A framework for dependable adaptation in probabilistic environments

António Casimiro
Paolo Lollini
Mônica Dixit
Andrea Bondavalli
Paulo Veríssimo

DI–FCUL TR–07–24

October 2007

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749–016 Lisboa
Portugal

A framework for dependable adaptation in probabilistic environments

A. Casimiro
casim@di.fc.ul.pt
Univ. of Lisboa*

P. Lollini
lollini@dsi.unifi.it
Univ. of Firenze†

M. Dixit
mdixit@lasige.di.fc.ul.pt
Univ. of Lisboa

A. Bondavalli
bondavalli@unifi.it
Univ. of Firenze

Paulo Veríssimo
pjv@di.fc.ul.pt
Univ. of Lisboa

October 2007

Abstract

Distributed applications executing in probabilistic environments, like the Internet, often need to make timing assumptions, for instance about the maximum message delay or the round-trip delay. In the case of adaptive systems these bounds should be computed at run-time, using probabilistic or other more or less ad hoc approaches, typically with the objective of improving the application performance. From a dependability perspective, however, the concern is to secure some properties on which the application can rely. Building on previous work where we set the stage for dependable adaptation, we introduce a general framework for implementing improved solutions for dependable adaptation. During its lifetime, a system alternates periods where its temporal behavior is well characterized, with transition periods where a variation of the environment conditions occurs. The proposed framework rests on the detection of these stable and transient phases, and it is generic in the sense that different detection mechanisms can be included. We use a set of simple phase detection mechanisms inspired in well-known probabilistic tests, along with synthetically generated data flows, to illustrate how the framework can be applied.

1 Introduction

Computer systems and applications are becoming increasingly distributed and we assist to the pervasiveness and ubiquity of computing devices. This openness and complexity means that the environment (including network and computational platforms) tends to be unpredictable, essentially asynchronous, making it impractical, or even incorrect, to assume any time-related bounds. On the other hand, there are increased concerns about the dependability of these systems and applications, in the sense of their ability to meet some specified quality of service (QoS) levels. One possible way to cope with the uncertain timeliness of the environment while meeting dependability constraints consists in building adaptive applications and ensure that they adapt in a dependable way, that is, they remain correct as a result of adaptation.

*Faculdade de Ciências da Universidade de Lisboa. Bloco C6, Campo Grande, 1749-016 Lisboa, Portugal. Navigators Home Page: http://www.navigators.di.fc.ul.pt. This work was partially supported by the EC, through project IST-FP6-STREP-26979 (HIDENETS), and by the FCT, through the Large-Scale Informatic Systems Laboratory (LaSIGE).

†Università degli Studi di Firenze - Dip. Sistemi e Informatica. Viale Morgagni 65, I-50134 Firenze, Italy. Resilient Computing Lab Home Page: http://rcl.dsi.unifi.it/.

In this paper we build on previous work that introduced the necessary architectural and functional principles over which a dependable adaptation approach should be based [1]. In essence, the idea behind dependable adaptation is to ensure that a *coverage stability* property is satisfied, that is, that the assumed bounds for fundamental variables (e.g. network delay) are secured with a known and constant probability. Clearly, this is only possible if some assumptions about the environment behavior are made. Then, using appropriate mechanisms and approaches it may be possible to probabilistically characterize the current operational state and derive the bounds that must be used to secure the coverage stability property.

Therefore, a crucial issue concerns the set of assumptions that are made about how the environment behaves. In our earlier work we made very weak assumptions— just that it would behave probabilistically, but with unknown distributions. This was sufficient to illustrate the feasibility of the dependable QoS adaptation approach. Now we take a step further, assuming that during its lifetime a system alternates periods where its probabilistic behavior is well characterized, with transition periods where a variation of the environment conditions occurs.

Because of that, we are able to shift from conservative (and pessimistic) solutions, with possibly little relevance in practice, to more reasonable ones, which may have interesting practical reach. We introduce a framework that is based on the use of probabilistic methods for the recognition of the "state" of the environment. The framework does not require a specific method to be used. On the contrary, it allows several methods to be used simultaneously. Given that the actual methods used in some implementation are not our main concern, we use some simple ones just to illustrate how the framework can be applied, how it may improve over the previous conservative approach, and what kind of trade-offs must be addressed when choosing a specific method to be applied. Nevertheless, the paper explains how the chosen methods were implemented, which is relevant to show how dependability constraints are handled in practice.

In fact, one important and distinguishing aspect of our work is that we are concerned with dependability objectives. These are incorporated in the framework by establishing the criteria that should be met in order to evaluate the effectiveness of specific probabilistic methods driving the adaptation process. In particular, we used this criteria to compare the above-mentioned simple methods, by performing a number of simulation experiments based on synthetic data flows generated from well-known probabilistic distributions. Based on these results we were able to conclude that the proposed framework may indeed allow to achieve dependable adaptation and improved time bounds, provided that adequate environment recognition methods are used for a given environment behavior.

The paper is organized as follows. In the next section we provide a motivation for this work and we discuss related work. Then, Section 3 describes the proposed framework for dependable adaptation in probabilistic environments. Implementation issues are then addressed in Section 4, while the simulation results are presented and discussed in Section 5. Some conclusions and future perspectives are finally presented in Section 6.

2 Motivation and Related Work

Providing QoS guarantees for the communication in spite of the uncertain or probabilistic nature of networks is a problem with a wide scope, which can be addressed from many different perspectives. We are fundamentally concerned with timeliness issues and with securing or improving the dependability of adaptive applications. In fact, our work is being carried out in the scope of the HIDENETS EU-FP6 project [6], which aims at devising improved end-to-end solutions for highly dependable IP-based networks, focusing on car-2-car applications and communication over ad hoc network infrastructures as motivating scenarios. Our contribution, however, is more general: we merely reason in terms of probabilistic environments, which can be, but are not constrained to, the environments considered in HIDENETS.

