Probabilistic Verification of Sensor Networks

Experimenting a New Framework for Sensor Networks

Akim Demaille[1] Thomas Hérault[2] Sylvain Peyronnet[1]

[1]EPITA Research and Development Laboratory (LRDE)

[2]LRI - University of Paris XI

Research, Innovation and Vision for the Future 2006

Sensor

A sensor:

- miniature device
- low-cost ($1)
- limited computation power
- limited energy

Sensor

A sensor:

- miniature device
- low-cost ($1)
- limited computation power
- limited energy

A sensor network:

- a large number of sensors (1M)
- ad-hoc network
- randomly spread

Sensor

A sensor:

- miniature device
- low-cost (\$1)
- limited computation power
- limited energy

To monitor an area, e.g.,

- intrusion detection
- fire surveillance
- ...

A sensor network:

- a large number of sensors (1M)
- ad-hoc network
- randomly spread

Sensor Networks

The Challenges

- to design efficient communication algorithms
- to ensure their correctness

Sensor Networks

The Challenges

- to design efficient communication algorithms
- to ensure their correctness

The Complications

- the behavior of the network is probabilistic
- the number of nodes is huge

Sensor Networks

The Challenges

- to design efficient communication algorithms
- to ensure their correctness

The Complications

- the behavior of the network is probabilistic
- the number of nodes is huge

The Techniques

- Simulation
- Model checking

Sensor Networks

The Challenges

- to design efficient communication algorithms
- to ensure their correctness

The Complications

- the behavior of the network is probabilistic
- the number of nodes is huge

The Techniques

- Simulation
- Model checking
- Approximate Probabilistic Model Checking

Probabilistic Verification of Sensor Networks

Randomized Approximation Scheme

$$p = \mathrm{Prob}\left[\psi\right]$$

Randomized Approximation Scheme

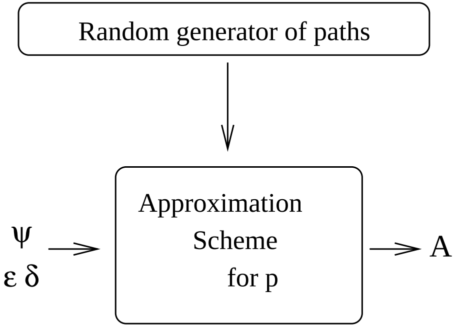
$$p = \mathrm{Prob}\left[\psi\right]$$

$$\mathrm{Prob}\Big[\,(p - \varepsilon) \leq A \leq (p + \varepsilon)\,\Big] \geq 1 - \delta$$

Randomized Approximation Scheme

$$p = \mathrm{Prob}\,[\psi]$$

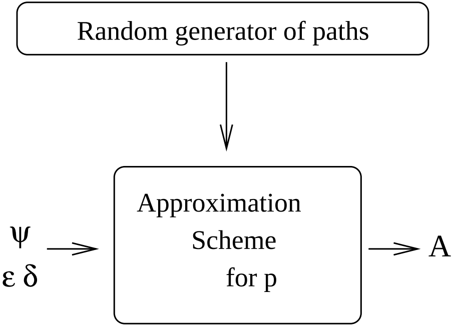
$$\mathrm{Prob}\Big[\,(p-\varepsilon) \leq A \leq (p+\varepsilon)\,\Big] \geq 1-\delta$$



Randomized Approximation Scheme

$$p = \mathrm{Prob}\,[\psi]$$

$$\mathrm{Prob}\Big[\,(p-\varepsilon) \leq A \leq (p+\varepsilon)\,\Big] \geq 1-\delta$$



Definition (FPRAS)

Fully polynomial randomized approximation scheme with time complexity
$\mathrm{poly}(\log(1/\delta), |\,\psi\,|, (1/\varepsilon))$

