to the same station where the error has occurred, because exactly 50 cycles separate consecutive token returns and errors. We therefore model the fixed station return strategy by letting the token return to a random station. This is justified by the fact that it does not matter whether the fixed or random station return takes place in the system, given that the system is symmetric and errors occur in steady-state. Then only the distribution of the difference between the numbers of the station where the token gets lost and where it returns is of importance. This distribution is equal for the fixed and random station return strategy.

Number of stations	$N = 20$
Utilization	$\rho = 0.25, 0.50, 0.75$
Mean transmission time	$1/\mu = 1.28, 2.56, 3.84.10^{-5}$ s.
Arrival rate (Poisson)	$\lambda = (\rho.\mu)/N$
Target Token Rotation Time	$\text{TTRT} = N/\mu$
Deadline	$\alpha = 2$ TTRT
Recovery time (Uniform)	$0.5 \text{ TTRT} \le \alpha \le 1.5 \text{ TTRT}$
Error probability	$p = 5\times10^{-5}$
Switch over time (500 meter)	$ER = 4.22\times10^{-6}$
Affected period	$n_f = 1000$ token passes

Table 1 Parameter choices

$\bar{F}(\alpha)$	$\rho = 0.25$	$\rho = 0.5$	$\rho = 0.75$
$1/\mu = 1.28\times10^{-5}$ $\alpha = 5.12\times10^{-4}$	1.3×10^{-4}	1.1×10^{-2}	0.80
$1/\mu = 2.56\times10^{-5}$ $\alpha = 1.024\times10^{-3}$	2.8×10^{-5}	4.3×10^{-4}	5.26×10^{-2}
$1/\mu = 3.84\times10^{-5}$ $\alpha = 1.536\times10^{-3}$	1.3×10^{-5}	2.5×10^{-4}	1.2×10^{-2}

Table 2 Probability of missing the deadline, fixed error probability

4.2.2 Results for fixed error probability

For fixed error probability $p = 5\times10^{-5}$ we subsequently consider the probability of missing the deadline, the optimal allocation fraction and the speed up factor.

Probability of missing the deadline

In Table 2 the deadline missing probabilities $\bar{F}(\alpha)$ for the different utilizations and service times from Table 1 are shown. Notice that for every row in the table the deadline of $\alpha = 2 \text{ TTRT} = 2N/\mu$ is the same for all three load values. The probability of missing the deadline grows fast with the increase of the utilization. Also, when the packet length decreases, the missing deadline probability increases. This can be explained by the fact that with the 1-limited service discipline every packet has overhead in the form of switching over of the token.

Special care has to be taken for the results derived for the low utilization values. For the error free model the results are close to zero, which can lead to estimation problems because very long simulations are necessary to let jobs miss their deadline often enough. This type of simulation is known in literature as rare event simulation. This rare event type shows up in the output, contrary to the rare event we consider in this paper, which is an element of the system behavior. To solve the problem of estimating small probabilities we can make use of the important benefit that in the two separate simulations of IS every technique can be used. So, when analytical results for the error free part are available these might be used. For the problem of simulating small probabilities, fast simulation

β_{opt}	ρ = 0.25	ρ = 0.5	ρ = 0.75
$1/\mu = 1.28\times10^{-5}$ $\alpha = 5.12\times10^{-4}$	0.64	0.09	0.04
$1/\mu = 2.56\times10^{-5}$ $\alpha = 1.024\times10^{-3}$	0.98	0.57	0.12
$1/\mu = 3.84\times10^{-5}$ $\alpha = 1.536\times10^{-3}$	0.99	0.94	0.16

Table 3 Optimal allocation fraction, fixed error probability

Speed up factor	ρ = 0.25	ρ = 0.5	ρ = 0.75
$1/\mu = 1.28\times10^{-5}$ $\alpha = 5.12\times10^{-4}$	9.3 5600 bts	1.0 477 bts	0.6 200 bts
$1/\mu = 2.56\times10^{-5}$ $\alpha = 1.024\times10^{-3}$	18.2 14746 bts	8.3 2866 bts	1.1 330 bts
$1/\mu = 3.84\times10^{-5}$ $\alpha = 1.536\times10^{-3}$	18.0 61246 bts	19.7 2149 bts	1.5 923 bts

Table 4 Speed up factor, fixed error probability

techniques like those based on Extreme Value Theory [18], could be used.