In [1], the fundamental architectural and functional principles for dependable QoS adaptation were introduced, providing some motivation for the work presented here. In this earlier work we also followed a dependable perspective, and analyzed why systems would fail as a result of timing assumptions be violated, as it may happen in asynchronous environments. A relevant effect is *decreased coverage* of some time bound [12], when the number of timing failures goes beyond an assumed limit. This effect can be handled in adaptive systems, by adapting the assumed bounds during the execution to ensure that coverage of that bound is secured. In other words, the objective is to satisfy the so called *Coverage Stability* property.

In more practical terms, this means that QoS is no longer expressed as a single value, a time bound to be satisfied, but as a $\langle bound, coverage \rangle$ pair, in which the coverage should remain constant while the bound may vary as a result of adaptation, to meet the conditions of the environment. On the other hand, deciding when and how to adapt depends on what is assumed about the environment. In [1] we followed a conservative approach, just assuming a probabilistic environment but not a specific probabilistic distribution for delays. Therefore, this led to a conservative solution with respect to the bounds required to guarantee some coverage.

Interestingly, in the last few years a number of works have addressed the problem of probabilistically characterizing the delays in IP-based networks based on real measured data, allowing to conclude that observed empirical delay distributions may be characterized by well known distributions, such as the Weibull distribution [9, 5], the shifted gamma distribution [10, 3], the exponential distribution [7] or the truncated normal distribution [4]. Based on this, we realized that it would be interesting and appropriate to consider less conservative approaches, by assuming that specific distributions may be identified and thus allowing to achieve better timing bounds for the same required coverage.

However, some of these works also recognize that probabilistic distributions may change over time (e.g. [10]), depending on the load or other sporadic occurrences, like failures or route changes. Therefore, in order to secure the required dependability attributes, it becomes necessary to detect changes in the distribution and hence use mechanisms for being able to do that. Fortunately, there is also considerable work addressing this problem and well-known approaches and mechanisms (see [13] for a nice overview). Among others, we can find approaches based on time-exponentially weighted moving histograms [8], on the Kolmogorov-Smirnov test [4] or the Mann-Kendall test. For instance, in [4] the authors apply the Kolmogorov-Smirnov test on round-trip time (RTT) data to detect state changes in a delay process. Between these state changes, the process can be assumed to be stationary with constant delay distribution. The stationary assumption was confirmed by the trend analysis test.

Given all these possibilities, we understood that the design of a framework for dependable adaptation could be made general by allowing the use of different approaches, even simultaneously, to allow the achievement of possibly better or more accurate predictions.

A fundamental distinguishing factor of our work is that we are concerned with system dependability. Therefore, while other works addressing adaptive systems are mainly concerned with performance improvements, our objective is to achieve dependable adaptive designs by ensuring coverage stability.

3 The Adaptation Framework

3.1 Assumptions

As mentioned in the previous section, in this work we advance on previous results by making more optimistic but not less realistic assumptions, in order to achieve improved and still dependable $\langle bound, coverage \rangle$ pairs. Instead of simply assuming that the environment behaves in a probabilistic way (as done in [1]) we now make the following assumptions:

Interleaved known probabilistic behavior - We assume that the environment alternates stability periods, during which it follows some known probabilistic behavior, with unstable periods, during which the probabilistic behavior is unknown or cannot be characterized. Note that, despite the unknown probabilistic behavior during unstable periods, we assume that the behavior is probabilistic.

Recognition abilities - The system has sufficient resources and can be equipped with the adequate mechanisms to detect changes from stable to unstable periods (and vice-versa) and recognize the best fitting probabilistic distribution during stable periods.

Sufficient stability - We assume that probabilistic changes are slower than the detection and recognition speed. In other words, we assume that typical execution environments are sufficiently stable for the time it takes to recognize the actual probabilistic distribution. If this turns out to be false, then, as we will see, only the improvements will be lost, but dependability will be kept.

Sufficient activity - We assume that there is sufficient system activity, and therefore enough observation points that we can use to feed some trend detection and recognition mechanisms. Hence it is possible to use observation samples, that is, measured delays, assuming that all of them report to a sufficiently recent observation period as required for accurately characterizing the state of the environment. We therefore discard any further concerns related to the time spacing of observations.

We must note that these assumptions may not hold in every environment and every situation. In fact, although many authors have observed the possibility of probabilistically characterizing end-to-end delays, it is also true that there is not a unique or a well defined distribution that can be used, which intuitively means that in some environments it may be difficult to determine, in practice, a given distribution that fits the observed delays. However, our main concern in this paper is to propose a framework to address the problem, allowing different mechanisms and solutions to be used together in order to better deal with these uncertainties. We propose some of the possible mechanism to highlight how the framework can be applied.

3.2 Environment recognition and adaptation

The proposed framework for dependable (QoS) adaptation can be seen as composed by two activities: identification of the current environment conditions and consequently QoS adaptation.

Environment recognition - The environment conditions can be inferred analyzing a real time data flow representing, for example, the end-to-end message delays in a network. When the analytical description of the data is not known, we need to determine the model that best describes the data. Using statistics the data may be described by a probability density function (pdf) or by a cumulative distribution function (CDF). We note that the data models can be so complex that they can not be described in terms of simple pdf or CDF; for example, the sequence of data could be properly represented by a complex mixed distribution. In the case in which the model that describes the data is known, the problem is reduced to estimating unknown parameters of a known model from the available data. The known model is based on the knowledge of the process that produced the data.

There are various methods, both numerical and graphical, for estimating the parameters of a probability distribution (see, e.g., [11]). From a statistical point of view, the method of maximum likelihood estimation (MLE) is considered to be one of the most robust techniques for parameter estimation, and it is used in our experimental implementation of the framework.

Adaptation - Once the most fitting distribution (together with its parameters) has been identified, its statistics properties can be exploited to find the pair $\langle bound_{spec}, coverage_{spec}\rangle$ that represents the coverage and the corresponding time bound specifically tailored for the selected distribution. With respect to the general pair $\langle bound, coverage\rangle$ defined in [1], the new pair provides better results, since the coverage stability objective can be reached using a lower time bound.

