Reliability Modeling of Proactive Fault Handling

Felix Salfner and Miroslaw Malek
Department of Computer Science
Humboldt-Universität zu Berlin
Berlin, Germany
{salfner|malek}@informatik.hu-berlin.de

Abstract

Research on dependable computing is undergoing a shift from traditional fault tolerance towards techniques that handle faults proactively. These techniques comprise two parts: (a) prediction of failures and (b) actions that are performed in case of an upcoming failure. This work provides the first reliability model that incorporates both correct and false predictions as well as both types of actions: failure prevention and recovery preparation. Closed form solutions to availability, reliability and hazard rate are provided.

Index Terms

Proactive fault handling, reliability, availability, failure prediction, preventive actions, recovery preparation.

Acronyms

CTMC	continuous time Markov chain
PFH	proactive fault handling
TTF	time to failure
TTR	time to repair
TTP	time to prediction
MTTF	mean time to failure
MTTR	mean time to repair
MTTP	mean time to prediction
TP	true positive
FP	false positive
TN	true negative
FN	false negative

Notation

n	number of predictions
n_F	number of failures
$n_{\overline{F}}$	number of non-failures
n_W	number of warnings
n_{TP}	number of true positives
n_{FP}	number of false positives
n_{TN}	number of true negatives
n_{FN}	number of false negatives
λ_p	Prediction rate
λ_{TP}	rate of true positive predictions

This work was supported in part by Intel Corporation and Deutsche Forschungsgemeinschaft DFG project MA 1773 / 3-1

λ_{FP}	rate of false positive predictions
λ_{TN}	rate of true negative predictions
λ_{FN}	rate of false negative predictions
ρ	reaction rate
μ	repair rate
Δl	lead time of failure predictor
P_{TP}	failure probability in case of true positive predictions
P_{FP}	failure probability in case of false positive predictions
P_{TN}	failure probability in case of true negative predictions
k	repair time improvement factor
p	precision
r	recall
f	false positive rate
Q	infinitesimal generator matrix
π_i	equilibrium probability for state i
A	steady-state availability
$R(t)$	reliability
$h(t)$	hazard rate
$F(t)$	cumulative distribution of TTF
$f(t)$	distribution density of TTF
α	initial state distribution

I. Introduction

ANTICIPATING failures before they occur and applying preventive strategies to avoid them or reducing time-to-repair by preparation for upcoming failures is a promising approach to further enhancement of system dependability. Since introduction of preventive maintenance a few decades ago, it has become an increasingly significant area in dependability research in the past few years, as can be seen from autonomic computing initiative [1], trustworthy computing [2], recovery-oriented computing [3], work on rejuvenation (e.g., [4]) and various conferences on self-*properties (see, e.g., [5]).

Proactive fault handling techniques subsume all those methods that build on *online failure prediction* to proactively deal with faults in order to prevent a system failure or to minimize downtime. This includes all methods where either preventive actions are triggered by the prediction of an upcoming failure or where repair actions are prepared for the upcoming failure such that TTR is reduced. Fig. 1 visualizes the approach.

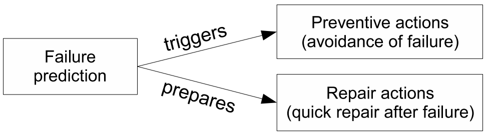

Failure prediction
triggers
prepares
Preventive actions (avoidance of failure)
Repair actions (quick repair after failure)

Fig. 1. PFH is the combination of failure prediction and proactive actions. Either preventive actions are triggered before the failure occurs or repair mechanisms are prepared for an imminent failure to reduce TTR.

The goal of this article is to investigate the effect of PFH on system reliability and steady-state availability in a comprehensive way that allows to model all the various methods that are encompassed by PFH. The model we have developed can help to evaluate application of PFH techniques for a given system or to investigate what is most effective to further increase system reliability / availability.

In order to develop our generic framework for reliability modeling, we first illustrate the working principle and present a categorized overview of preventive as well as repair actions. Reliability modeling is carried out by the use of two CTMC models prescinding the key features of PFH. The transition rates of the CTMCs are determined by metrics capturing the essential properties of

PFH such as precision of failure prediction or repair time improvement. The CTMCs are analyzed and key quantities such as steady-state availability A or reliability $R(t)$ are computed. In order to apply the presented modeling approach to a real system, the metrics used for modeling have to be estimated from measurement data. Therefore, an estimation procedure is described in the last part of this work.

The article is structured as follows: In Section II explains the working principle is explained and a classification of preventive and repair actions are introduced. Next, in Section III the measures used for assessment are introduced. In Section IV and V availability and reliability modeling are described and an example is presented in Section VI. Finally, in Section VII a procedure for extracting modeling metrics from measurement data is outlined followed by conclusions.

II. Proactive Fault Handling

Proactive reaction to faults is at first glance closely coupled with fault detection: A fault needs to be detected before a system can react to it. However to be precise, not just a fault but mainly the failure is the kind of event that should be avoided, which makes a big difference especially in the case of fault-tolerant systems. Hence, efficient proactive fault handling requires the prediction of *failures*, to judge whether a faulty situation bears the risk of a failure or not. To achieve this, an online failure predictor is necessary to continuously monitor the system in order to make a decision whether some failure seems to occur in the near future or not.

If the failure predictor's analysis suggests that a failure will occur, it raises a *failure warning*. It is obvious that any failure predictor can make wrong predictions: the predictor might forecast an upcoming failure even if this is not the case, which is called FP, or the predictor might miss to predict a failure that is imminent in the system, which is called FN. Tab. I lists all four cases that may occur.