Optimal allocation fraction

In Table 3 the optimal allocation fraction β_{opt} for the error probability $p = 5\times10^{-5}$ is shown. It denotes what percentage of the observations in the IS simulation should be dedicated to simulation of the affected system. Given the error probability we see from (7) that the value of the optimal allocation fraction depends on the expression for the variance of the affected and non-affected observations in (4). The value of β_{opt} says whether the influence of errors is such that it is necessary to have results for more affected observations than in the direct simulation. In a direct observations we would have, on the long run, a fraction ω_f of affected observations. In our simulation study $\omega_f = n_f p = 0.05$. When β_{opt} is high, relative to ω_f, the affected observations are thus important and should be simulated in a greater amount. In a direct simulation deriving information about the relative importance of affected and non-affected observations requires much more simulation time for small error probabilities. So, IS gives information, in a relatively short simulation, about the influence of occurring errors, and thus about the importance of simulating more errors. This is an extra benefit of Injection Simulation. Combining Table 3 and Table 2 we see that, both in the columns and rows, the relative importance of the affected observations increases when the missing deadline probability decreases. In many cases we should dedicate over 50 percent of our time to the affected simulation. For the highest load $\rho = 0.75$ and shortest packet length of 1280 bits, we see that it is not necessary to simulate more errors than in a direct simulation would occur. Nevertheless IS would also reduce the simulation time when ω_f would have been chosen as allocation fraction, because the variability in the number of errors that occurs in a direct simulation is taken out of the system. From arguments in Cochran [17] and from [13] it follows that taking ω_f as allocation fraction would lead to a variance reduction equal to the right most term of (9).

Speed up factor

In Table 4 the simulation time saving or speed up factor of the performed IS simulations is shown. This speed up is obtained by applying the algorithm described in Section 4.1 with an allocation fraction which is optimal for $p = 5\times10^{-5}$ (see Table 3). We see that in general the reduction increases when the deadline missing probability decreases. This can be explained by the fact that then the affected observations become of importance to the measure (see Table 3), despite their low weight. The speed up factor depends on various aspects of the model but can reach values over 100 [21]. Earlier in this section we decided to start with 100 affected and 100 non-affected batches. For the highest utilization and shortest packet length we saw in Table 3 that the optimal allocation fraction was 0.04, i.e., 8 out of 200 batches. The value 0.6 in the right upper corner shows the simulation has been slowed down because of simulating in the starting phase too many batches in the affected system.

The number of batches given in Table 4 shows the total number of batches (bts) simulated to obtain a confidence interval such that the actual probability of missing the deadline is with 90 percent certainty within 10 percent of the estimate. We see that in general more batches are needed when the missing probability becomes smaller. This illustrates the problem of simulating small probabilities.

Running the QNAP2 program [21] for 1000 batches on a SUN4 workstation took around one hour computer time. So, estimating the smallest deadline missing probabilities asks for long simulations. We see from Table 4 that speed up is high when the required number of batches is high, which implies that IS speeds up the simulation when this is indeed necessary. The speed up factor 18 for $\rho = 0.25$ and packet length 1536 bits, means a time saving of about 1000 hours or 1 month.

