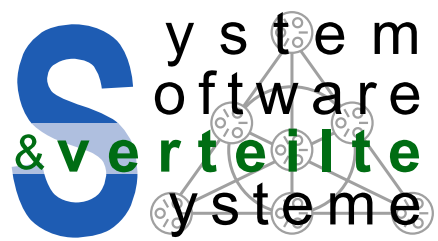
Combining Decomposition and Lumping to Evaluate Semi-hierarchical Systems

Nils Müllner, Oliver Theel, Martin Fränzle

Fakultät II, Department für Informatik
Carl von Ossietzky Universität Oldenburg



May 16, 2014

Agenda I

1. Brief Problem Outline
2. Practical Application
3. Conclusion

Underlying question

- How *dependable* is a distributed system over time?
- Measurable answer via **reliability** (failsafe fault tolerant) or **availability** (non-masking fault tolerant)
  - availability:
    probability that system is *safe at some time point*
  - reliability:
    probability that system is *safe until some time point*

Approach

Analytically computing system dependability:

- constructing a transition model (e.g. Markov model or Petri net) from a deterministic system model and a probabilistic environment model.

Challenge with this method:

- the transition model is exponential in the size of the system model $\Rightarrow$ *state space explosion!*

Problem Outline

To compute the dependability of a distributed system, its transition model must be constructed. The approach is inherently confined as the transition model is exponential in the size of the system model. A method to dampen the state space explosion is required.

General Approach

Dissect/Decompose system; analyze sub-systems step-by-step.

Related Work

Previous work covered

- **hierarchically structured** systems [AINA2012]: decomposed *top→down*.

A parallel study emphasized that

- **unstructured systems** [AINA2014a] are no challenge: decomposition *arbitrary*.

This study now discusses

- **semi-hierarchical systems**: accounting for local cyclic dependencies.