Our algorithm

Algorithm (Generic approximation)

input: ψ, *diagram*, ϵ, δ

Let $P := 0$

Let $N := \frac{1}{2}\log(\frac{2}{\delta})/\epsilon^2$

For i *from* 1 *to* N *do*

1. *Generate a random path* σ *of depth* k
2. *If* ψ *is true on* σ *then* $P := P + 1$

Return $A = P/N$

Our algorithm

Algorithm (Generic approximation)

input: $\psi, \textit{diagram}, \epsilon, \delta$

Let $P := 0$

Let $N := \frac{1}{2}\log(\frac{2}{\delta})/\epsilon^2$

For i *from* 1 *to* N *do*

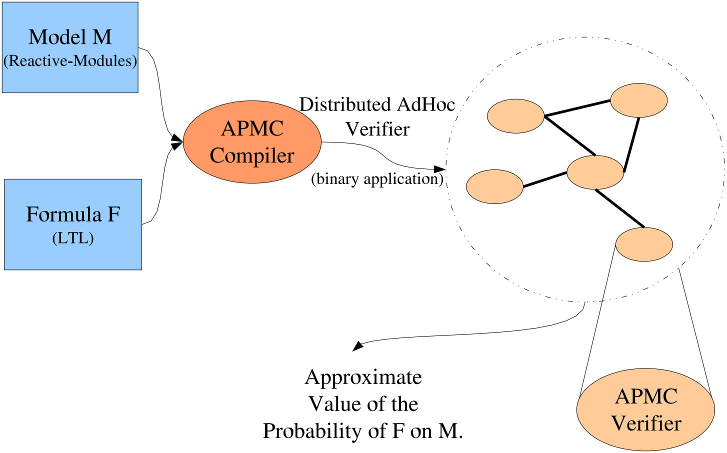
1. *Generate a random path* σ *of depth* k
2. *If* ψ *is true on* σ *then* $P := P + 1$