3.3 The conservative approach

As detailed in Section 3.1 we are assuming an interleaved probabilistic behavior of the environment. Therefore, we can consider that the system alternates periods during which the conditions of the environment remain fixed (**stable phases**), with periods during which the environment conditions change (**transient phases**). In the first case, the statistical process that generates the data flow (e.g., the end-to-end message delays) is under control and then we can compute the corresponding distribution using an appropriate number of samples; the higher h is (where h is the sample size), the higher is the accuracy of the estimated distribution. On the contrary, if the environment conditions are changing, then the associated statistical process is actually varying, so no fixed distribution can describe its real behavior.

Therefore, the system lifetime can be seen as an alternate sequence of phases: a stable phase, during which the distribution is computable, and a transient phase, during which the distribution can not be computed. A transient phase corresponds to the period during which the original distribution is changing and is moving towards the distribution of the next stable phase.

As previously mentioned, during the transient phases it is not possible to build a pdf, so in this period we adopt a **conservative approach** and we set the pair $\langle bound, coverage \rangle$ using the one-sided inequality as in [1]. It is a pessimistic bound, but it holds for all the distributions. As soon as the presence of a stable phase is detected, a proper pdf is identified and the bound is modified according to the new distribution.

In order to do this, it appears evident that we need a mathematical method that verifies whether the system is in a stable phase or in a transient one. In other words, we need a sort of **phase detection mechanism** capable of identifying the beginning of a new transient phase as soon as the environment conditions start changing, and capable to identify the beginning of a new stable phase as soon as the environment conditions stabilize.

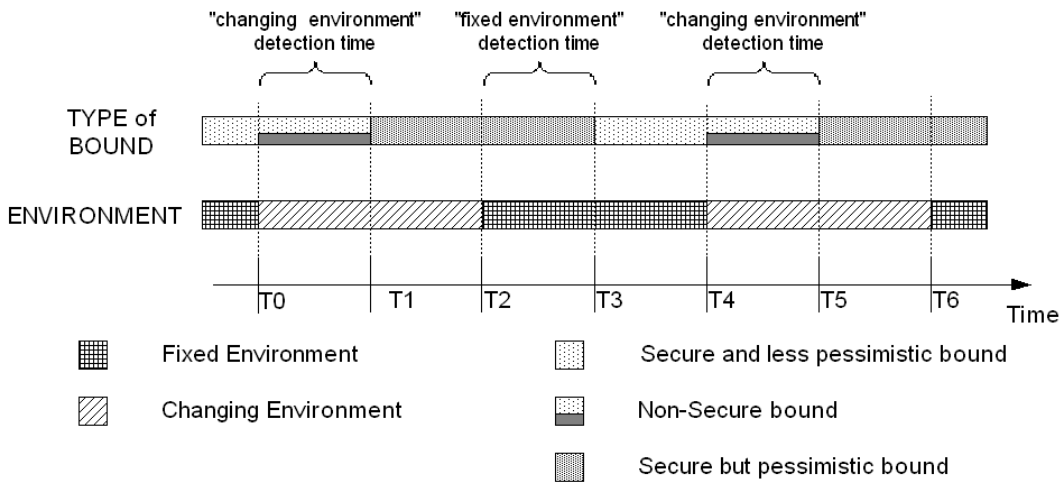

Figure 1: The conservative approach.

Figure 1 depicts the conservative approach applied to a typical scenario, such as the one identified by the following temporal events:

- Before time T0 the environment conditions are fixed.
- At time T1 the phase detection mechanism detects that the environment is changing (or that it has already changed if T1>T2), then the bound is set to the secure but pessimistic level as in [1].
- At time T2 the environment reaches a new stable configuration.

- At time T3 the phase detection mechanism detects the presence of a stable phase and then a new less pessimistic bound (tailored for the identified distribution) can be computed.
- At time T4 the environment conditions change again.
- At time T5 the phase detection mechanism detects that the environment is changing (or that it has already changed if T5>T6), then the bound is set to the secure but pessimistic level as in [1].
- At time T6 the environment reaches a new stable configuration, and so on.

We note that there is an alternation of periods during which the bound is less pessimistic and secure (e.g. [T3;T4]), non secure (e.g. [T4;T5]) and secure but pessimistic ([T5;T6]). The effectiveness of this conservative approach mainly depends on two factors:

- the "changing environment" detection time, that is the time that it takes to detect that the environment is changing.
- the "stable environment" detection time, that is the time that it takes to detect that the environment has reached a new stable configuration.

The "changing environment" detection time is of particular importance in this context, since it directly affects the quality (or the accuracy) of the adaptation mechanism. During such critical periods, the environment is changing but the bound is tailored for a particular set of condition environments that do not hold anymore. The ideal situation arises when the detection of a changing environment is instantaneous (e.g., T0=T1).

3.4 Phase detection mechanisms

A phase detection mechanism must detect as soon as possible the beginning of a new transient phase, as well as the beginning of a new stable phase. Although marginal to the main focus of the paper, here we propose some simple phase detection mechanisms that can be implemented in the framework. Such mechanisms, that we used in the experimental implementation of the framework to illustrate its feasibility, can be used individually, or combined together. For example, one can be more conservative by configuring the framework so that a stable phase is detected only if all the mechanisms are actually detecting a stable phase at the same time, while in a less conservative approach a stable phase is detected if at least one of the mechanisms actually detects a stable phase.

- **Mechanism 1**. A stable period with distribution $\hat{D}$ is detected when the estimated mean $E(\hat{D})$ and the estimated variance $V(\hat{D})$ satisfy some relations that are always true for the postulated distribution $\hat{D}$. For example, if $E(\hat{D}) = \sqrt{V(\hat{D})}$ (or, more practically, if $|\sqrt{V(\hat{D})} - E(\hat{D})| < \tau$, being τ a tolerance bound) we could hypothesize that the samples come from an exponential distribution. Actually this equality is not a sufficient condition for having an exponential distribution (it is only a necessary condition), so the mechanism could lead to the identification of a wrong distribution not resembling the real environment conditions.
- **Mechanism 2**. At time t_1 we compute the expected mean $E(\hat{D})_{t_1}$ and observed variance $V(\hat{D})_{t_1}$ using the last h collected sampling points. A stable phase will be detected if the following f measurements $E(\hat{D})_{t_i}$ and $V(\hat{D})t_i$, with $i = 2, \ldots, f+1$ (each one performed using the available last h collected sampling points), will remain within a given tolerance level e_{dev} and v_{dev} from $E(\hat{D})_{t_1}$ and $V(\hat{D})_{t_1}$, respectively. In other words:

$$\begin{aligned} |E(\hat{D})_{t_i} - E(\hat{D})_{t_1}| &\leq e_{dev} \quad \text{with } i = 2, \ldots, f+1, \text{ and} \\ |V(\hat{D})_{t_i} - V(\hat{D})_{t_1}| &\leq v_{dev} \quad \text{with } i = 2, \ldots, f+1 \quad . \end{aligned}$$

The same mechanism could be extended to analyze, in general, the k-th moment of the random variable D.