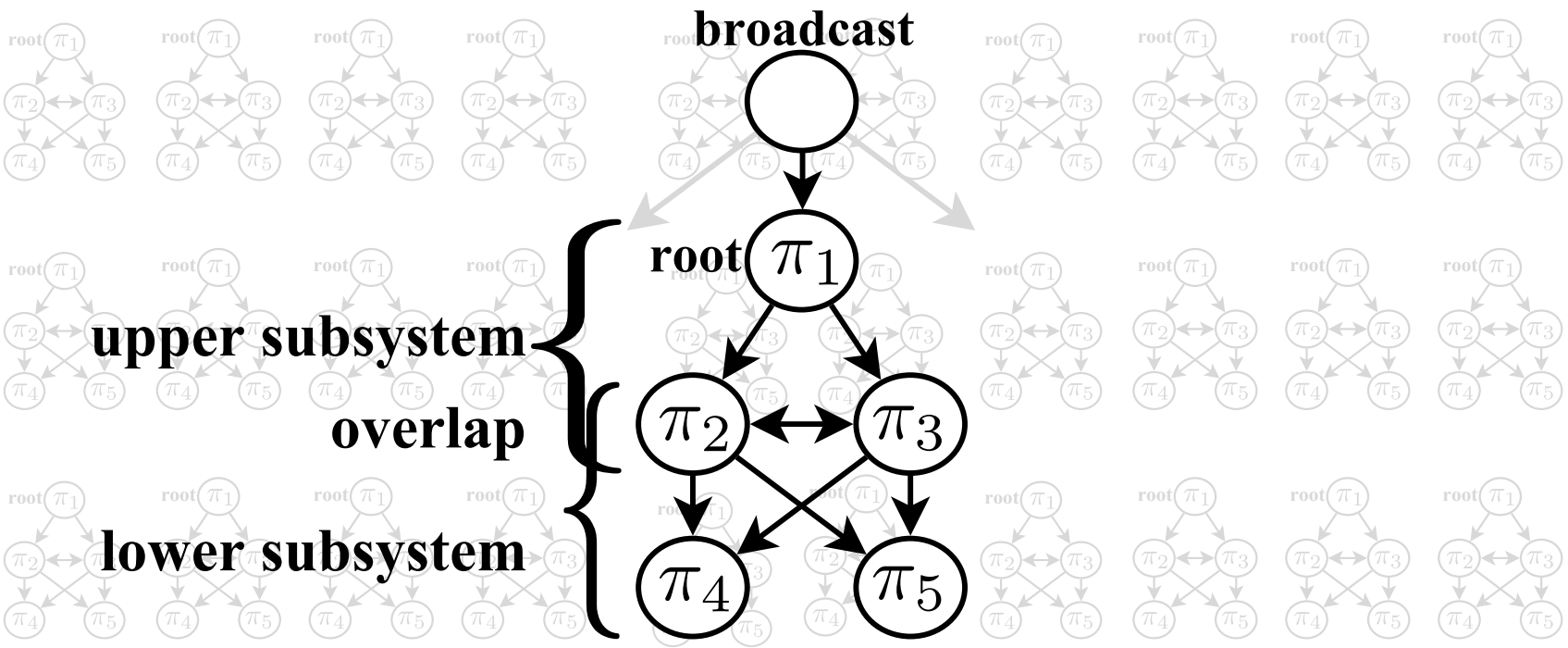
Motivation

- Decomposition of distributed systems relies on the dependencies among processes.
- When there are no dependencies, each process can be evaluated on its own, eventually arriving at a *counting abstraction* over all processes.
- On the opposite site, each process relies on all other processes (heterarchy). Global cyclic dependencies cannot be solved (accurately).
- Decomposition of hierarchic systems received the BPA two years ago [AINA2012].
- To close this cycle, decomposition with **local** cyclic dependencies is discussed.

Motivation

- Decomposition of distributed systems relies on the dependencies among processes.
- When there are no dependencies, each process can be evaluated on its own, eventually arriving at a *counting abstraction* over all processes.
- On the opposite site, each process relies on all other processes (heterarchy). **Global** cyclic dependencies cannot be solved (accurately).
- Decomposition of hierarchic systems received the BPA two years ago [AINA2012].
- To close this cycle, decomposition with **local** cyclic dependencies is discussed.

Wireless Sensor Network



Particularities

- semi-parallel execution semantics
- weak probabilistic scheduler
- sporadic faults affect only volatile process memory
- sub-systems overlap in cycle
- safety: each process stores currently broadcasted type

Particularities

- semi-parallel execution semantics
- weak probabilistic scheduler
- sporadic faults affect only volatile process memory
- sub-systems overlap in cycle
- safety: each process stores currently broadcasted type

Particularities

- semi-parallel execution semantics
- weak probabilistic scheduler
- sporadic faults affect only volatile process memory
- sub-systems overlap in cycle
- safety: each process stores currently broadcasted type

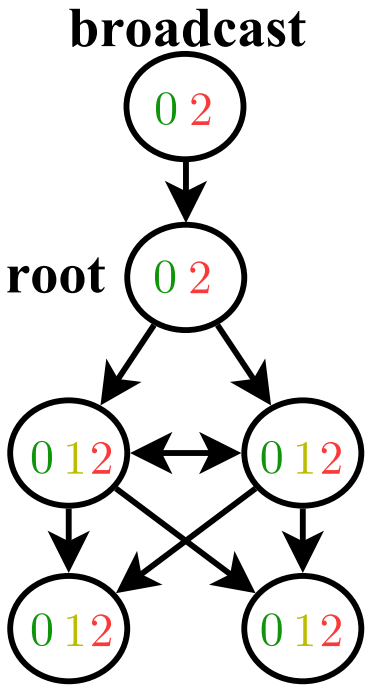
Particularities

- semi-parallel execution semantics
- weak probabilistic scheduler
- sporadic faults affect only volatile process memory
- sub-systems overlap in cycle
- safety: each process stores currently broadcasted type

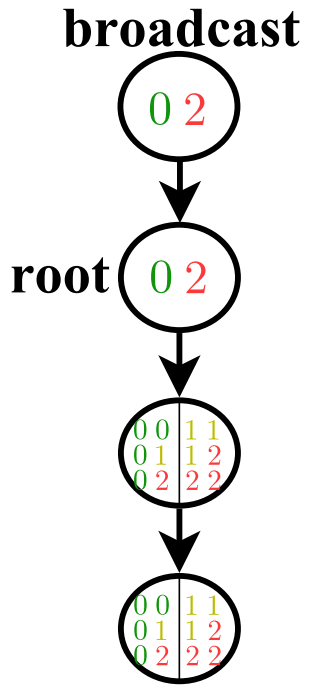
Particularities

- semi-parallel execution semantics
- weak probabilistic scheduler
- sporadic faults affect only volatile process memory
- sub-systems overlap in cycle
- safety: each process stores currently broadcasted type

