Lightweight Verification with Stochastic Abstraction

Axel Legay

INRIA

in collaboration with Verimag, Grenoble

24/10/2010

Introduction

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

Introduction

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

- Verifying applications working within a subset of components of the system

Introduction

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

- Verifying applications working within a subset of components of the system

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

- Verifying applications working within a subset of components of the system

Problem:

Introduction

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

- Verifying applications working within a subset of components of the system

Problem:

- *A component may use global ressources*

Introduction

Our goal:

- Specification and analysis of large and complex heterogeneous systems

Verification:

- Verifying applications working within a subset of components of the system

Problem:

- *A component may use global ressources*
- *Behaviors of other components have to be considered*

Solution

Introduction

Solution

- *Stochastic Abstraction*

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system
  - Use Statistical Model Checking

Introduction

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system
  - Use Statistical Model Checking

- Advantages

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system
  - Use Statistical Model Checking

- Advantages
  - Stochastic abstraction considerably reduces the size of the heterogeneous system

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system
  - Use Statistical Model Checking

- Advantages
  - Stochastic abstraction considerably reduces the size of the heterogeneous system
  - Probabilities to quantify the level of failures

Introduction

Solution

- *Stochastic Abstraction*
  - Abstraction of global behaviors with probabilities
  - Analyse the resulting (smaller) stochastic system
  - Use Statistical Model Checking

- Advantages
  - Stochastic abstraction considerably reduces the size of the heterogeneous system
  - Probabilities to quantify the level of failures
  - Statistical Model Checking allows to go beyond classical reasonings

The rest of this talk

- An introduction to statistical model checking;
- An application of stochastic abstraction (EADS, COMBEST);
- A discussion regarding possible future work.

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

Verification Process

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

- $\mathcal{S}$ is a Stochastic system;

Verification Process

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

- $\mathcal{S}$ is a Stochastic system;
- $P_{\geq\theta}(X)$ means : "the probability for X to happen should be greater or equal to θ";
- θ is a probability threshold.

Verification Process

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

- $\mathcal{S}$ is a Stochastic system;
- $P_{\geq\theta}(X)$ means : "the probability for X to happen should be greater or equal to θ";
- θ is a probability threshold.
- Executions of $\mathcal{S}$ are sequences of states (random variables) that can be reached by following the transitions from an initial state;

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

- $\mathcal{S}$ is a Stochastic system;
- $P_{\geq\theta}(X)$ means : "the probability for X to happen should be greater or equal to θ";
- θ is a probability threshold.
- Executions of $\mathcal{S}$ are sequences of states (random variables) that can be reached by following the transitions from an initial state;
- φ is some execution-based property (specification language);

Verification Process

Question

Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?

where :

- $\mathcal{S}$ is a Stochastic system;
- $P_{\geq\theta}(X)$ means : "the probability for X to happen should be greater or equal to θ";
- θ is a probability threshold.
- Executions of $\mathcal{S}$ are sequences of states (random variables) that can be reached by following the transitions from an initial state;
- φ is some execution-based property (specification language);

Learning from a Simple Problem

- Consider a machine that flips a (possibly biaised) coin;
- Is the probability p of having a head greater or equal to some θ?

Learning from a Simple Problem

- Consider a machine that flips a (possibly biaised) coin;
- Is the probability p of having a head greater or equal to some θ?

A solution

- Do several flips and deduce the answer from them;
- If the number of flips is infinite, our answer will be correct up to some type error.

This is the statistical model checking approach!

Hypothesis Testing

Test $H_0 : P(\text{having a head}) \geq \theta$ against $H_1 : P(\text{having a head}) < \theta$

With (Type error):

Hypothesis Testing

Test $H_0 : P(\text{having a head}) \geq \theta$ against $H_1 : P(\text{having a head}) < \theta$

With (Type error):

1. α : the probability to accept H_1 while H_0 is true;
2. β : the probability to accept H_0 while H_1 is true.