TABLE I
Four cases of failure prediction

	Imminent failure (F)	**No imminent failure ($\overline{F}$)**
warning (W)	correct warning (TP)	false warning (FP)
no warning ($\overline{W}$)	missing warning (FN)	correct no-warning (TN)

Raising a failure warning leads either to triggering of a preventive action trying to avoid the failure or it leads to preparation of a repair action for the upcoming failure such that TTR can be reduced. If the failure predictor's analysis suggests that the system is running well and hence no failure is anticipated in the near future, no action occurs. In case of a FP prediction, preventive actions or preparation of repair actions are performed unnecessarily and in case of a FN prediction, nothing is done about the upcoming failure and standard repair is taking place after the failure has occurred. Tab. II summarizes all four cases.

TABLE II
Actions performed after prediction

Prediction	**Preventive actions**	**Repair actions**
TP	Try to prevent failure	Prepare repair
TN	No action	No action
FP	Unnecessary action	Unnecessary preparation
FN	No action	Standard repair

In order to give an idea of the types of actions that are covered by our modeling approach, the following sections will briefly describe repair as well as preventive actions covered by PFH.

A. Repair actions

Repairing the system after failure occurrence is the classical way of failure handling. It is based on detection mechanisms such as coding checks, replication checks, timing checks or plausibility checks. Within PFH, these repair actions do still react to failures, but if its occurrence is anticipated a preparation might take place which in turn may reduce time-to-repair.

The goal of repair actions is recover from failure and bring the system into a consistent state. If the consistent state is a previous fault-free one (a so-called *checkpoint*), the action applies a *roll-backward scheme*. All computations from the last checkpoint up to the time of failure occurrence have to be recomputed. Typical examples are recovery from a checkpoint or the recovery block scheme introduced by [6]. In case of a *roll-forward scheme*, the system is moved forward to a consistent state by either dropping or approximating the computations that have failed.

Both schemes may comprise *reconfiguration* such as switching to a hardware spare or another version of a software program, changing network routing, etc. Reconfiguration takes place before computations are redone or approximated.

In the traditional case without PFH, checkpoints are saved independently of upcoming failures, e.g., periodically. When a failure occurs, reconfiguration takes place until the system is ready for recomputation / approximation and all the computations from the last checkpoint up to the time of failure occurrence have to be redone. TTR is determined by two factors: time needed for reconfiguration and the time needed for recomputation or approximation of lost computations, which is determined by the length of the time interval between the checkpoint and the time of failure occurrence. In some cases recomputation may take less time than originally but the implication still holds. Please also note that not all repair actions exhibit both factors contributing to TTR. Fig. 2-a shows the interrelation.

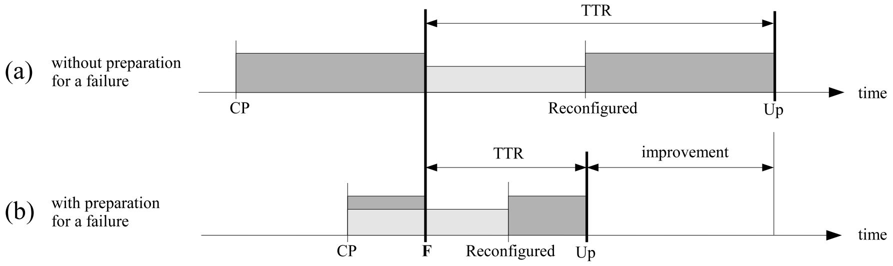


Fig. 2. Improved TTR for prediction-driven repair schemes. (a) sketches classical recovery and (b) improved recovery in case of preparation for an upcoming failure. “CP” denotes the last checkpoint before failure, “F” the time of failure occurrence, “Reconfigured” the time when reconfiguration has finished and “Up” the time when the system is up again.

A large variety of repair actions exist that can benefit from failure prediction and it seems infeasible to list all of them here. Anyway, coupling with a failure predictor can in principle reduce both factors of TTR. Time needed for reconfiguration can be reduced since reconfiguration can be prepared for an upcoming failure. Think, for example, of a cold spare: Booting the spare machine can be started right after an upcoming failure has been predicted such that it is almost up when the failure occurs. Additionally, PFH allows to save a checkpoint close to the failure which reduces the amount of computations that need to be repeated and hence minimizes the amount of time consumed by recomputations. Fig. 2-b sketches the effects. On the other hand when a failure is anticipated it might not be wise to take a checkpoint since the state might be corrupted already but as a preparation for such a case we may take additional periodic checkpoints and use them if necessary.

B. Preventive actions

Preventive actions are triggered by a failure predictor in order to prevent the occurrence of a failure that seems to be imminent in the system but has not yet occurred. We have identified four categories of mechanisms that can anticipate failures before they appear: preventive restarts, state clean-up, preventive failover and system mollification.

- *Preventive restarts* try to avoid upcoming failures by reset. This may range from component restart up to complete reboot of a system. One example is software rejuvenation [7], which is about preventively restarting components to counteract aging of software (e.g., memory leaks).
- *State clean-up* tries to avoid failures by cleaning up resources. Examples include garbage collection, clearance of queues, correction of corrupt data or elimination of useless processing.
- *Preventive failover* techniques perform a switch to some spare hardware or software unit. Several variants of this technique exist. For example, failure prediction-driven load balancing can accomplish gradual "failover" from a failure-prone to a failure-free component [8].
- *System mollification* (ease-up) is a common way to prevent failures. For example, web-servers reject connection requests in order not to become overloaded. Within proactive fault handling, the number of allowed connections is adaptive and would depend on the risk of failure.

Preventive actions affect TTF since in case of successful failure avoidance TTF is increased. However, if avoidance does not succeed, nothing is done to improve repair and hence TTR remains unchanged.

III. Assessing Proactive Fault Handling

In order to assess the effect of PFH on quantities such as $R(t)$, measures have to be identified to quantify the effectiveness of a PFH approach. In particular, effectiveness assessment implies measurement of the failure predictor's accuracy, success rate of preventive actions and efficiency of repair actions.

A. Failure prediction

Precision and recall, originally defined to evaluate information retrieval strategies [9], are frequently used to express prediction quality.