Return $A = P/N$

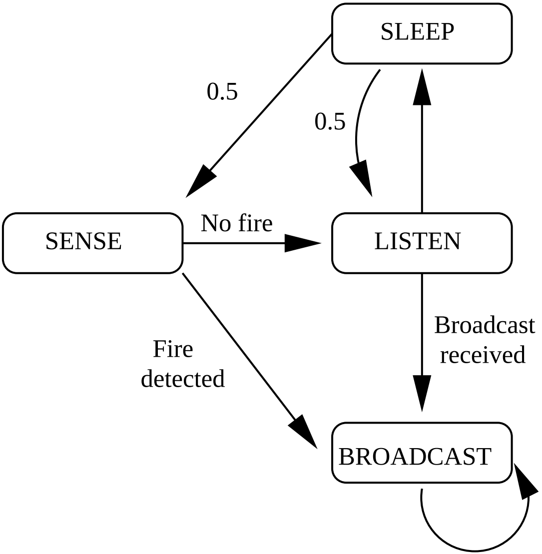
Theorem

This algorithm is an additive FPRAS for $\mathrm{Prob}[\psi]$

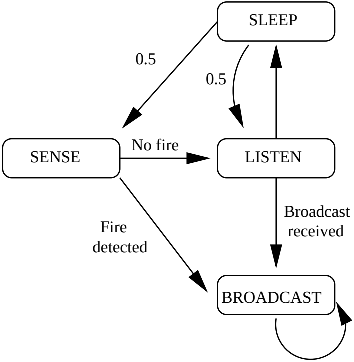
Architecture



A single sensor

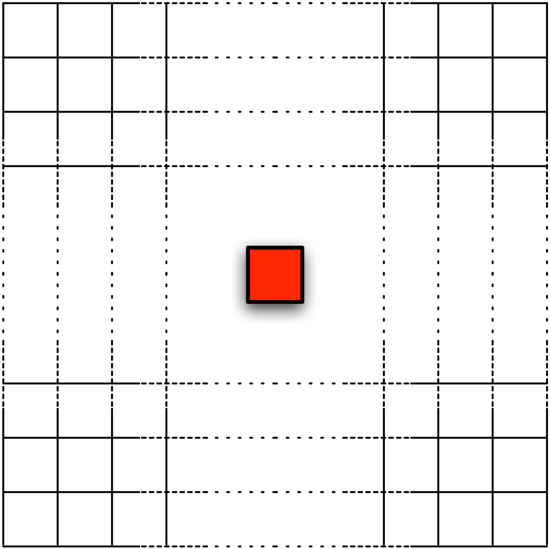


A single sensor

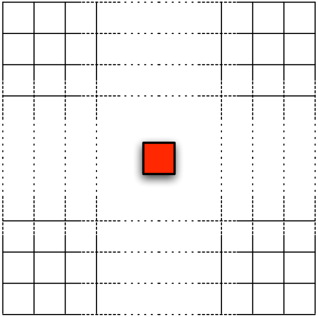



```
module sensor

  s : [0..4]  init SENSE;

  [] s = SENSE
    -> s = senses ? BROADCAST : LISTEN;
  [] s = LISTEN
    -> s = receives ? BROADCAST : SLEEP;
  [] s = SLEEP
    -> 0.5 : s = SENSE
     + 0.5 : s = LISTEN;
  [] s = BROADCAST
    -> s = BROADCAST;

endmodule
```

A sensor network

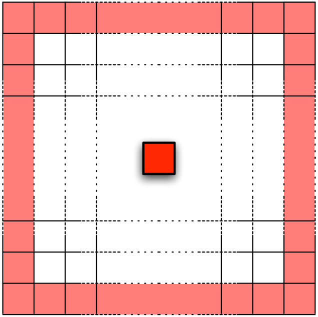


A sensor network

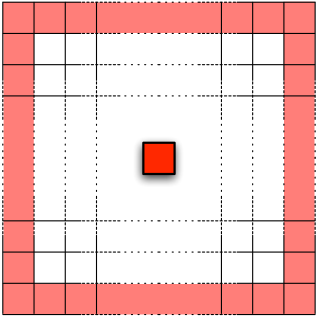


```
m4_for([X], [0], MAX_X, [1],
       [m4_for([Y], [0], MAX_Y, [1],
               [rm_sensor(X, Y)])])
```

The detection of an event

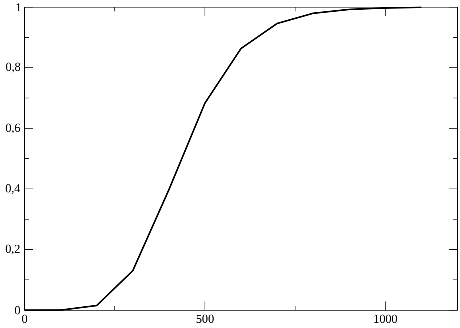


The detection of an event

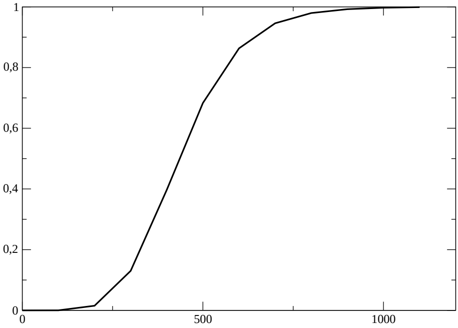


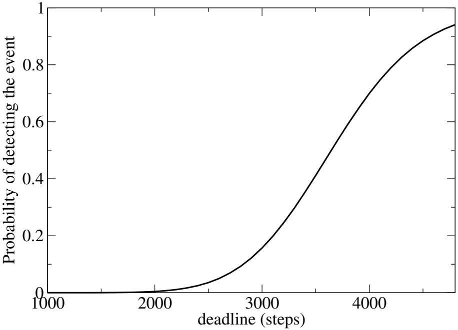
```
m4_for([T], [0], LENGTH, [100],
[true U (t <= T
          & (0 rm_foreach_boundary([X], [Y],
            [ | rm_state(X, Y) = BROADCAST])))
])
```