The approach can also be used to compute the probability (PESTIMATION, Monte Carlo)

Indifference Region

We want to test :

$$H_0 : p \geq p_0 \text{ against } H_1 : p \leq p_1, \text{ where}$$
$$p_0 = \theta + \delta \text{ and } p_1 = \theta - \delta.$$

With:

Indifference Region

We want to test :

$$H_0 : p \geq p_0 \text{ against } H_1 : p \leq p_1, \text{ where}$$
$$p_0 = \theta + \delta \text{ and } p_1 = \theta - \delta.$$

With:

- Type erros α and β, and
- Indifference region 2δ (needed to terminate in finite time).
- Parameters influence the number of simulations.

Bernouili Variables for experiments

- Bernouili variable X_i of parameter p
  - Takes two values : $X_i = 0$ or $X_i = 1$;
  - $P[X_i = 1] = p$ and $P[X_i = 0] = 1 - p$;
  - Realization is denoted x_i.

Bernouili Variables for experiments

- Bernouili variable X_i of parameter p
  - Takes two values : $X_i = 0$ or $X_i = 1$;
  - $P[X_i = 1] = p$ and $P[X_i = 0] = 1 - p$;
  - Realization is denoted x_i.
- Experiments:
  - We assume independent trials;
  - We can generate as much trials as we want;
  - p is the probability to get a head ;
  - Associate a bernouili variable X_i to each trial;
  - $X_i = 1$ iff the trial is a tail.

Two Algorithms

- Algorithm 1 : Single Sampling plan (SSP):
  - Pre-compute a number *n* of experiments;
  - *n* depends on δ,α, and β.

Two Algorithms

- Algorithm 1 : Single Sampling plan (SSP):
  - Pre-compute a number *n* of experiments;
  - *n* depends on δ,α, and β.

- Algorithm 2: Basically a on-the-fly version of the Single Sampling Plan

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;
- n observations $x_1, \ldots, x_n$ for n samplings $X_1, \ldots, X_n$;

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;
- n observations $x_1, \ldots, x_n$ for n samplings $X_1, \ldots, X_n$;
- $Y = \sum_{i=1}^{n} x_i$;

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;
- n observations $x_1, \ldots, x_n$ for n samplings $X_1, \ldots, X_n$;
- $Y = \sum_{i=1}^{n} x_i$;
- Accept H_0 if $Y \geq c$ and H_1 otherwise;

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;
- n observations $x_1, \ldots, x_n$ for n samplings $X_1, \ldots, X_n$;
- $Y = \sum_{i=1}^{n} x_i$;
- Accept H_0 if $Y \geq c$ and H_1 otherwise;

Single Sampling plan : Principles

- Choose n and c with $c \leq n$;
- n observations $x_1, \ldots, x_n$ for n samplings $X_1, \ldots, X_n$;
- $Y = \sum_{i=1}^{n} x_i$;
- Accept H_0 if $Y \geq c$ and H_1 otherwise;

Difficulty : Find n and c such that α and β are satisfied

Single Sampling plan : Disadvantages

- Computing c and n is equivalent to solve an optimal problem on a sequence of binomial equations;
- This is difficult : No unique solution;
- Difficult to minimize n;
- Approximation algorithms exist (Haakan Youness).
- Better for black box systems

Sequential Hypothesis Testing

- Check hypothesis after each sample and stop as soon as possible
- We can find an acceptance line and a rejection line given $\alpha, \beta, \theta, \delta$.

Wald's Testing (SPRT)

Compute

$$W = \prod_{i=1}^{m} \frac{Pr(X_i = x_i \mid p = \theta - \delta)}{Pr(X_i = x_i \mid p = \theta + \delta)} = \frac{(\theta - \delta)^{d_m}(1 - \theta + \delta)^{m - d_m}}{(\theta + \delta)^{d_m}(1 - \theta - \delta)^{m - d_m}}, \qquad (1)$$

where $d_m = \sum_{i=1}^{m} x_i$.