Precision is the ratio of the number of correctly identified failures to the number of all positive predictions (warnings):

$$p = \frac{n_{TP}}{n_{TP} + n_{FP}} = \frac{n_{TP}}{n_W} \tag{1}$$

Recall is defined as the ratio of the number of correctly predicted failures to the total number of failures that actually occurred. Recall is sometimes also called true positive rate.

$$r = \frac{n_{TP}}{n_{TP} + n_{FN}} = \frac{n_{TP}}{n_F} \tag{2}$$

Consider the following example for clarification: If a prediction algorithm achieves precision of 0.8, a generated failure warning is correct (refers to a true failure) in 80% of all cases and 20% are false warnings. A recall of 0.9 expresses that 90% of all actual failures are predicted (and 10% are missed). In most cases, precision and recall show an inverse proportionality: improving recall lowers precision and vice versa.

Precision and recall do not take TN predictions into account. False-positive rate is a standard measure accounting for it:

$$f = \frac{n_{FP}}{n_{FP} + n_{TN}} = \frac{n_{FP}}{n_{\overline{F}}} \tag{3}$$

B. Effects on TTR

As have been shown in the previous section, TTR is affected by repair actions. In order to assess reliability, it is sufficient to subsume all effects of repair actions by measuring mean relative improvement of TTR. It is the ratio of MTTR without preparation to MTTR with preparation:

$$k = \frac{MTTR}{MTTR_{prep}} \tag{4}$$

Obviously, we would expect that preparation for upcoming failures improves MTTR, thus $k > 1$, but the definition also allows $k < 1$ corresponding to a change for the worse.

C. Effects on TTF

Avoidance of failures by preventive actions increases TTF. However, the opposite effect can also occur: due to additional load generated by failure prediction and actions, failures can be provoked that would not have occurred if no PFH had been in place. Modeling of system reliability must take both cases into consideration. We do therefore not analyze the effects of preventive actions separately but rather consider the four cases of predictions presented in Tab. I and introduce three probabilities:

P_{TP} is the probability that a failure occurs in case of a correct warning. An effective preventive mechanism that is avoiding most of the failures and is not causing additional ones results in low P_{TP}.

P_{FP} is the probability of failure occurrence in case of a false positive warning. It corresponds to the probability that a failure is provoked by the extra load of failure prediction, preventive actions and preparation of repair actions.

P_{TN} is the probability that an extra failure is provoked by the prediction alone (since it is a negative prediction, no action is triggered / prepared).

There is no need to define a probability for FN predictions since nothing is done about the failure that will occur and the probability is hence equal to 1.

If the PFH system comprises several actions, some dispatcher is necessary in order to decide what action to trigger or to prepare. Decision accuracy of the dispatcher is inherently contained within P_{TP}, P_{FP}, and P_{TN}. Think, for example, of the case that the dispatcher chooses to prepare a repair action instead of triggering a preventive action then the probability of failure occurrence is increased while k is improved.

IV. Modeling Availability

Reliability modeling is most useful for systems that are not yet existing —otherwise reliability could be *measured* rather than modeled. Reliability modeling dates back a few decades and CTMCs have become one of the dominating modeling techniques (see, e.g., [6] for an overview). Models of preventive schemes have been developed along with preventive maintenance in the seventies (an overview is given in [10]), but these models were mainly based on (static) lifetime distributions and have not covered online prediction. In the case of software rejuvenation Huang et al. [7] used CTMCs to demonstrate improved availability / reliability. Those models have been improved over the years to model the rejuvenation process more realistically. Especially a paper by Bao et al. [11] has introduced online measurement and estimation of an upcoming failure. However, to our knowledge none of the models published so far have explicitly (a) modeled the process of failure prediction including false predictions and (b) covered preventive as well as prediction-driven repair actions. The model presented here is therefore based on three types of *parameters* that can be assessed independently:

1) MTTF and MTTR of a system without PFH
2) parameters of prediction techniques that are considered to be used (p, r and f)
3) parameters of the actions under consideration (P_{TP}, P_{FP}, P_{TN}, and k)

We use a CTMC for modeling the process of PFH and to compute steady-state availability A as a function of these parameters.

A. The model

Steady-state availability of a system is usually computed by modeling a repairable system that is either in an “up” or “down” state (See Fig. 3, and [6] for an overview). System failure rate λ determines the transition from “up” to “down” and repair rate μ the transition back to the up-state.

Steady-state availability can then be computed as the equilibrium state probability of the up-state. In order to include failure prediction we have added four failure-prone states corresponding to the

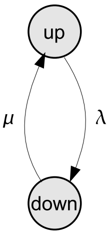


Fig. 3. The simplest CTMC for availability modeling. λ is the failure rate and μ the repair rate.

four cases of prediction correctness. Improved TTR is modeled by two failure states. The case that the system has been prepared for the failure (F_p) or that the failure occurs without preparation ($F_{\bar{p}}$). See Fig. 4.