4.2.3 Results for a range of error probabilities

In this section we make use of the extrapolative possibilities of IS. When we do one simulation study for a fixed error probability p, we can derive with (2) results for a range of error probabilities by computing the appropriate weighting factors in

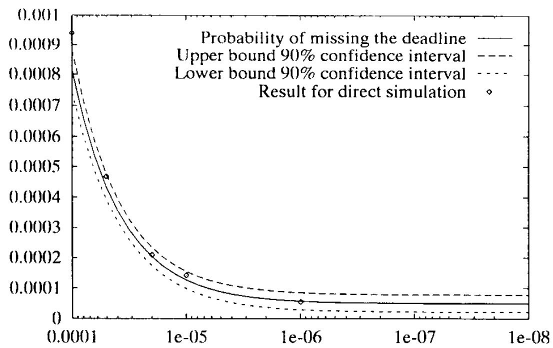


Figure 4 Probability of missing the deadline, for system with ρ = 0.5, *packet length* 2560 *bits.*

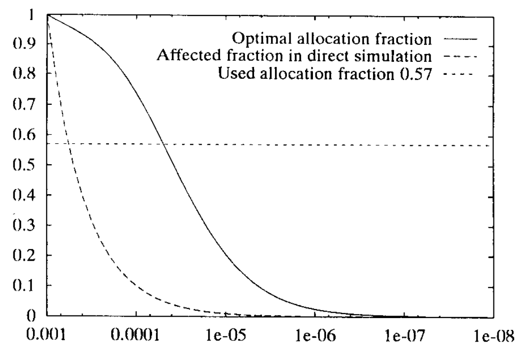


Figure 5 Optimal allocation fraction, for system with ρ = 0.5, *packet length* 2560 *bits.*

(1). Required for the values of p in this range is that the effect of an error can be assumed to fade out, i.e., the system returns to steady-state of the error free system after each error. For higher error probabilities this seems to be a less valid assumption (in Section 4.3 we will come back to this). In this section we look at the system with $1/\mu = 2.56\times10^{-5}$ and $\rho = 0.5$, and choose as range for p: $p \in [10^{-4}, 10^{-8}]$. We again consider respectively the deadline missing probabilities $\bar{F}(\alpha)$, optimal allocation fractions β_{opt} and speed up possibilities.

Probability of missing the deadline

In Figure 4 the deadline missing probabilities $\bar{F}(\alpha)$ for the range of error probabilities $p \in [10^{-4}, 10^{-8}]$ is shown. For the affected part IS gave 6.98×10^{-3} as realization for $\hat{Y}_f$ in (2) and for the non-affected part 5×10^{-5} as realization for $\hat{Y}_0$ in (2). The allocation fraction used is the optimal allocation fraction for $p = 5\times10^{-5}$ ($\beta_{opt} = 0.57$, see Table 3). The central line is the point estimator, the other lines form the borders of the 90 percent confidence interval. From the fact that the line comes close to the results for the error free system (i.e. 5×10^{-5}) when the error probabilities are close to $p = 10^{-8}$ we see that the influence of error occurrences on the probability of missing the deadline becomes almost negligible in that area. As the token loss probability in FDDI is at about 3×10^{-8}, token loss will not degrade the considered QoS for this model. It is not very likely that other possible recovery triggering events (see Section 2) increase the error probability p very much. So, *for the considered model and parameter values* we can conclude that the recovery process in FDDI is fast enough to guarantee the desired performance level in the presence of errors.

Optimal allocation fraction

We can also compute the optimal allocation fraction for other error probabilities than $p = 5\times10^{-5}$. By filling in (1) and (4) we compute the correct values for the variables in (7), thus obtaining the optimal allocation fraction for a range of error probabilities, as indicated by the solid curve in Figure 5. The straight dotted line shows the allocation fraction with which we performed the IS simulation, namely 0.57, the optimal allocation fraction for $p = 5\times10^{-5}$. The dotted curve shows what the expected fraction of affected observations would have been in a direct simulation, namely ω_f. We see from Figure 5 that it is always useful to simulate more errors than in the direct simulation, since the optimal allocation fraction always exceeds the weighting factor. This has to do with the variance terms in (7); the affected observations show more variance. On the other hand it is also not always useful to simulate as many affected observations as done when applying the optimal allocation fraction for $p = 5\times10^{-5}$, since the curve for the optimal allocation fraction lies under the fraction 0.57 for error probabilities smaller than $p = 5\times10^{-5}$.