Wald's Testing (SPRT)

Compute

$$W = \prod_{i=1}^{m} \frac{Pr(X_i = x_i \mid p = \theta - \delta)}{Pr(X_i = x_i \mid p = \theta + \delta)} = \frac{(\theta - \delta)^{d_m}(1 - \theta + \delta)^{m - d_m}}{(\theta + \delta)^{d_m}(1 - \theta - \delta)^{m - d_m}}, \tag{1}$$

where $d_m = \sum_{i=1}^{m} x_i$.

Stop when :

- $W \geq (1 - \beta)/\alpha$: H_1 is accepted;
- $W \leq \beta/(1 - \alpha)$: H_0 is accepted.

More Mathematics

- In theory : the test does not guarantee α and β!
- New parameters α' and β' such that

More Mathematics

- In theory : the test does not guarantee α and β!
- New parameters α' and β' such that
  - $\alpha' \leq \frac{\alpha}{1-\beta}$ and $\beta' \leq \frac{\beta}{1-\alpha}$
  - $\alpha' + \beta' \leq \alpha + \beta$;
- In practice : one observes that α and β are almost often guarantee, and it may even be better!

Example

Let $p_0 = 0.5$, $p_1 = 0.3$, $\alpha = 0.2$, $\beta = 0.1$:

- In theory : $\alpha' \leq \frac{0.2}{0.9} = 0.222...$ and $\beta' \leq \frac{0.1}{0.8} = 0.125$;
- Computer simulation : $\alpha' = 0.175$ and $\beta' = 0.082$.

From Flipping a coin to Model Checking

- Flipping a coin is nothing more than testing whether a finite execution satisfies a property.
- Consequence : Wald's testing directly applies to model check properties of white-box stochastic systems.

From Flipping a coin to Model Checking

- Flipping a coin is nothing more than testing whether a finite execution satisfies a property.
- Consequence : Wald's testing directly applies to model check properties of white-box stochastic systems.

Properties

- Natural : those that can be checked on finite executions: next, bounded until;
- Better than classical logics : Clock drift, Fourier Transform, Systems Biology.

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);
- Independent of system's size;

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);
- Independent of system's size;
- Independent of system's probability distribution;

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);
- Independent of system's size;
- Independent of system's probability distribution;
- Easy to trade accuracy for speed;

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);
- Independent of system's size;
- Independent of system's probability distribution;
- Easy to trade accuracy for speed;
- Uniform approach;

Advantages

- Easy to parallelize (independent sampling, unbiased distributed sampling);
- Independent of system's size;
- Independent of system's probability distribution;
- Easy to trade accuracy for speed;
- Uniform approach;
- Easy to implement :

- In most cases, one only need to implement a "trace checker" that tests whether an execution satisfies a given property;
- No need for complex data structures.

Case Study: Accuracy of clock Synchronization (EADS)

Chalenges:

- Heterogeneous System over an Ethernet backbone
  - Distributed application
  - 280 communicating components
- Local clocks synchronized using the Precision Time Protocol
- Requirement: Verify that the difference between any 2 clocks is lower than a given bound

Case Study: Accuracy of clock Synchronization (EADS)

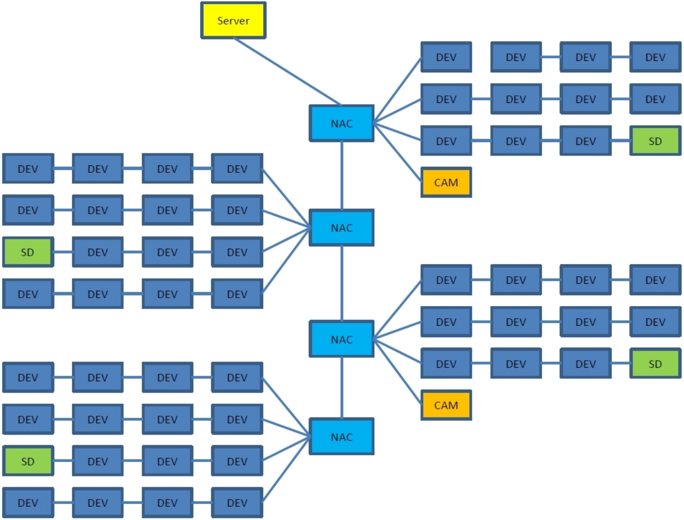
Chalenges:

- Heterogeneous System over an Ethernet backbone
  - Distributed application
  - 280 communicating components
- Local clocks synchronized using the Precision Time Protocol
- Requirement: Verify that the difference between any 2 clocks is lower than a given bound
- Our goals: (1) Compute the best bound to satisfy this requirement without analyzing the whole architecture in a step, (2) compute the probability for a bound fixed by EADS ($50\mu s$).