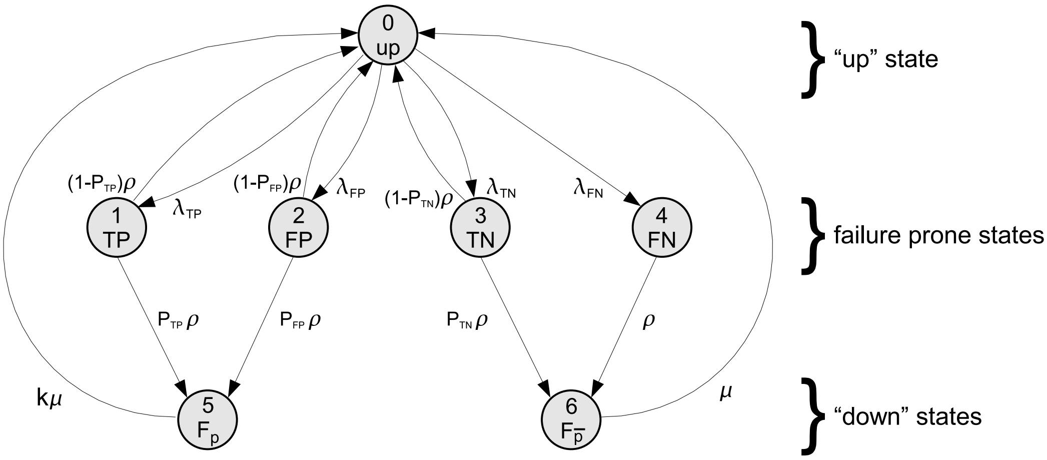


Fig. 4. The CTMC for availability modeling. States 1-4 correspond to the four cases of failure prediction correctness: TP, FP, TN and FN. States 5 and 6 correspond to "down" states where F_P denotes failure handling where the failure had been anticipated and repair was prepared and $F_{\bar{P}}$ accounts for the unprepared counterpart.

Starting from the up-state at some point in time a failure prediction is performed. If the predictor came to the conclusion that a failure was imminent and raised a warning and that was true (something was really going wrong in the system) the prediction is a TP. Due to the warning, preventive actions are triggered and/or repair actions are prepared. As $1 - P_{TP}$ is the success probability for preventing the failure, the system returns to the up-state with probability $1 - P_{TP}$ while with probability P_{TP} a failure occurs and the system enters state F_P. The inverse of the duration of the transitions is reaction rate ρ. In the second case (where the failure occurred) improved repair takes place due to the failure warning raised by the predictor and the subsequent preparation for the failure. The improved repair rate is $k\mu$.

In case of a FP prediction, the predictor came to the false conclusion that something is going wrong in the system and hence actions are performed unnecessarily. Due to the additional workload induced by prediction and preventive as well as preparatory actions, there is some probability (P_{FP}) that a failure is caused by PFH. Therefore, the model transits to state F_P with probability P_{FP} from where it returns to the up-state with improved repair rate.

In case of TN predictions, there is no imminent failure in the system and no warning is raised, which means that no actions are performed. Nevertheless, failure prediction causes some additional load with a certain risk (P_{TN}) that a failure is caused by prediction. In this case the model transits to the unprepared failure $F_{\bar{p}}$ since the predictor has not raised a failure warning and hence no preparation has taken place. The transition back to the up-state is taking place with standard repair rate μ.

If the predictor does not recognize that something is going wrong in the system and that a failure is coming up, the prediction is a FN. Since nothing is done about the upcoming failure there is no transition back to the up-state and the model transits without any preparation to $F_{\bar{p}}$.

In summary every PFH technique should strive to achieve two goals:

1) the portion of true predictions (either TP or TN) should be maximized, and
2) failure occurrence probabilities P_{TP}, P_{FP} and P_{TN} should be as close to zero as possible.

B. Temporal properties of the modeled system

CTMC models express temporal behavior using exponential transition distributions that are determined by a single parameter: the transition rate. This section determines the rates of our model, which are λ_{TP}, λ_{FP}, λ_{TN}, λ_{FN}, ρ, and μ.

In traditional reactive systems, a system was considered to be in a failure-free state until a failure occurs followed by a repair period. The measures MTTF and MTTR naturally arose from this notion. In PFH, failure predictions take place in between the occurrence of failures. Therefore, a switch from MTTF to mean-time-to-prediction MTTP is necessary for our modeling (see Fig. 5-a). Additionally, predictions are performed some time ahead in order to have enough time to perform preventive as well as preparatory actions. This time interval is called *lead time* Δl (see Fig. 5-b).

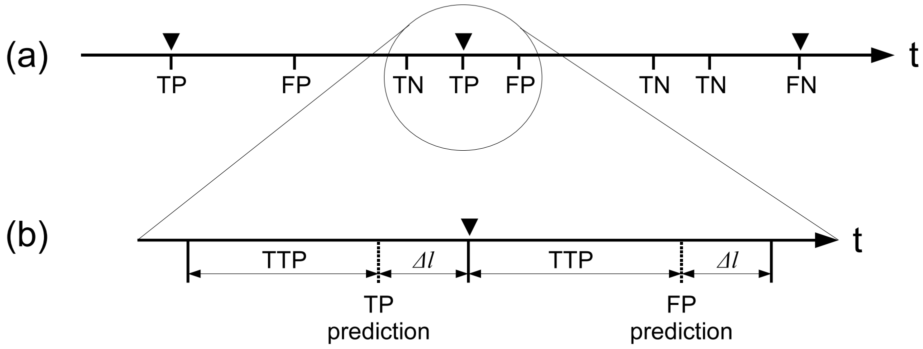


Fig. 5. A timeline showing failures (▼) and predictions (TP, FP, TN, FN). Blow-up (b) shows that predictions take place some lead time Δl ahead. Time to prediction is indicated by TTP.

MTTP can be computed from MTTF of a system without PFH and the parameters of the failure predictor. It can be seen from Fig. 5-a and from Tab. I that the total number of predictions n is determined by:

$$\begin{aligned} n = n_F + n_{\overline{F}} \qquad &= n_F + \frac{n_{FP}}{f} \qquad = n_F + \frac{n_W}{f} - \frac{n_{TP}}{f} \\ = n_F + \frac{n_{TP}}{p\,f} - \frac{n_{TP}}{f} &= n_F + n_{TP}\left(\frac{1}{p\,f} - \frac{1}{f}\right) = n_F + n_F\,r\left(\frac{1}{p\,f} - \frac{1}{f}\right) \\ = n_F\left(1 + r\,\frac{1-p}{p\,f}\right) & \end{aligned} \tag{5}$$

Therefore, there are $\left(1 + r\,\frac{1-p}{p\,f}\right)$ as many predictions as failures. From this follows that MTTP is determined by:

$$MTTP = \frac{MTTF}{\left(1 + r\,\frac{1-p}{p\,f}\right)} - \Delta l \tag{6}$$

and hence the overall prediction rate λ_p is:

$$\lambda_p = \frac{1}{MTTP} \tag{7}$$

The rate of predictions λ_p has to be distributed among the four states TP, FP, TN, and FN. This is achieved by computing the fraction of TP, FP, TN and FN predictions.