Speed up factor

For our simulation we have used the optimal allocation fraction for the case $p = 5\times10^{-5}$. The speed up factor over direct simulation for the different error probabilities when this allocation fraction is used, is presented in Figure 6. This speed up factor is obtained by filling in (5) for the used allocation fraction $\beta = 0.57$ and compare it with (9). This is not the maximal possible speed up, because this maximum is reached for a particular error probability when we use the optimal allocation fraction belonging to that error probability. The

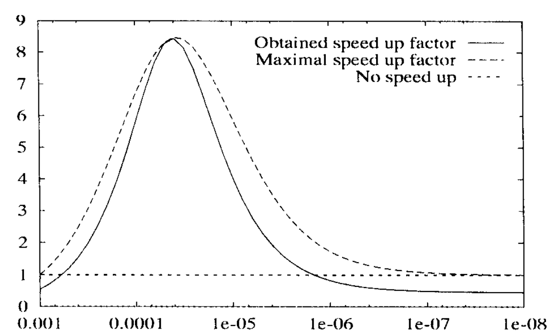


Figure 6 Speed up factor, for system with ρ = 0.5, *packet length* 2560 *bits.*

dotted line in Figure 6 gives the speed up factors we would have achieved when we would have used these individual optimal allocation fraction. It is obtained by estimating and comparing (8) and (9). We see that the optimal allocation fraction, belonging with $p = 5\times10^{-5}$ gives good results for a range $p \in [10^{-4}, 10^{-8}]$, as the lines do not differ much.

4.3 Validity of the Injection Simulation estimator

To determine the correctness of the obtained results we have done direct simulations for different error probabilities p, for the system with $p \in [10^{-4}, 10^{-8}]$, packet length 2560 bits and load $\rho = 0.5$ (as in Section 4.2.3). The ◊-marks in Figure 4 show the results for the point estimates of the direct simulation. We stopped the direct simulations when the confidence interval was within 10 percent of the estimate or when 30,000 batches were simulated (maximal confidence interval 20 percent around point estimate). From Figure 4 it can be seen that the point estimators of the original system are within the bounds of the confidence interval of the IS simulation, except for $p = 10^{-4}$. The latter can be explained by the fact that in the chosen implementation of IS the probability of a second error occurring in the affected period is nil. For high error probabilities this will be a less valid assumption. We note here that another implementation of IS is possible which takes care of errors which happen in the affected period [13]. A disadvantage is that with this extension IS loses a part of its extrapolative character.

The results from the direct simulation match those of the IS simulation. To be sure that for instance not by accident two faulty results together give a correct answer in IS, we also derived results for the validity of the chosen length of the affected period, $n_f = 1000$. The basic idea behind IS is that in the original system the influence of an error has faded out after n_f token passes. The observations within these n_f token passes are the affected observations. Therefore we have collected in the direct simulations the observations within n_f token passes after every occurring error. The other observations give the result for the non-affected observations. IS gives correct results when the results of the affected observations in the direct simulations are similar to the result for the affected observations in IS.

In Table 5 the results for the affected and non-affected observations in the direct simulations, as well as in the IS simulation, are shown. We see that the confidence intervals have an overlap for the affected observations. The confidence interval for the smallest error probabilities are wider because less errors have occurred in the direct simulation. Note that the relatively large confidence intervals for the non-affected observations again demonstrate the problem of estimating small probabilities. When the affected period length n_f would have been chosen too small, the affected observations in IS would have been higher than in the original system as errors would have had influence on each other. This is not the case and thus the chosen affected period is long enough. The slightly lower value for the point estimate for the affected part of IS indicates that the influence of a second error within the affected period is not negligible.