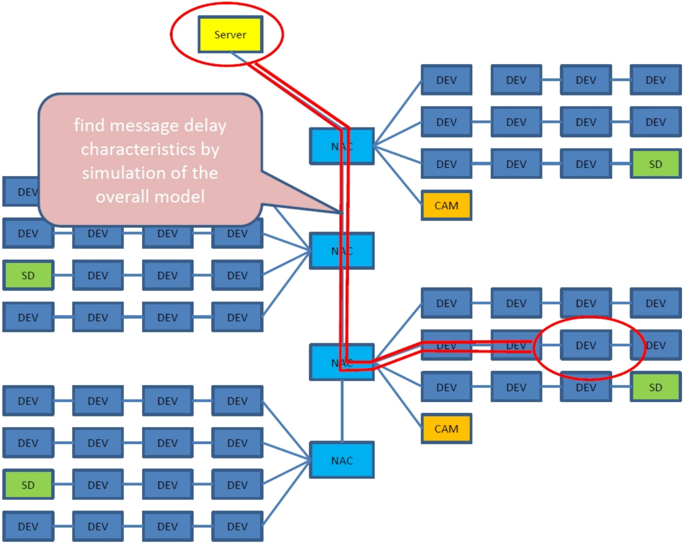
Solution

- Apply stochastic abstraction between any device an the system;
- Compute the probability to synchronize for several values of the bound;
- Proceed similarly for all the devices;
- Keep the minimal bound for which synchonization is guarantee with probability 1

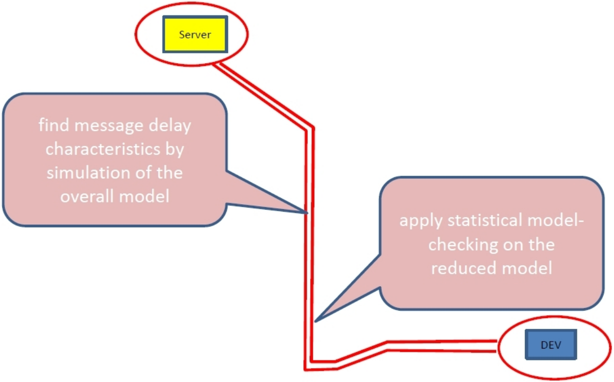
How to Compute Stochastic Abstraction?



How to Compute Stochastic Abstraction?



How to Compute Stochastic Abstraction?



What do we need?

A tool

- whose input language is powerful enough to describe the EADS case study;
- in where stochastic aspects can easily be described;
- in where statistical model checking algorithms can easily be implemented;
- with an engine capable to generate executions in an efficient manner.

Our Solution: BIP!

First step

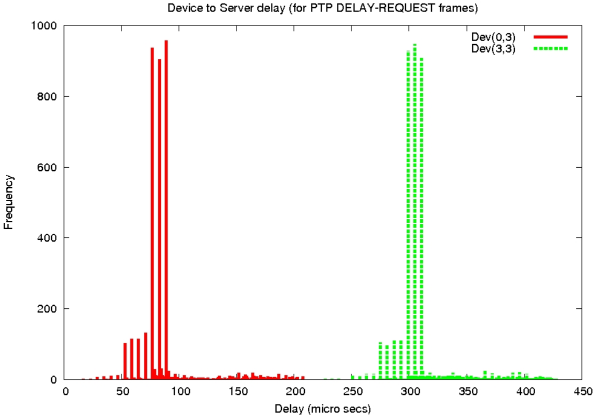
1. Learn the Probability distributions (1)

Run several simulations on the big model and extract the delays and the number of time they occurs;

First step

1. Learn the Probability distributions (1)

Run several simulations on the big model and extract the delays and the number of time they occurs;



2. Use the distributions to study PTP

Let assume that we measured the delays 33 times. The result will be a series of delay values and, for each value, the number of times it has been observed. As an example, delay 5 has been observed 3 times, delay 19 has been observed 30 times. The probability distribution is represented with a table of 33 cells. In our case, 3 cells of the table will contains the value 5 and 30 will contain the value 19. The BIP engine will simply select a value in the table following a uniform probability distribution.