Starting with $\frac{TP}{n}$ we have:

$$\begin{aligned} n &= n_F + n_{\overline{F}} \\ &= \frac{n_{TP}}{r} + \frac{n_{FP}}{f} = \frac{n_{TP}}{r} + \frac{n_W - n_{TP}}{f} \\ &= \frac{n_{TP}}{r} + \frac{n_{TP}}{p\,f} - \frac{n_{TP}}{f} \end{aligned} \tag{8}$$

$$\Rightarrow \quad \frac{n_{TP}}{n} = \frac{1}{\frac{1}{r} + \frac{1}{p\,f} - \frac{1}{f}} \tag{9}$$

In order to compute the fraction of FP, FN, and TN predictions, we need to know:

$$\frac{n_W}{n} = \frac{1}{p}\frac{n_{TP}}{n} \tag{10}$$

$$\frac{n_F}{n} = \frac{1}{r}\frac{n_{TP}}{n} \tag{11}$$

and then we have

$$\frac{n_{FP}}{n} = \frac{n_W}{n} - \frac{n_{TP}}{n} \tag{12}$$

$$\frac{n_{FN}}{n} = \frac{n_F}{n} - \frac{n_{TP}}{n} \tag{13}$$

$$\frac{n_{TN}}{n} = \frac{n_{\overline{F}}}{n} - \frac{n_{FP}}{n} = 1 - \frac{n_F}{n} - \frac{n_{FP}}{n} \tag{14}$$

To compute transition rates to the failure-prone states for TP, FP, FN and TN predictions, we distribute the overall prediction rate λ_p according to the fractions:

$$\lambda_{TP} = \frac{n_{TP}}{n} * \lambda_p \tag{15}$$

$$\lambda_{FP} = \frac{n_{FP}}{n} * \lambda_p = \left(\frac{1}{p} - 1\right)\lambda_{TP} \tag{16}$$

$$\lambda_{FN} = \frac{n_{FN}}{n} * \lambda_p = \left(\frac{1}{r} - 1\right)\lambda_{TP} \tag{17}$$

$$\lambda_{TN} = \frac{n_{TN}}{n} * \lambda_p = \lambda_p + \left(1 - \frac{1}{p} - \frac{1}{r}\right)\lambda_{TP} \tag{18}$$

The reaction rate ρ is defined by lead-time Δl:

$$\rho = \frac{1}{\Delta l} \tag{19}$$

and repair rate μ is —as usual— the inverse of MTTR:

$$\mu = \frac{1}{MTTR} \tag{20}$$

C. Computing availability

In order to simplify representation we use numbers 0 to 6 to identify the states of the CTMC (as indicated in Fig. 4).

Steady state availability is defined as the portion of uptime versus lifetime, which is equivalent to the portion of time, the system is up. In terms of our CTMC model, this quantity can be determined by the equilibrium state distribution.

The infinitesimal generator Matrix Q of the CTMC shown in Fig. 4 is:

$$Q = \begin{pmatrix} -\lambda_p & \lambda_{TP} & \lambda_{FP} & \lambda_{TN} & \lambda_{FN} & 0 & 0 \\ (1-P_{TP})\rho & -\rho & 0 & 0 & 0 & P_{TP}\rho & 0 \\ (1-P_{FP})\rho & 0 & -\rho & 0 & 0 & P_{FP}\rho & 0 \\ (1-P_{TN})\rho & 0 & 0 & -\rho & 0 & 0 & P_{TN}\rho \\ 0 & 0 & 0 & 0 & -\rho & 0 & \rho \\ k\,\mu & 0 & 0 & 0 & 0 & -k\,\mu & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & -\mu \end{pmatrix} \tag{21}$$

The equilibrium distribution of a CTMC defines a probability distribution over the states, such that the global balance equations are fulfilled. This is equivalent to a solution to the following equations [12]:

$$\vec{\pi}\, Q = \vec{0} \tag{22}$$

$$s.t. \quad \sum_{i=0}^{6} \pi_i = 1 \tag{23}$$

If $\vec{\pi}$ is a solution to (22) then each scaling of $\vec{\pi}$ is also a solution to (22) and hence, an infinite number of solutions exist, one of which fulfills (23). Therefore, we arbitrarily set $\pi_6 = 1$ and solve the inhomogeneous equation system $\vec{\pi}\,'\,\hat{Q} = \vec{b}$ by Gaussian elimination yielding a single solution $\vec{\pi}\,'$ where $\hat{Q}$ is

$$\hat{Q} = \begin{pmatrix} -\lambda_p & \lambda_{TP} & \lambda_{FP} & \lambda_{TN} & \lambda_{FN} & 0 \\ (1-P_{TP})\rho & -\rho & 0 & 0 & 0 & P_{TP}\rho \\ (1-P_{FP})\rho & 0 & -\rho & 0 & 0 & P_{FP}\rho \\ (1-P_{TN})\rho & 0 & 0 & -\rho & 0 & 0 \\ 0 & 0 & 0 & 0 & -\rho & 0 \\ k\,\mu & 0 & 0 & 0 & 0 & -k\,\mu \end{pmatrix} \tag{24}$$

and

$$\vec{b} = \begin{pmatrix} -\mu & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{25}$$

The solution $\vec{\pi}$ that fulfills (23) is obtained by scaling of π_i':

$$\begin{aligned} \pi_i &= \frac{\pi_i'}{\left(\sum_{i=0}^{5} \pi_i'\right) + 1} \qquad i \in \{0 \ldots 5\} \\ \pi_6 &= \frac{1}{\left(\sum_{i=0}^{5} \pi_i'\right) + 1} \end{aligned} \tag{26}$$

Results are provided in Tab. III.