Estimating the path length

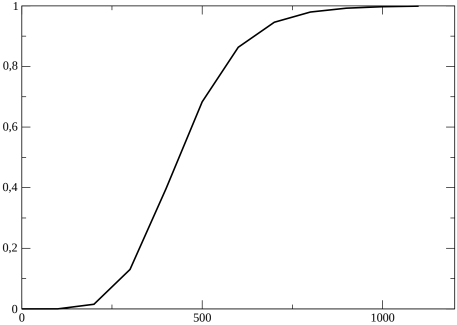


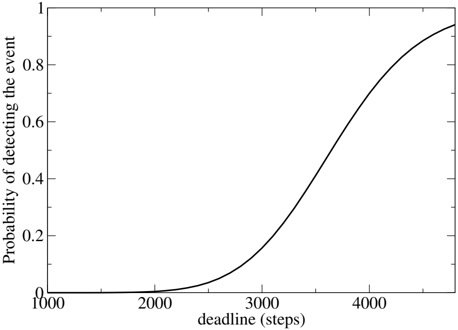
Estimating the path length





Estimating the path length



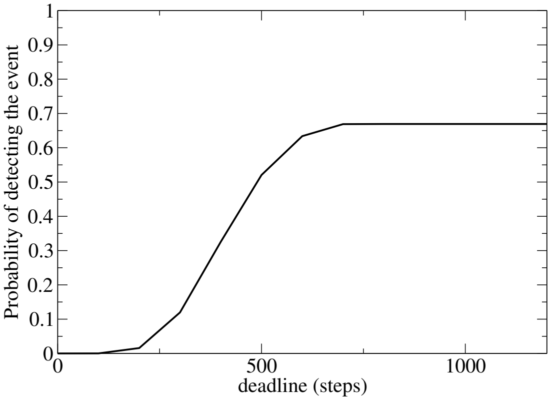


$$\text{pathlength} = 15 \times \text{area}$$

Impact of the limited energy

Costs:

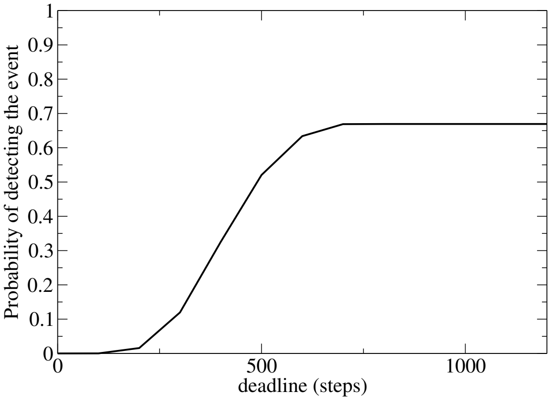
SLEEP	1
SENSE	2
LISTEN	2
BROADCAST	3



Impact of the limited energy

Costs:

SLEEP	1
SENSE	2
LISTEN	2
BROADCAST	3

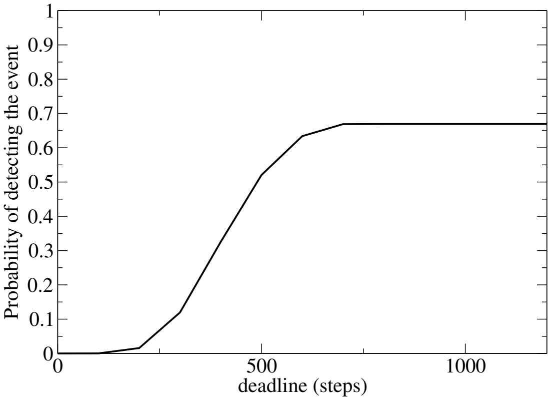


This amount of energy is insufficient.