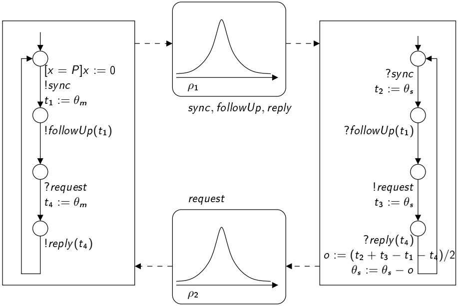
Second Step

3. Producing Stochastic Abstraction

Stochastic choices are directly integrated in the BIP engine!

Second Step

3. Producing Stochastic Abstraction



Stochastic choices are directly integrated in the BIP engine!

Statistical model checking (1)

What are the questions?

- Qualitative Question : Does $\mathcal{S} \models P_{\geq\theta}(\varphi)$?
- Quantitative Question : What is the probability for $\mathcal{S}$ to satisfy φ?

Principle

- Reason on a finite set of executions and answer the question;
- We may make mistakes, but we should be as precise as we want!

Statistical model checking (2)

Qualitative question :

- Two algorithsm for the qualitative question : SPRT and SSP;
- They say yes or no, but can make a mistake (confidence).

Quantitative question :

- PESTIMATION computes an estimation p' of the probability for $\mathcal{S}$ to satisfy φ;
- The estimation can be bound : $|p - p'| < q$;
- The algorithm can make a mistake.

Statistical model checking (3)

- PESTIMATION is much slower that SSP or SPRT;
- A good strategy for answering the quantitative question :
    - Computer and estimation with a low confidence;
    - Validate this estimation with SSP or SPRT, but with a high confidence.

In Practice

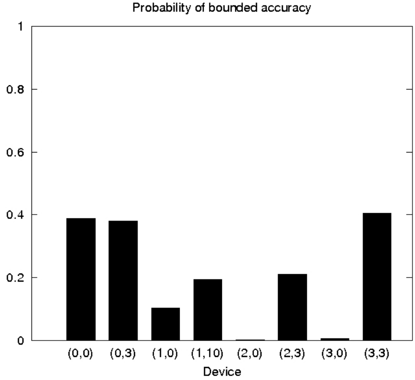
Model/Abstraction:

- PTP and HCS modeled using BIP
- Distributions of delays: 2000 measures

Statistical Model Checking:

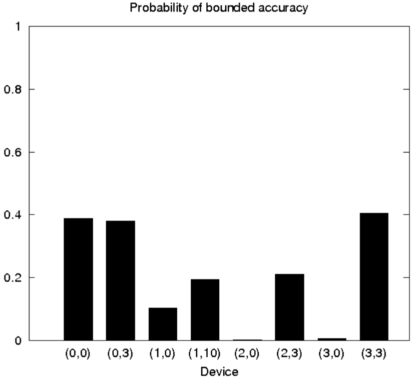
- Quantitative question: precision 10^{-2}, confidence 10^{-2}: 100000 simulations
- Qualitative question: precision 10^{-3}, confidence 10^{-10}: 300000 simulations

Some Results 1/2



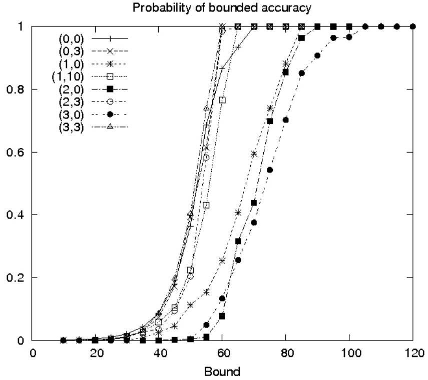
Probability of satisfying Bounded Accuracy for a bound of $50\mu s$