- **Mechanism 3**. A stable period with distribution $\hat{D}$ is detected when some goodness-of-fit (GoF) tests establish the goodness of fit between the postulated distribution $\hat{D}$ and the evidence contained in the experimental observations. For example, the Kolmogorov-Smirnov test [2] is the preferred GoF test in case of a continuous population distribution.

In the following section we further detail on how these mechanisms have been implemented in the context of the proposed framework. This also serves to clarify the conditions under which the experimental results presented and discussed in Section 5 were obtained.

4 Implementation Issues

We use the scheme depicted in Figure 2 to help us explaining some important issues concerned with the implementation of this framework for dependable adaptation.

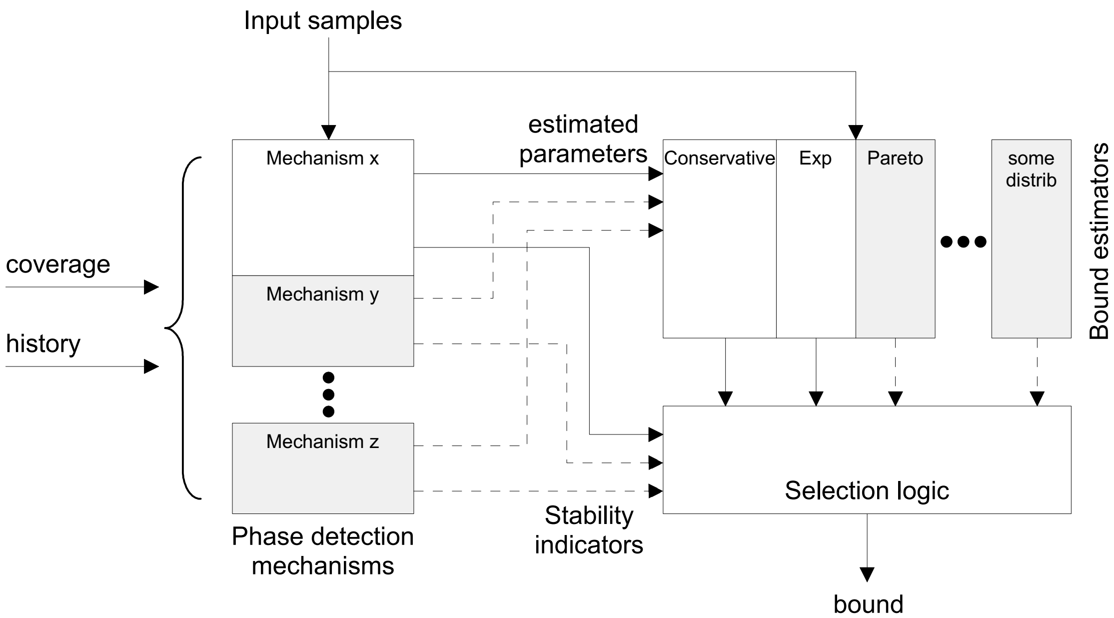

Figure 2: Schematic view of the framework for dependable adaptation.

Looking at the figure, we see that the framework can be modelled as a service that:

- accepts two dependability related parameters as input: the history size and the required coverage;
- reads samples/measured delays as input: these are the actual values that fill up the history and are used by the phase detection mechanisms and for the estimation of distribution parameters;
- provides, as output, a bound that should be used in order to achieve the specified coverage.

In this experimental implementation we only considered two probabilistic distributions: the **exponential** and the **Pareto** distributions. This was an early decision that we took based on the fact that these seemed to be relatively easy distributions to work with, and would perfectly serve our purpose of highlighting the way in which the framework should be used. In practical systems, however, one is supposed to consider several other possible probabilistic distributions, as means to better find a good fit for the actual environment behavior. On the other hand, depending on the considered probabilistic distributions, it is possible that the concrete configuration of phase detection mechanisms varies. In our case, we have configured the phase detection mechanisms for working correctly and secure the desired coverage for these two distributions only. The actual number of different mechanisms that could be used and configured for different distributions, is only limited by the available resources, which we assume to be enough, as per the "Recognition abilities" assumption (see Section 3.1).

4.1 Setting up parameters

The way in which the coverage and history parameters are handled or incorporated in the phase detection mechanisms can assume two basic forms: either it is possible to derive analytical expressions that "automatically" relate the outcome of phase detection with these input parameters, or it is necessary to stipulate some values (like e_{dev} and v_{dev} when using "Mechanism 2" described in Section 3.4) as necessary to secure the desired coverage.

As previously mentioned, it is possible to use several phase detection mechanisms simultaneously. We have implemented the three mechanisms described earlier, which were instantiated to detect the exponential distribution. For the detection of the Pareto distribution we only configured one mechanism, an instantiation of "Mechanism 1", as explained ahead.

In order to instantiate "Mechanism 1" to recognize the exponential probabilistic distribution we used the following reasoning.

"Mechanism 1" estimates the expected value $E(\hat{D})$ and the variance $V(\hat{D})$, and it detects a stable exponential period if $|\sqrt{V(\hat{D})} - E(\hat{D})| < \tau$, being τ a tolerance bound to encompass for the estimation errors. Clearly, the ability to detect the distribution depends on the size (h) of the sample history (taken from the trace) that is used to estimate $E(\hat{D})$ and $V(\hat{D})$, and on τ.