Impact of the limited energy

Costs:

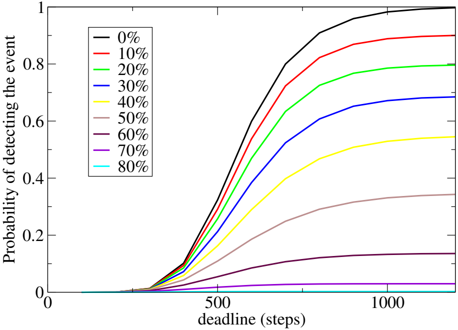
SLEEP	1
SENSE	2
LISTEN	2
BROADCAST	3



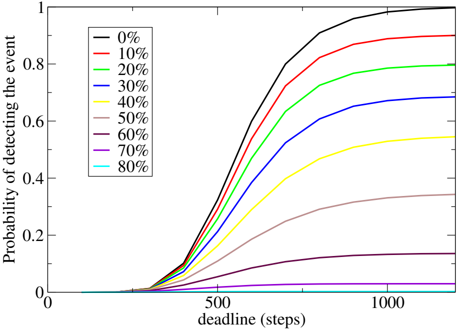
This amount of energy is insufficient.

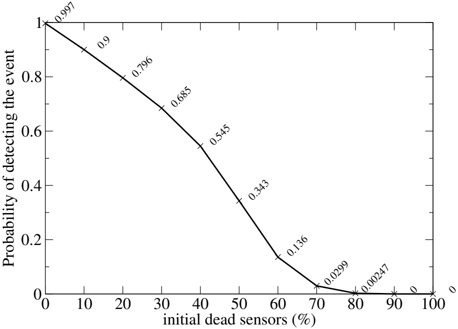
Application: optimize the time spent in each mode.

Resistance to the initial loss

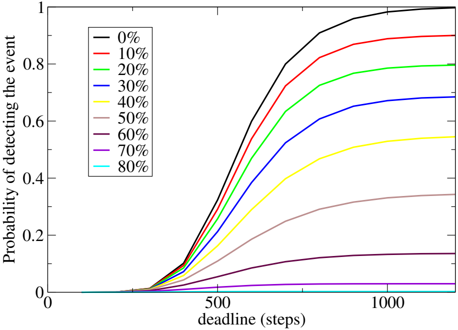


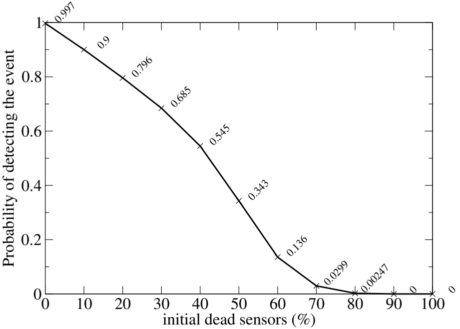
Resistance to the initial loss





Resistance to the initial loss





Two phases:

- $\leq 35\%$ linear, elastic, robust — delayed delivery.
- $35\% \leq$ brutal decreasing, compromised delivery.

Lessons Learned

- APMC is suitable for large models

Lessons Learned

- APMC is suitable for large models
- Our straightforward modeling with DTMC does not model faithfully battery consumption

Lessons Learned

- APMC is suitable for large models
- Our straightforward modeling with DTMC does not model faithfully battery consumption
- Reactive Module is inconvenient for large simple models

Lessons Learned

- APMC is suitable for large models
- Our straightforward modeling with DTMC does not model faithfully battery consumption
- Reactive Module is inconvenient for large simple models
- Some design decisions made for protocol study have to be reconsidered for large models

Future Work

- Extend Reactive Modules to program in the large

Future Work

- Extend Reactive Modules to program in the large
- Remove arbitrary limitations from APMC

Future Work

- Extend Reactive Modules to program in the large
- Remove arbitrary limitations from APMC
- Use CTMC to model conveniently battery consumption

Future Work

- Extend Reactive Modules to program in the large
- Remove arbitrary limitations from APMC
- Use CTMC to model conveniently battery consumption
- Use a CTMC-able probabilistic model checker

Questions?