Some Results 1/2



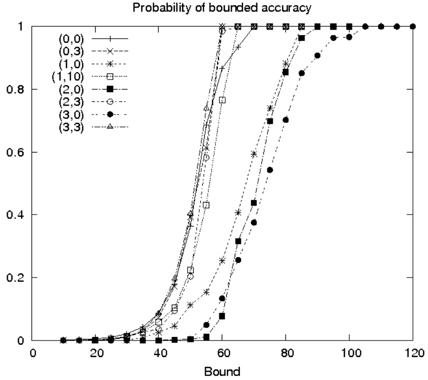
Probability of satisfying Bounded Accuracy for a bound of $50\mu s$

- The property is not satisfied for the given bound !



Probability of satisfying Bounded Accuracy as a function of the bound

Some Results 2/2



Probability of satisfying Bounded Accuracy as a function of the bound

- The best bound for which B.A. is satisfied with probability 1 is $105\mu s$

SPRT VS. SSP VS. PESTIMATION

Precision	10^{-1}		10^{-2}		10^{-3}	
Confidence	10^{-5}	10^{-10}	10^{-5}	10^{-10}	10^{-5}	10^{-10}
PESTIMATION	4883 $17s$	9488 $34s$	488243 $29m$	948760 $56m$	48824291 $> 3h$	94875993 $> 3h$
SSP	1604 $10s$	3579 $22s$	161986 $13m$	368633 $36m$	16949867 $> 3h$	32792577 $> 3h$
SPRT	316 $2s$	1176 $7s$	12211 $53s$	22870 $1m38s$	148264 $11m$	311368 $31m$

Precision	10^{-1}		10^{-2}		10^{-3}	
Confidence	10^{-5}	10^{-10}	10^{-5}	10^{-10}	10^{-5}	10^{-10}
SSP / SPRT	110 $1s$	219 $1s$	1146 $6s$	2292 $13s$	11508 $51s$	23015 $1m44s$

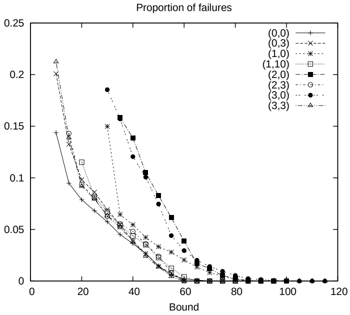
Proportion of Failures (1)

- Proportion of failure = number of failure divided by number of observations on a simulation.
- What is the average proportion of failure?
- What is the worst proportion of failure?

Solution : compute it on 2000 simulations with $\Delta = 50\mu s$:

- Device $(2, 0)$ gives $0, 24$;
- Device $(0, 0)$ gives $0, 076$.

Proportion of Failures (2)



Average proportion of failures as function of the bound Δ for the asymmetric version

Influence of Drift?

- Drift is used to model the fact that, due to the influence of the material, clocks of the master and the device may not progress as the same rate;
- Each time the clock of the server is increased by 1 time unit, the clock of the device is increased by $1 + 10^{-3}$ time units

Observations :

- Experiments : almost the same probabilities; Reason : the value of the drift is much smaller than the on of the jitter;
- With a drift of one time unit : Strong influence : Device $(0, 0)$ goes from a probability of $0,387$ to a probability of $0,007$.

Drift + Simulation = EASY;
Drift + exhaustive Verification = ALMOST IMPOSSIBLE

Results for stochastic abstraction

- Abstraction and verification method
- Applied to 2 case studies:
  - HCS case study [FORTE'10]
  - AFDX network [RV'10]

Future work

- Static analysis of the code for automatic extraction of the abstraction;
- Mixing confidence on the distribution with confidence on the statistical model checking algorithm;
- Handling non determinism;
- Using our knowledge of the system to improve the statistical model checking procedure;
- Various components described in various formalisms.