States



(a) Full state space



(c) Lumped state space

Figure: State space reduction by lumping bisimilar states

$2 \cdot 2 \cdot 3^4 = 324$ states $2 \cdot 2 \cdot 6 \cdot 6 = 144$ states

Decomposition Schema

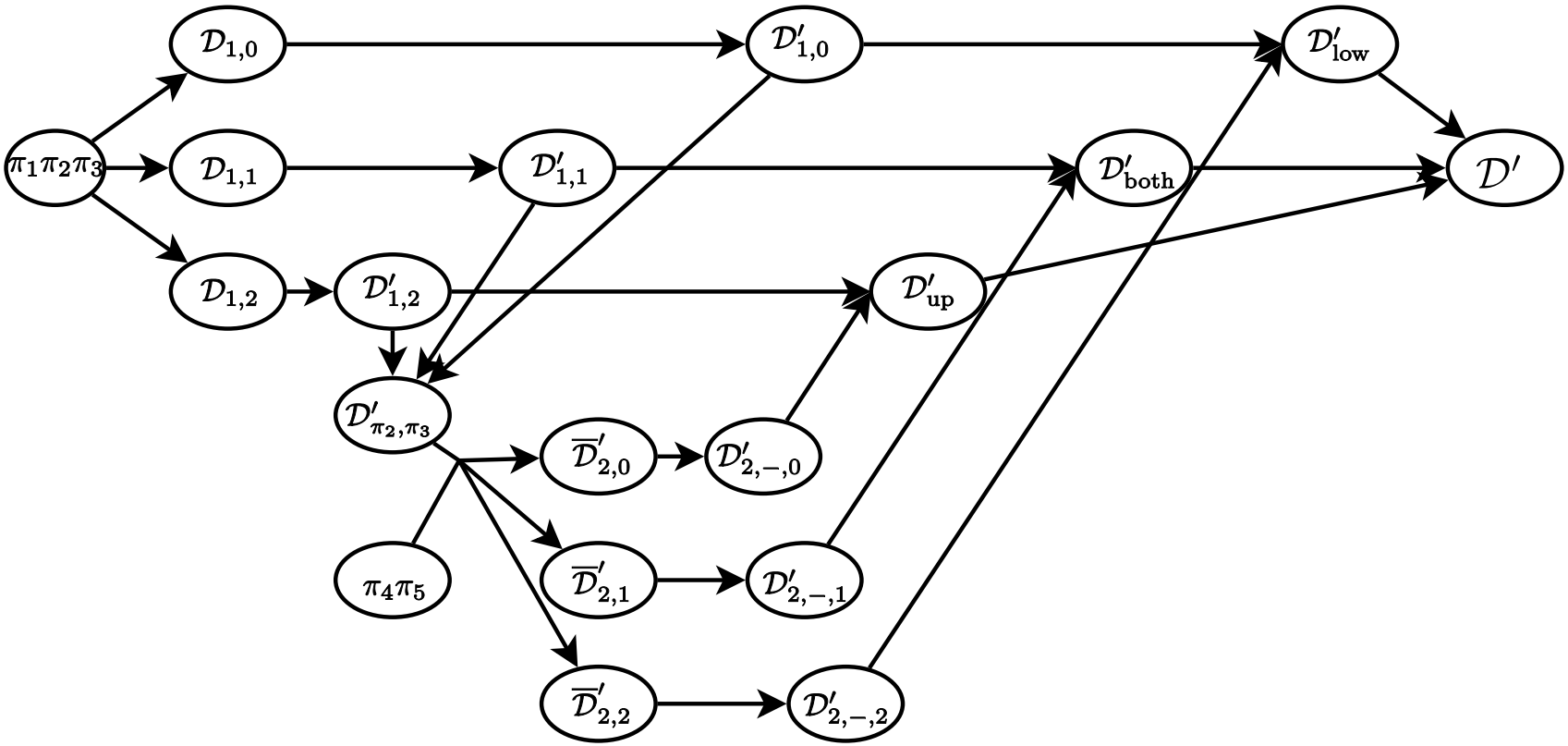


Figure: Decomposing schema for the WSN transition system

Decomposition Schema

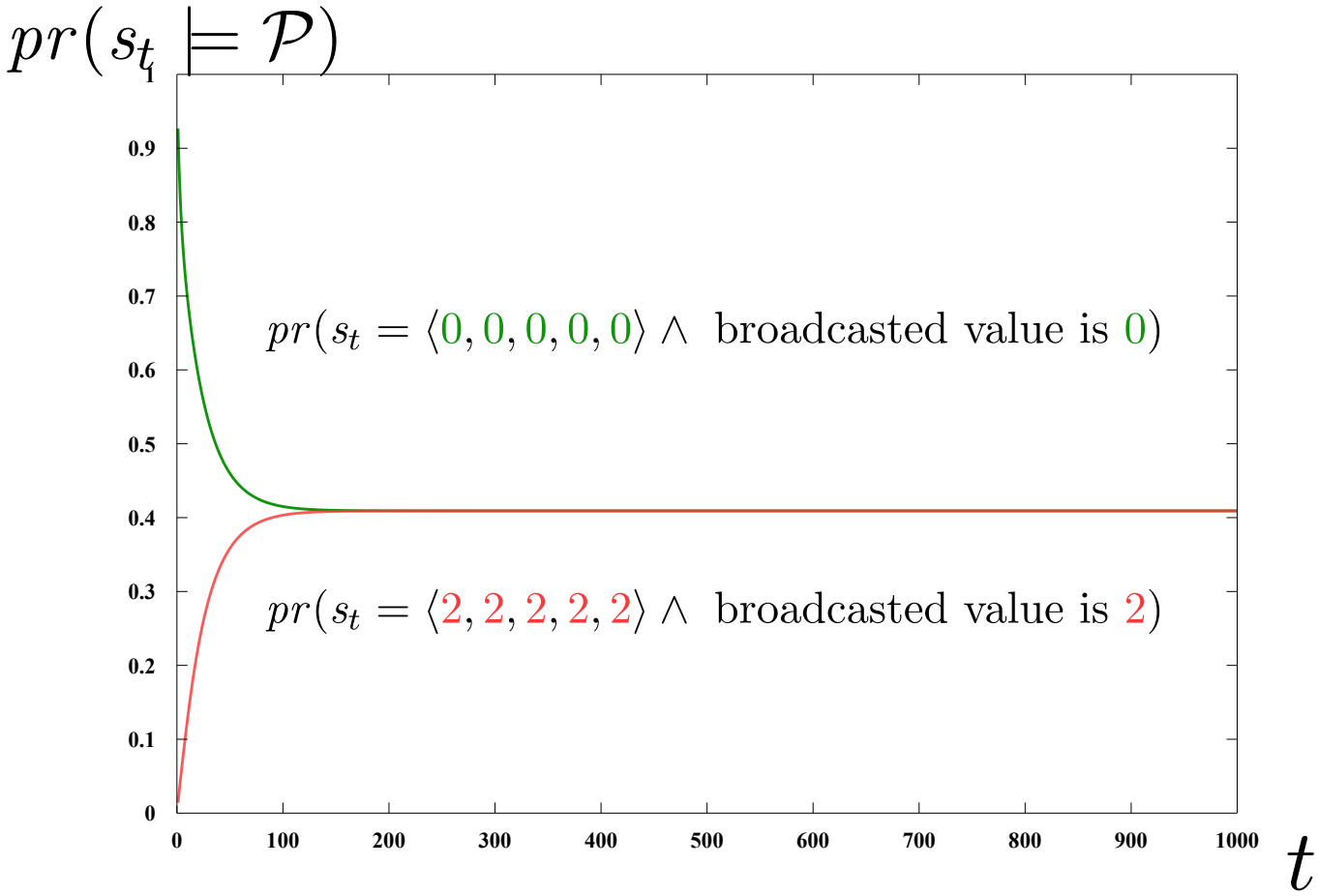


Figure: Result of the WSN example – converging consistency and inertia

Decomposition Schema

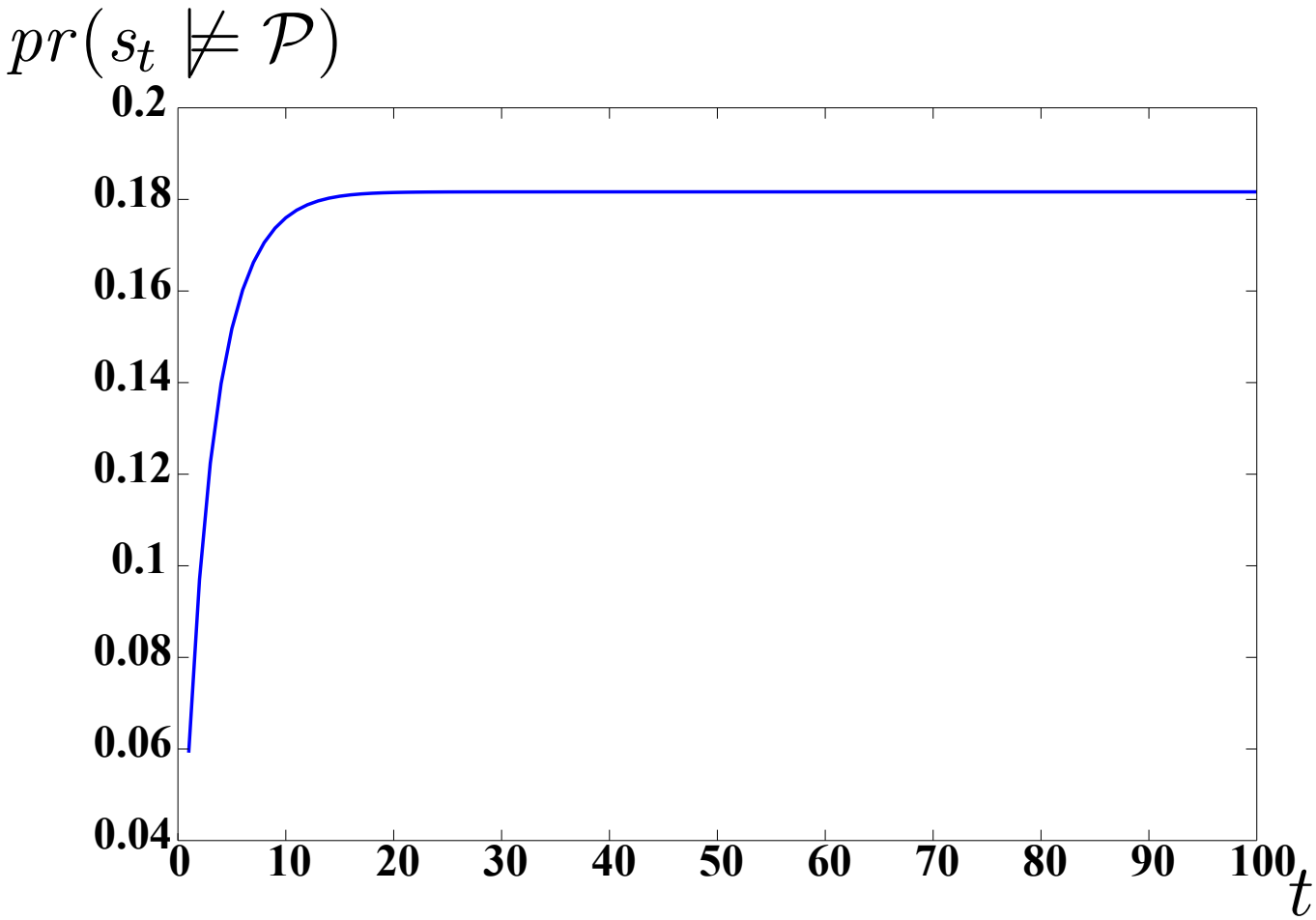


Figure: Result of the WSN example – converging consistency and inertia

- **scope**: computing fault tolerance and dependability measures of distributed systems
- **method**: decomposition and local lumping to cope with state space explosion