Direct simulation		
Error probability	affected observations	non affected observations
$p = 1.0\times10^{-4}$	$8.00 \pm 0.75\times10^{-3}$	$4.87 \pm 10.5\times10^{-5}$
$p = 5.0\times10^{-5}$	$7.35 \pm 0.73\times10^{-3}$	$5.94 \pm 4.95\times10^{-5}$
$p = 2.0\times10^{-5}$	$7.41 \pm 0.97\times10^{-3}$	$4.31 \pm 2.64\times10^{-5}$
$p = 1.0\times10^{-5}$	$7.90 \pm 1.38\times10^{-3}$	$5.36 \pm 2.05\times10^{-5}$
$p = 1.0\times10^{-6}$	$9.91 \pm 5.91\times10^{-3}$	$4.34 \pm 1.01\times10^{-5}$
IS simulation		
	affected observations	non-affected observations
	$6.98 \pm 0.58\times10^{-3}$	$4.91 \pm 2.80\times10^{-5}$
last 20% affected system in IS:		4.08×10^{-5}

Table 5 Results for affected and non-affected observations from direct simulation and IS simulation

We can also validate the IS simulation by itself, i.e. without having to do direct simulations. This can be done by comparing the result for the last 20 percent of the observations in affected batches with the error free result. In our case we computed the result for the observations during token passes 801 through 1000 in every affected batch in IS (i.e. in S_f). This result should be, and is, almost equal to the error free results in the IS simulation. Both are shown in Table 5.

Concluding this section, we have shown that Injection Simulation can be a useful method to solve the problem of analyzing systems with rarely occurring events. It gives information about the required number of errors to simulate by the computed optimal allocation fraction, has an extrapolative property, and, most important, leads to simulation time saving, thus making it possible to simulate systems that otherwise would have required too much simulation time.

5 Conclusions and further research topics

Although the fault tolerance mechanisms in FDDI are fast and triggered automatically, error occurrences can degrade the Quality of Service to a level which is no longer acceptable for network user or supplier. In this paper we have discussed how to analyze the performance consequences of rarely occurring errors in FDDI. We have developed Injection Simulation, a fast simulation technique, which speeds up the simulation considerably. Furthermore it gives results for a range of error probabilities with only one simulation study, and gives information whether errors influence the measure of interest and whether it is necessary to simulate more errors than in a direct simulation would occur. Injection Simulation can be implemented straightforwardly. The simulation results show the usefulness and the validity of Injection Simulation for analyzing the error recovery process in FDDI. In case the probabilities of missing a deadline are very small, the use of a fast simulation technique within the two simulation parts of IS, can result in an even more powerful simulation technique. Further topics of interest regarding IS most of all have to do with determining the length of the affected period. Concerning FDDI, analyzing the consequences of physical resource errors, which lead to reconfiguration of the ring, is of interest. Also in this case IS might be a powerful simulation technique to analyze systems with rarely occurring events.

Acknowledgment

We thank Johan Doeven for the QNAP2 implementation of Injection Simulation for FDDI [21].

References

[1] **International Organization for Standardization,** *Information Processing Systems - Fibre Distributed Data Interface, Part I, II and III, ISO 9314-1 1989, ISO 9314-2 1989, and ISO 9314-3 1990.*

[2] **K.C. Sevcik and M.J. Johnson,** "Cycle time properties of the FDDI token ring protocol," *IEEE Transactions on Software Engineering,"* vol. 13, no. 3, March 1987, pp. 376-385.

[3] **M.J. Johnson,** "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Proceedings of the 10th Conference on Local Computer Networks,* Minneapolis 1985, pp. 124-133.

[4] **A.P.A. van Moorsel, B.R. Haverkort and I.G. Niemegeers,** "Fault Injection Simulation: A variance reduction technique for systems with rare events," in *Proceedings of the Second International Working Conference on Dependable Computing for Critical Applications,* Tucson, AZ, February 18-20 1991, pp. 57-64; Forthcoming in *Dependable Computing and Fault-Tolerant Systems,* vol. 6, J.F Meyer, R.D. Schlichting, Eds., Springer-Verlag, 1991.