Let us try to find a relation that statistically links the real mean to the history size h and to the estimated mean. Let us indicate with $E(D)$ the real mean and with $V(D)$ the real variance (if they exist).

From statistic we know that if D is exponentially distributed, then a $100(1-\alpha)\%$ confidence interval of $E(D)$ is

$$\frac{2h}{\chi^2_{2h;\alpha/2}} E(\hat{D}) < E(D) < \frac{2h}{\chi^2_{2h;1-\alpha/2}} E(\hat{D}) \quad ,$$

where χ^2 is the critical value of chi-square distribution with α degrees of freedom. Let C be the coverage that must be guaranteed for an adaptive application. We know that in the case of exponential distribution $\hat{D}$ of the delays, the correspondent time bound t is

$$t = \frac{1}{E(\hat{D})} ln \frac{1}{1-C} \quad .$$

Now let's consider what happens if

- $E(\hat{D}) = E(D)$: we are in the optimal case, since the estimated mean corresponds to the real mean, so we maintain the coverage with the minimum bound $t_{estimated} = t_{real}$.
- $E(\hat{D}) > E(D)$: in this case we are still maintaining the coverage because $t_{estimated}$ is greater than t_{real}. Therefore we have no coverage problems but we are not in an optimal situation.
- $E(\hat{D}) < E(D)$: in this case $t_{estimated} < t_{real}$, therefore the coverage is no more maintained to the C value, since more timing failures will occur.

From these considerations it follows that, in order to maintain the coverage at a given level C, we have to find a value for $E(\hat{D})$ such that $E(\hat{D}) \geq E(D)$. Therefore, we can implement "Mechanism 1" such that it detects a stable phase when the square root of the estimated variance ($\sqrt{V(\hat{D})}$) is inside the confidence interval, and in this case we use the right (upper) bound of the confidence interval to set the time bound for the application, so as to maintain the required coverage. If $\sqrt{V(\hat{D})}$ is outside the confidence interval then it cannot be (probabilistically) equal to the exact mean and then the sampling points cannot be (probabilistically) generated by an exponential distribution, therefore a transient phase is detected. The error τ will be always lower than the size of the confidence interval.

If the confidence level (the percentage $100(1-\alpha)\%$) increases and/or the history size (h) decreases, "Mechanism 1" will more likely detect a stable phase since the confidence interval becomes larger. At the same time the probability to wrongly detect a stable exponential phase increases. Therefore we can use the confidence level and the history size to tune the level of "aggressiveness" of the mechanism.

The configuration of the other two mechanisms for the exponential distribution was done by exactly as explained in Section 3.4, which required some values to be postulated. In particular, for the case of "Mechanism 2", we concluded based on experimental evidence that for an history $h = 30$, we should use $e_{dev} = 0.5 \times E(D)_{t1}$ and $v_{dev} = 0.5 \times V(D)_{t1}$. With these settings it is possible to apply the mechanism in a dependable way. On the other hand, we postulated $f = 10$, making it smaller than h, but still large enough to be more aggressive than "Mechanism 1". For the case of "Mechanism 3", we just had to postulate the critical values of Kolmogorov-Smirnov statistics for the exponential distribution with unknown mean, which we did by using standard values provided in statistics tables.

With respect to the instantiation of "Mechanism 1" for the Pareto distribution, we had to postulate a value for τ, which in practice depends on the history h and on shape parameter of the distribution. For a Pareto distribution with shape parameter $\alpha = 4.2$ and $h = 30$, we used $\tau = 0.05$ (as per the values of Table 1, that we constructed based on several tests to empirically determine which values would ensure the dependability of the mechanism). The location parameter of the distribution is always set to the lower value observed in the history. A stable Pareto distribution is detected when $|V(\hat{D} - E(\hat{D}^2/\alpha(\alpha-2)| < \tau$.

α	$\tau(h = 30)$
(0,1.2]	1
(1.2,2.4]	0.4
(2.4,3.2]	0.1
(3.2,inf]	0.05

Table 1: Values for τ, Pareto distribution.

4.2 Defining bound estimators

Depending on the output of the phase detection mechanisms, which consists both in stability indicators and in estimated parameters that characterize a certain distribution (e.g. λ in the case of exponential), one of the bounds calculated by the implemented bound estimators will be selected as the output of the framework. The selection is performed with the help of some logic, which might be made more or less aggressive. For instance, in the case when distinct mechanisms detect a stable phase, it may be possible to choose between two bounds, if the phase detection mechanisms provide different distribution parameters to the bound estimators. When no mechanism detects a stable phase, then the conservative approach for estimating the bound is always used.

In this experimental implementation we defined three bound estimators: the conservative estimator, which is always required, and estimators for the exponential and for the Pareto distributions. They are presented in Table 2.

Estimator	Minimum time bound t
conservative	$bound = \frac{2E(D)+\sqrt{4E(D)^2-4(E(D)^2+V(D)-\frac{V(D)}{1-coverage})}}{2}$
exponential	$bound = \frac{1}{\lambda}\ln\frac{1}{1-coverage}$
Pareto	$bound = \frac{k}{\sqrt[\alpha]{1-coverage}}$

Table 2: Bound estimators for a given required coverage.

5 Results and Discussion

We did a few experiments to test the implemented framework and to observe, in particular, what would be the improvements achievable with a less conservative, and adaptive approach, in comparison with the simple conservative and pessimistic approach described in [1]. In fact, we did a few more tests to observe the effective behavior of the mechanisms (we focused on "Mechanism 1") in the presence of a distribution change, and with a Pareto distribution instead of an exponential distribution. The results are reported below.

The experiments were done using synthetic data traces following the exponential and Pareto distributions, represent observed delivery delays over a period of time (whose length is not relevant here). We used some common parameters in all experiments: traces with 1,500 sample points; expected coverage of 0.98 (i.e., to guarantee the coverage stability, no more than 30 timing failures should occur); and a history size of $h = 30$.