Steady-state availability is determined by the portion of time the stochastic process stays in one of the up-states 0 to 4 (see Fig. 4):

$$\begin{aligned} A &= \sum_{i=0}^{4} \pi_i = 1 - \pi_5 - \pi_6 \\ A &= \frac{(\rho + \lambda_p)\mu k}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})} \end{aligned} \tag{27}$$

yielding a closed-form solution for steady-state availability of systems with PFH.

TABLE III

SOLUTION TO THE STEADY STATE EQUATIONS FOR AVAILABILITY.

π_i	Solution
π_0	$\frac{\mu k \rho}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_1	$\frac{\mu k \lambda_{TP}}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_2	$\frac{\mu k \lambda_{FP}}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_3	$\frac{\mu k \lambda_{TN}}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_4	$\frac{\mu k \lambda_{FN}}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_5	$\frac{\rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP})}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$
π_6	$\frac{k\rho(P_{TN}\lambda_{TN} + \lambda_{FN})}{\mu k(\rho + \lambda_p) + \rho(P_{FP}\lambda_{FP} + P_{TP}\lambda_{TP} + kP_{TN}\lambda_{TN} + k\lambda_{FN})}$

V. MODELING RELIABILITY

Reliability $R(t)$ is defined as the probability of failure occurrence up to time t given that the system is fully operational at $t = 0$. In terms of CTMC modeling this is equivalent to a non-repairable system and computation of the first passage time into the down-state.

A. The Model

Since we are modeling a non-repairable system, the distinction between two down-states (F_p and $F_{\bar{p}}$) is not required anymore. Furthermore, there's no transition back to the up-state. That is why we use a simpler topology where the failure states F_p and $F_{\bar{p}}$ are merged into one absorbing state F as shown in Fig. 6.

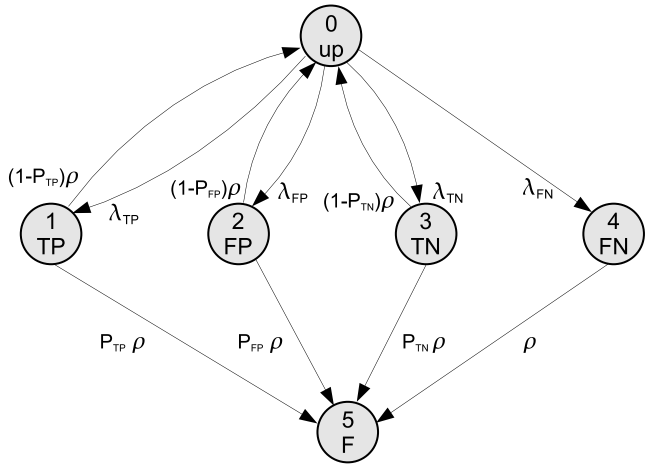


Fig. 6. The CTMC to model reliability. Failure states 5 and 6 of Fig. 4 have been merged into one absorbing state.

The Q-Matrix for this model has the form:

$$Q = \begin{pmatrix} T & t_0 \\ 0 & 0 \end{pmatrix} \tag{28}$$

where T is:

$$T = \begin{pmatrix} -\lambda_p & \lambda_{TP} & \lambda_{FP} & \lambda_{TN} & \lambda_{FN} \\ (1-P_{TP})\,\rho & -\rho & 0 & 0 & 0 \\ (1-P_{FP})\,\rho & 0 & -\rho & 0 & 0 \\ (1-P_{TN})\,\rho & 0 & 0 & -\rho & 0 \\ 0 & 0 & 0 & 0 & -\rho \end{pmatrix} \tag{29}$$

and t_0 equals:

$$t_0 = [\,0 \quad P_{TP}\,\rho \quad P_{FP}\,\rho \quad P_{TN}\,\rho \quad \rho\,]^T \tag{30}$$

B. Reliability and hazard rate

$R(t)$ and hazard rate $h(t)$ can be computed from the probability of moving into the down-state F, which is the probability distribution of TTF. In terms of CTMCs this quantity is called first passage time distribution $F(t)$ and respectively $f(t)$. Reliability and hazard rate can be computed from it in the following way:

$$R(t) = 1 - F(t) \tag{31}$$

$$h(t) = \frac{f(t)}{1 - F(t)} \tag{32}$$

$F(t)$ and $f(t)$ are the cumulative distribution and density of a phase-type exponential distribution [12] and we have:

$$F(t) = 1 - \alpha \exp(t\,T)\,e \tag{33}$$

$$f(t) = \alpha \exp(t\,T)\;t_0 \tag{34}$$

where e is a vector with all ones and $\exp(t\,T)$ denotes the matrix exponential. A closed form expression for the matrix exponential exists and can be computed using a symbolic computer algebra tool. However, the solution would fill several pages[1] and will hence not be provided here. α is the initial state probability distribution. It can be determined from the fact that reliability is defined such that the system is fully operational at time $t = 0$. Hence:

$$\alpha = [1 \quad 0 \quad 0 \quad 0 \quad 0] \tag{35}$$

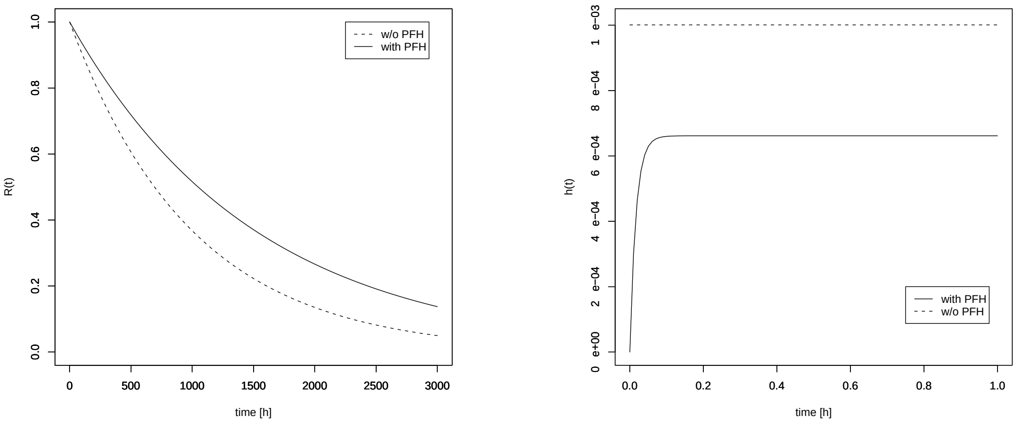
VI. An example

In order to give an idea about the effects of PFH approach on steady-state availability and reliability, we provide an example. Due to the lack of experimental data, the values have been chosen arbitrarily except for the failure predictor's parameters p, r and f, which are taken from experiments with a commercial telecommunication platform (see [13]). The parameter values that we have used are $MTTF = 999$h, $MTTR = 1$h, lead-time $\Delta l = 1$min, $p = 0.83$, $r = 0.9$, $f = 0.01$, $P_{TP} = 0.4$, $P_{FP} = 0.1$, $P_{TN} = 0.01$, $k = 2$. The system without PFH has unavailability 0.001 (i.e., availability 0.999). Employing PFH with the given parameters would reduce steady-state unavailability by a factor of about two to 0.000472 (i.e., availability 0.999528). Reliability is also improved (see Fig. 7-a) and the hazard rate is constantly below the hazard rate of a system without PFH. Especially, during the first half an hour a notable improvement is observed, as can be seen from Fig. 7-b.

VII. Estimating the Parameters from Experiments

The previous sections described how availability and reliability for systems with PFH can be determined as a function of seven parameters: p, r, f, P_{TP}, P_{FP}, P_{TN} and k. As it seems impossible to derive the parameters analytically from system specification, they must be estimated from experiments. The dilemma is that an assessment of availability and reliability is of most interest during system design when experiments cannot be carried out. Therefore, the values must be estimated from experiments in similar environments. This is possible since the estimation procedure separates the mutual influence of failure prediction and reaction schemes.

[1] The solution found by Maple$^{\text{TM}}$ contains approximately 3000 terms.



(a) Reliability

(b) Hazard rate

Fig. 7. Reliability and hazard rate for $MTTF = 999$h, $MTTR = 1$h, lead-time $\Delta l = 1$min, $p = 0.83$, $r = 0.9$, $f = 0.01$, $P_{TP} = 0.4$, $P_{FP} = 0.1$, $P_{TN} = 0.01$, $k = 2$.

A. Experiment I: Failure prediction accuracy

During the first experiment, only those parameters characterizing failure prediction (namely p, r, and f) are investigated without any feedback onto the system. This can either be accomplished by performing predictions offline working with previously recorded logfiles or performing them on a separate machine. Side effects such as additional workload caused by prediction are incorporated into the measures assessed in the second and third experiment.

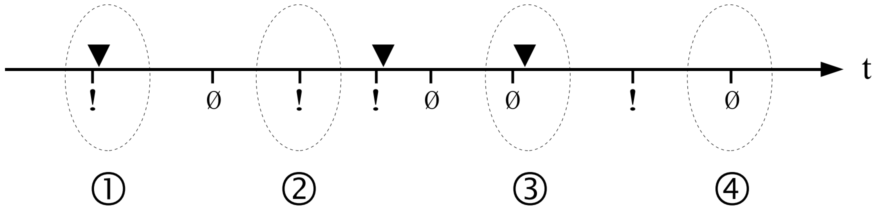


Fig. 8. A timeline obtained from an experiment showing true failures (▼) and prediction results. "!" indicate positive predictions (warnings) and "Ø" negative predictions. Four cases are highlighted that need to be counted for parameter estimation.

Starting from a timeline such as Fig. 8, predictions can be assigned to be TP (case ①), FP (case ②), FN (case ③) or TN (case ④). From this assignment a table like Tab. I can be set up and p, r, and f can be computed as defined in (1) to (3).

B. Experiment II: Failure probability P_{TP}

The goal of the second experiment is to assess the capability of the preventive measures to avoid an upcoming failure, which is represented by the probability P_{TP}. Once again, P_{TP} is the probability that a failure occurs given an upcoming true failure and a positive prediction. To estimate it, the second experiment has to be carried out with failure predictions and actions have to be performed on a test system that mimics key features of the modeled system as close as possible. The outcome of the experiment is again a timeline as in Fig. 8. However, the simple assignment of cases to TP, FP, etc. is not possible any more due to the following observations:

case ① can either refer to a TP prediction where the triggered action could not prevent the occurrence of the failure, or it can refer to a FP prediction successively leading to a failure induced by, e.g., the additional load caused by the prediction algorithm and the action that was triggered.

case ② can either refer to a FP prediction or to a TP prediction where the upcoming failure had been prevented.

case ③ can either refer to a FN prediction or to a TN prediction where the additional load caused a failure.

The conclusion of this is that it has to be clear which positive prediction is TP and which is FP. To solve this problem, the experiment of the second experiment must include fault injection to identify when a true failure is imminent in the system.

P_{TP} can then be estimated by analyzing only those time intervals where a failure was known to be imminent:

$$P_{TP} = \frac{count(\text{warnings with subsequent true failure in second experiment})}{count(\text{warnings in second experiment})} \tag{36}$$

where $count(\cdot)$ denotes the number of occurrences in the test result.