[5] **F.E. Ross,** "An overview of FDDI: The Fiber Distributed Data Interface," *IEEE Journal on Selected Areas in Communications,* vol. 7, no. 7, September 1989, pp. 1043-1051.

[6] **H. Takagi,** "Queueing analysis of polling models: An update," *Stochastic Analysis of Computer and Communication Systems,* H. Takagi, Ed., Elsevier Science Publishers, 1990, pp. 267-318.

[7] **W.I. Genter and K.S. Vastola,** "Delay analysis of the FDDI synchronous data class," *IEEE INFOCOM '90,* 1990, pp.766-773.

[8] **A.P.A. van Moorsel, B.R. Haverkort and I.G. Niemegeers**, "Simulation of failure prone token rings using fault injection simulation," *Memoranda Informatica 90-44,* University of Twente, Tele-Informatics and Open Systems, Enschede, The Netherlands, August 1990.

[9] **V.S. Frost, W.W. Larue Jr. and K.S. Shanmugan,** "Efficient techniques for the simulation of computer communication networks," *IEEE Journal on Selected Areas in Communications,* vol. 6, no. 1, pp. 146-157, January 1988.

[10] **A. Goyal, P. Shahabuddin, Ph. Heidelberger, V.F. Nicola and P.W. Glynn,** "A unified framework for simulating Markovian models of highly dependable systems," *IBM Research Report RC 14772,* November 1989 (to appear in IEEE Transactions on Computers).

[11] **G. Carter and E.J. Ignall,** "Virtual measures: A variance reduction technique for simulation," *Management Science,* vol. 21, pp. 607-616, February 1975.

[12] **A.P.A. van Moorsel, B.R. Haverkort and I.G. Niemegeers,** "Fault Injection Simulation: A variance reduction technique for studying the performance consequences of rarely occurring failures in communication networks," in *Proceedings of the First International Workshop on Performability Modelling of Computer and Communication Systems,* University of Twente, The Netherlands, February 1991, pp. 99-105.

[13] **A.P.A. van Moorsel,** "Bias and variance considerations in Injection Simulation," *Memoranda Informatica 91-57,* University of Twente, Tele-Informatics and Open Systems, Enschede, The Netherlands, June 1991.

[14] **P.D. Welch,** "The statistical analysis of simulation results," in *Computer Performance Modelling Handbook,* S.S. Lavenberg, Ed., New York, Academic Press, 1983, pp. 267-329.

[15] **G.S. Fishman,** *Principles of Discrete Event Simulation,* John Wiley & Sons, New York, 1978.

[16] **M.A. Crane and A.J. Lemoine,** "An introduction to the regenerative method for simulation analysis," in *Lecture Notes in Control and Information Sciences,* A.V. Balakrishnan, M. Thoma, Eds., Springer-Verlag, Berlin, 1977.

[17] **W.G. Cochran,** *Sampling Techniques,* New York, John Wiley & Sons, 1977.

[18] **M. Guida, D. Iovino and M. Longo,** "Comparative performance analysis of some extrapolative estimators of probability tails," *IEEE Journal on Selected Areas in Communications,* vol. 6, no. 1, pp. 76-84, January 1988.

[19] **K. Pawlikowski,** "Steady-state simulation of queueing processes: A survey of problems and solutions," *ACM Computer Surveys,* vol. 22, pp. 123-170, June 1990.

[20] **A. Villalobos, A.M. Martinez, L. Vidaller and C. Moran,** "Real-time applications on FDDI," *in Proceedings of EFOC/LAN,* 1989, pp. 162-166.

[21] **J. Doeven,** *Using Fault Injection Simulation for the Performance Analysis of the Fibre Distributed Data Interface,* Report 075-TV-9115, University of Twente, Tele-Informatics and Open Systems, Enschede, The Netherlands, June 1991.