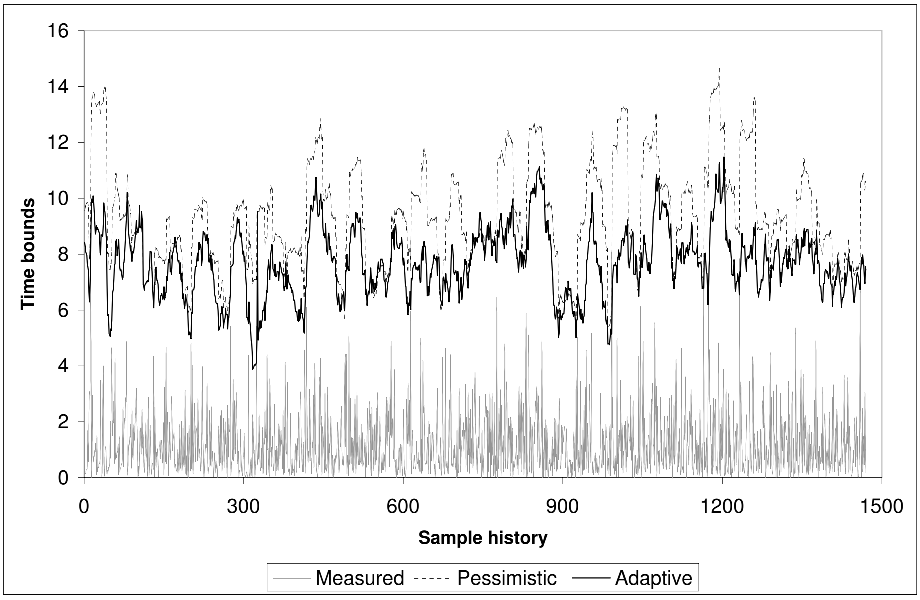

Figure 3: Comparison of pessimistic and adaptive approaches, using "Mechanism 1" for an exponential trace.

Figure 3 shows the result of the first experiment, which is based on a synthetic data trace that follows an exponential distribution with $\lambda = 0.8$. The purpose of this experiment is to validate "Mechanism 1" in detecting stable phases. This mechanism detected an exponential distribution in 97% of the sample points. The adaptive time bounds generated only five timing faults in the entire observation period, maintaining the coverage stability. Moreover, in the majority of the sample points, the adaptive bounds are more aggressive than the conservative bounds, as expected. This is supported by the lower time bounds average of the adaptive approach: 7.6 against 9.3 of the conservative method. However, there are few points where the adaptive bound is bigger than the conservative one. This rare situation happens when the adaptive bound is calculated using a bigger estimated $E(\hat{D})$ (both bounds are directly proportional to $E(\hat{D})$). While the conservative approach estimates $E(\hat{D})$ as the history average, the adaptive bound uses the greatest value in the confidence interval of $E(D)$.

We executed similar experiments using the same exponential data trace as input, to evaluate "Mechanism 2", "Mechanism 3" (which implements the Kolmogorov-Smirnov test), and a combination of "Mechanism 1" and "Mechanism 3". The results are presented in Table 3. They are expressed in terms of the following metrics:

- the number of observed faults (which must be lower than some value to secure that coverage is achieved);
- the average value of the time bounds that were used as output of the framework.

Mechanism	*Stable detection*	*Timing faults*	*Bounds average*
1	97%	5	7.6
2	96%	5	7.7
3	95%	4	7.8
1 + 3	75%	4	8.1

Table 3: Observed results for different mechanisms using the exponential trace.

The results of "Mechanism 2" demonstrate that its aggressiveness is quite similar to "Mechanism 1": the exponential distribution was detected in 96% of the sample points, the number of timing faults was also 5 and the adaptive bounds average was 7.7. After detecting a stable phase, "Mechanism 2" uses the same formulas as "Mechanism 1" to identify the probability distribution and calculate the adaptive bounds. Thus, this similarity was expected. We remind that the parameter values used on this experiment were $f = 10$, $e_{dev} = 0.5 \times E(\hat{D})_{t1}$ and $v_{dev} = 0.5 \times V(\hat{D})_{t1}$, i.e., for each sample point, a stable phase was detected if the last 10 measured $E(\hat{D})$ and $V(\hat{D})$ were at most 50% higher than current $E(\hat{D})$ and $V(\hat{D})$, respectively.

"Mechanism 3" also produced similar results. Exponential stable phases were detected in 95% of the sample points, 4 timing faults occurred, and an average of 7.8 was verified for the adaptive bounds. The level of significance for the Kolmogorov-Smirnov test, which indicates the probability of a stable phase being recognized as a transient one, was $\alpha = 0.05$.

As expected, a more conservative result is obtained when combining "Mechanism 1" and "Mechanism 3". Whenever "Mechanism 1" detects an exponential distribution, it executes the Kolmogorov-Smirnov test to confirm it. This double-checking implementation decreases the aggressiveness of the adaptive approach (exponential detection in 75% of the sample points), but makes it more dependable. Even being more conservative than individual mechanisms, this combination still remains with lower bounds in comparison to the pessimistic approach (adaptive bounds average of 8.1).

Analogously, Figure 4 shows the results of using "Mechanism 1", but now based on a synthetic data trace that follows a Pareto distribution with $\alpha = 4.2$ and $k = 0.65$. The Pareto distribution was detected in 75% of the sample points and the bounds average decreased from 2.5 to 1.9. Also, it was observed the occurrence of 28 timing faults. We have noted that the higher the k parameter of the input trace, the lower is the number of timing faults. For example, when using Pareto data traces with $\alpha = 4.2$ and $k = 1.5$, we verify approximately 15 timing faults.