- **focus**: semi-hierarchic systems, i.e. globally hierarchic systems containing local cycles
- **result**: not only residential processes sub-systems may be cyclic dependent, but also processes within overlaps

- **scope**: computing fault tolerance and dependability measures of distributed systems
- **method**: decomposition and local lumping to cope with state space explosion
- **focus**: semi-hierarchic systems, i.e. globally hierarchic systems containing local cycles
- **result**: not only residential processes sub-systems may be cyclic dependent, but also processes within overlaps

- **scope**: computing fault tolerance and dependability measures of distributed systems
- **method**: decomposition and local lumping to cope with state space explosion
- **focus**: semi-hierarchic systems, i.e. globally hierarchic systems containing local cycles
- **result**: not only residential processes sub-systems may be cyclic dependent, but also processes within overlaps

- **scope**: computing fault tolerance and dependability measures of distributed systems
- **method**: decomposition and local lumping to cope with state space explosion
- **focus**: semi-hierarchic systems, i.e. globally hierarchic systems containing local cycles
- **result**: not only residential processes sub-systems may be cyclic dependent, but also processes within overlaps

This closes a cycle of eight papers that resulted in my dissertation. Five of which were published within AINA context.

THANK YOU AINA Committees, especially Makoto Takizawa, Leonard Barolli and Fatos Xhafa!

Questions?