C. Experiment III: Failure probabilities P_{FP} and P_{TN}

P_{FP} and P_{TN} assess the risk of additional failures that are caused by PFH. In case of P_{TN} the failure is caused by predictions only, while in case of P_{FP} it is caused by the prediction and subsequent actions. Both probabilities can be estimated from a third experiment with a system having failure prediction and actions installed, but no fault injection (as in the second experiment) is needed.

In order to estimate P_{FP} we start with case ①, which reflects positive predictions followed by the occurrence of a failure (see Fig. 8). This situation can occur along with TP or FP predictions where in case of a TP prediction a failure occurs with probability P_{TP} and in case of a FP prediction with probability P_{FP} (see Fig. 6). Therefore, the following equation holds:

$$count(\text{case ①}) = P_{TP} \cdot count(\text{TP}) + P_{FP} \cdot count(\text{FP}) \tag{37}$$

In order to compute $count(\text{TP})$ and $count(\text{FP})$, we count the number of positive predictions (warnings) in the experiment and use the values for p, r and f that have already been estimated in the first experiment of the estimation procedure.

$$count(\text{TP}) = p \cdot count(\text{warnings in third experiment}) \tag{38}$$

$$count(\text{FP}) = (1-p) \cdot count(\text{warnings in third experiment}) \tag{39}$$

Since P_{TP} is known from the second experiment of the estimation procedure, the solution for P_{FP} is:

$$P_{FP} = \frac{count(\text{case ①}) - P_{TP} \cdot p \cdot count(\text{warnings in third experiment})}{(1-p) \cdot count(\text{warnings in third experiment})} \tag{40}$$

P_{TN} can be estimated in a similar manner. Equivalent to (37) the following equation holds:

$$count(\text{case ④}) = (1 - P_{TP}) \cdot count(\text{TN}) \tag{41}$$

and hence we obtain

$$P_{TN} = 1 - \frac{count(\text{case ④})}{count(\text{TN})} \tag{42}$$

where the number of TN predictions is computed by

$$count(\text{TN}) = count(\text{predictions in third experiment}) \cdot \frac{n_{TN}}{n} \tag{43}$$

where $\frac{n_{TN}}{n}$ is the fraction of TN predictions. Following (14), this fraction can be computed from p, r and f, which are known from the first experiment of the experiment.

D. Experiment IV: Repair time improvement

In order to estimate the repair time improvement factor k, an experimental trace such as Fig. 8 that additionally includes TTR is needed. As k is the ratio of MTTR without preparation and MTTR with preparation, mean values for both cases need to computed. Occurrences of case ① contribute to MTTR with preparation and occurrences of case ③ to MTTR without preparation, respectively.

VIII. Conclusions

With proliferation of failure prediction methods and proactive recovery techniques, there is a need to capture them in dependability models in order to assess their impact. The model presented here explicitly incorporates failure prediction including correct and false decisions as well as it accounts for preventive actions and improved repair at the same time. Our model is based on parameters where

- precision, recall and false positive rate are used for assessment of failure prediction accuracy
- probability of failure occurrence in case of true positive, false positive or true negative predictions are used to assess success of preventive actions as well as the occurrence of additional failures that are caused by proactive fault handling
- a repair time improvement factor accounts for the effect of preparing repair actions for an upcoming failure.

We have used continuous-time Markov chains for modeling and have derived a closed-form solution for steady state availability, reliability, and hazard rate, and we have proposed a procedure for estimating the selected parameters from experiments.

References

[1] P. Horn, “Autonomic computing: IBM’s perspective on the state of information technology,” New Orchard Road, Armonk, NY 10504, Oct. 2001. [Online]. Available: http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[2] C. Mundie, P. de Vries, P. Haynes, and M. Corwine, “Trustworthy computing,” Microsoft Corp., Tech. Rep., Oct. 2002. [Online]. Available: http://www.microsoft.com/mscorp/twc/twc_whitepaper.mspx

[3] A. Brown and D. A. Patterson, “To err is human,” in *Proceedings of the First Workshop on Evaluating and Architecting System dependabilitY (EASY ’01)*, Göteborg, Sweden, Jul. 2001.

[4] S. Garg, A. Puliafito, M. Telek, and K. Trivedi, “Analysis of preventive maintenance in transactions based software systems,” *IEEE Trans. Comput.*, vol. 47, no. 1, pp. 96–107, 1998.

[5] O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, van Moorsel A., and M. van Steen, Eds., *Self-Star Properties in Complex Information Systems*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2005, vol. 3460.

[6] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems*, 2nd ed. Bedford, MA: Digital Press, 1992.

[7] Y. Huang, C. Kintala, N. Kolettis, and N. Fulton, “Software rejuvenation: Analysis, module and applications,” in *Proceedings of IEEE Intl. Symposium on Fault Tolerant Computing, FTCS 25*, 1995.

[8] V. Castelli, R. Harper, H. P., S. Hunter, K. Trivedi, K. Vaidyanathan, and W. Zeggert, “Proactive management of software aging,” *IBM Journal of Research and Development*, vol. 45, no. 2, pp. 311–332, Mar. 2001.

[9] C. Van Rijsbergen, *Information Retrieval*, 2nd ed. London: Butterworth, 1979.

[10] I. Gertsbakh, *Reliability Theory: with Applications to Preventive Maintenance*. Berlin, Germany: Springer-Verlag, 2000.

[11] Y. Bao, X. Sun, and K. Trivedi, “Adaptive software rejuvenation: Degradation model and rejuvenation scheme,” in *Proceedings of the 2003 International Conference on Dependable Systems and Networks (DSN’2003)*. IEEE Computer Society, 2003.

[12] V. G. Kulkarni, *Modeling and Analysis of Stochastic Systems*, 1st ed. London, UK: Chapman and Hall, 1995.

[13] F. Salfner and M. Malek, “Predicting failures of computer systems: A case study for a telecommunication system,” in *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006), DPDNS workshop*, Rhodes Island, Greece, Apr. 2006.