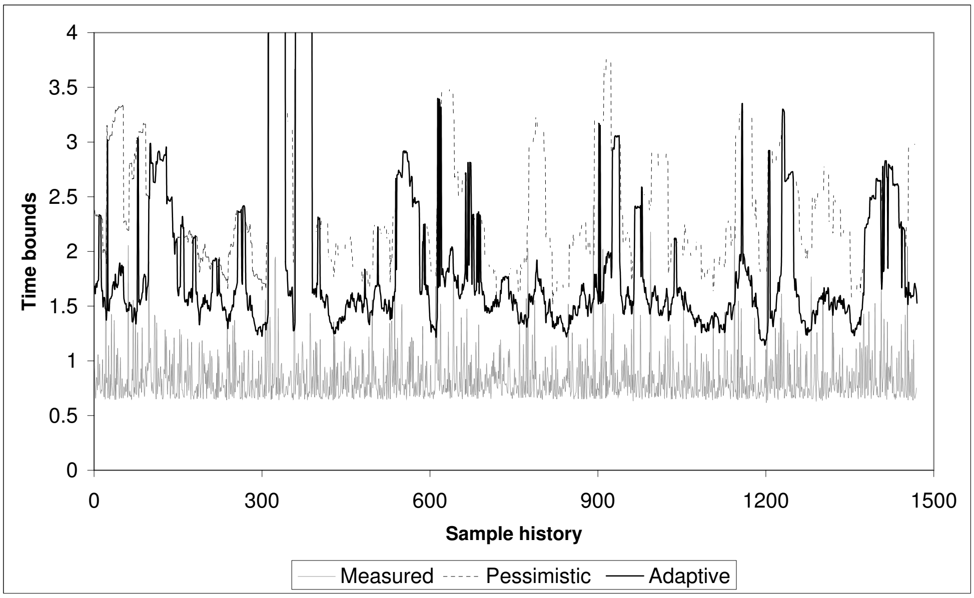

Figure 4: Comparison of pessimistic and adaptive approaches, using "Mechanism 1" for a Pareto trace.

We also executed some experiments using data traces that do not follow neither a exponential nor a Pareto distribution to verify if the mechanisms do not incorrectly detect stable phases. The results of one of these experiments is shown in Figure 5. This experiment uses a data trace generated by the *random* method of the Java class *Math*, which returns an approximated uniform distribution. When analyzing Figure 5 it is easy to note that in almost every sample point, the adaptive bound is equal to the conservative bound, which means that the pessimistic approach was used. Actually, "Mechanism 1" detected an exponential distribution in 10% of the sample points. Nevertheless, the coverage stability was not violated, since no timing faults occurred.

Another interesting aspect that can be observed using simulated experiments is the influence of the history size when detecting changes in the environment conditions. The experiment shown in Figure 6 used a trace where the first 750 sample points follow an exponential distribution with $\lambda = 3.2$ and the last 750 points follow a different exponential distribution, with $\lambda = 0.8$. Figure 6 only represents the interval between sample points 700 and 800, which is the interesting part of the trace for this matter. The figure shows that the adaptive bounds (which are more optimistic than the conservative ones) in the changing period are lower than the trace bounds, generating some timing faults. This occurs because the bounds are calculated based on the history mean, which do not correctly represent the expected value immediately after the environment change. We can observe in the figure that the "stable envirnment" detection time depends on the history size (which in our experiment is 30).

We believe that the fundamental conclusion to derive from the full set of experiments is that it is possible to define simple and efficient mechanisms that correctly detect stable and transient phases and, for the stable ones, well characterize the observed probabilistic distribution. Besides, the parameters for each mechanism can be fine tuned to better fit a given behavior and to configure more optimistic or more conservative approaches, depending on the considered applications and on the environment under

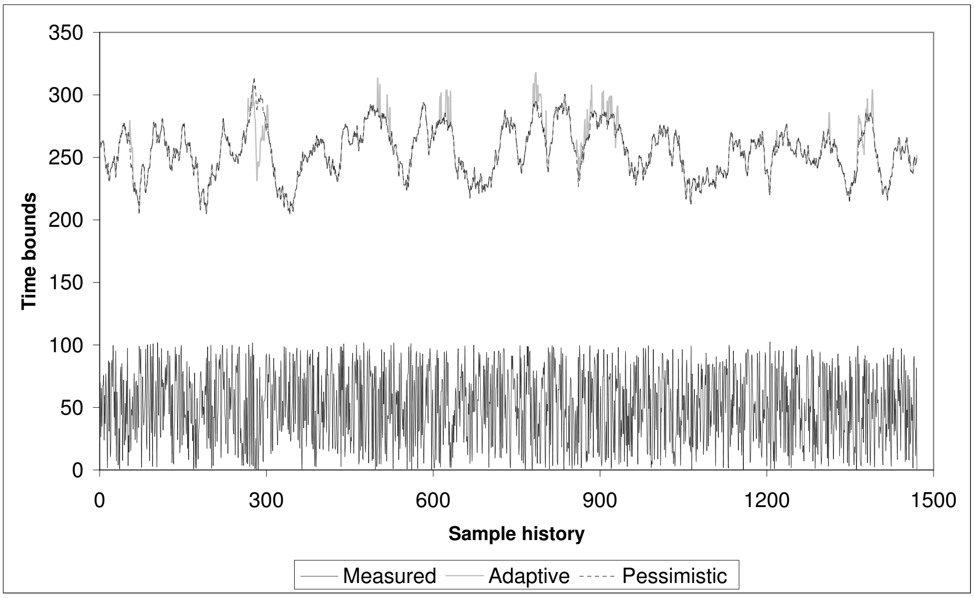

Figure 5: Pessimistic and adaptive approaches, using a random trace.

evaluation.

It is important to emphasize that the purpose of the presented experiments was to demonstrate that it is possible to implement simple but efficient mechanisms on characterizing the environment. To reach this objective, we used synthetic data traces following some known distributions. We did not present any result using real collected data because right now our framework is limited to only two distributions (exponential and Pareto). Since the current mechanisms have presented good results, we intend in the near future to extend the set of distributions as part of the development of a general middleware for dependable adaptation.

6 Conclusions and Future Work

In this paper we addressed the problem of supporting adaptive systems and applications in probabilistic environments, from a dependability perspective. While existing solutions are typically designed for specific environments and thus make more restrictive assumptions about the possible observable distributions, our goal is to design a general framework, independent of the specific distributions and detection mechanisms, and providing the adequate support to develop dependable adaptive applications whose bound can be secured with a known probability, or coverage.

We significantly advanced on previous work by making a few additional simple, but important assumptions: the environment behaves in a probabilistic way and can be characterized by known probabilistic distributions, which remain stable for sufficiently long periods of time. In fact, we assumed that a system alternates stable periods, during which the environment characteristics are fixed, and unstable periods, in which a variation of the environment conditions occurs. Based on that we proposed a general framework for adaptation, which allows to dynamically set optimistic time bounds when a stable phase is detected, while it provides pessimistic but still dependable bounds during transient phases. The proper identification of stable/unstable phases is therefore of primary importance, and so explained how the framework may be implemented with several simple phase detection mechanisms to be used in isolation or in cooperation.

We also provided some results that were obtained using an experimental implementation of the frame-

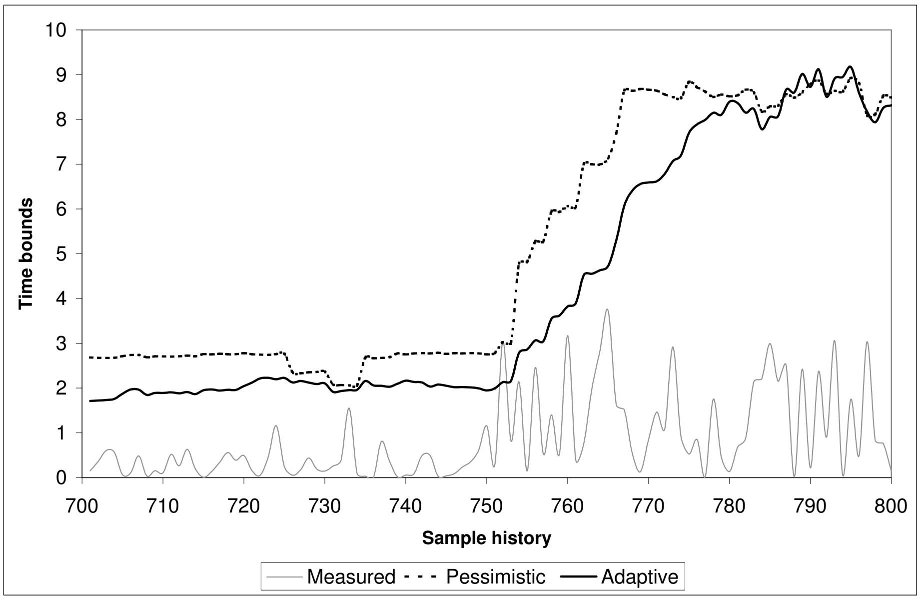

Figure 6: Reaction to a distribution change, using “Mechanism 1” for an exponential trace.

work, and with synthetic data flows generated from exponential and Pareto distributions. The implementation work and the achieved results provided interesting insights on the specific issues that must be addressed in some practical implementation, and on the possible outcomes and improvements achievable by using this framework.

In the near future we are planning to extend the set of available phase detection mechanisms, in particular considering other probabilistic distributions, and analyze their behavior in the presence of more or less complex environment changes. In addition, after adding support for other probabilistic scenarios, we aim to conducting experiments with real data traces, in particular considering some of the communication scenarios that were defined in the HIDENETS project.

Acknowledgements

We wish to acknowledge the contribution of Odorico Mendizabal, who participated in an earlier version of the document, and to Hans-Peter Schwefel and Hans Reiser, who read earlier manuscripts and provided helpful comments.

References

[1] A. Casimiro and P. Veríssimo. Using the Timely Computing Base for Dependable QoS Adaptation. In *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, pages 208–217, New Orleans, USA, Oct. 2001.

[2] I. M. Chakravarti, R. Laha, and J. Roy. Long-range dependence in variable-bit-rate video traffic. *Handbook of Methods of Applied Statistics*, 1:392–394, 1967.

[3] A. Corlett, D. Pullin, and S. Sargood. Statistics of one-way internet packet delays, Mar. 2002. http://www.potaroo.net/ietf/all-ids/draft-corlett-statistics-of-packet-delays-00.txt.

[4] T. Elteto and S. Molnar. On the distribution of round-trip delays in tcp/ip networks. In *Local Computer Networks, 1999. LCN '99. Conference on*, pages 172–181, 1999.

[5] J. A. Hernández and I. W. Phillips. Weibull mixture model to characterise end-to-end internet delay at coarse time-scales. *IEE Proc. Communications*, 153(2):295–304, Apr. 2006.

[6] HIDENETS Consortium. IST-FP6-STREP-26979 (HIDENETS: HIghly DEpendable IP-based NETworks and Services). Web page: http://www.hidenets.aau.dk.

[7] A. Markopoulou, F. A. Tobagi, and M. J. Karam. Loss and delay measurements of internet backbones. *Computer Communications*, 29(10):1590–1604, June 2006.

[8] M. Menth, J. Milbrandt, and J. Junker. Time-exponentially weighted moving histograms (TEWMH) for application in adaptive systems. In *Proceedings of the Global Telecommunications Conference (GLOBECOM '06)*, pages 1–6, Nov. 2006.

[9] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot. Measurement and analysis of single-hop delay on an ip backbone network. *IEEE Journal on Selected Areas in Communications. Special Issue on Internet and WWW Measurement, Mapping, and Modeling*, 21(6):908–921, Aug. 2003.

[10] N. Piratla, A. Jayasumana, and H. Smith. Overcoming the effects of correlation in packet delay measurements using inter-packet gaps. In *Proceedings of the 12th IEEE International Conference on Networks (ICON 2004)*, pages 233–238, Nov. 2004.

[11] K. S. Trivedi. Probability and statistics with reliability, queuing and computer science applications. John Wiley and Sons, 2002.

[12] P. Veríssimo and A. Casimiro. The Timely Computing Base model and architecture. *Transactions on Computers - Special Issue on Asynchronous Real-Time Systems*, 51(8):916–930, Aug. 2002.

[13] M. Yang, X. R. Li, H. Chen, and N. S. V. Rao. Predicting internet end-to-end delay: an overview. In *Proceedings of the 36th IEEE Southeastern Symposium on Systems Theory*, pages 210–214, Atlanta, GA, Mar. 2004.