Markov Chains & Functional Safety

Monika Heiner and Martin Schwarick

Brandenburg University of Technology Cottbus (BTU)
– Data Structures and Software Dependability –

Philotech Academy
October 17, 2012

Safety Assessment Methods

Aerospace Recommended Practice standard (ARP 4761)

- Fault Tree Analysis (FTA)
- Markov Analysis (MA)

"*MA calculates the probability of the system in various states as function of time.*

—

** A state in the model represents the system status as a function of both the fault-tree and faulty components and the system redundancy.*
** A transition from one state to another occurs at a given transition rate, which reflects component failure rates and redundancy.*
** A system changes state due to various events such as component failure, reconfiguration after detection of a failure, completion of repair, etc. . . .*"

[ARP 4761, p.24]

Markov Analysis

Basic terms of ARP 4761, Appendix F

- Markov chains, properties:
  - stiff
  - homogeneous
  - ergodic
- states, transitions, rates, probability
- extended stochastic Petri nets (ESPN)

Markov Analysis

Questions to be answered:

What are Markov chains?

What can I do with Markov chains?

Where do they come from?

Markov Analysis

Questions to be answered:

What are Markov chains?

What can I do with Markov chains?

Where do they come from?

Markov Analysis

Questions to be answered:

What are Markov chains?

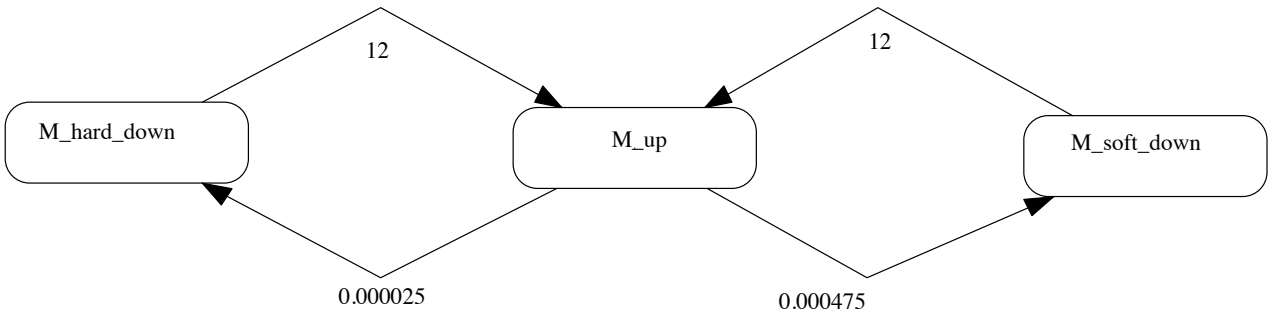
What can I do with Markov chains?

Where do they come from?

Markov Analysis

What are Markov chains?

directed graphs modelling
the **states** of a system,
the **state transitions**, and
the **rates** at which state transitions take place



Markov Analysis (MA)

What can I do with Markov chains?

- probability distributions
  - **transient behaviour**

$$\pi(0.1) = \begin{pmatrix} 2.766025533491E-05 \\ 9.999577740581E-01 \\ 1.455802912363E-05 \end{pmatrix}$$

  - **steady state behaviour**

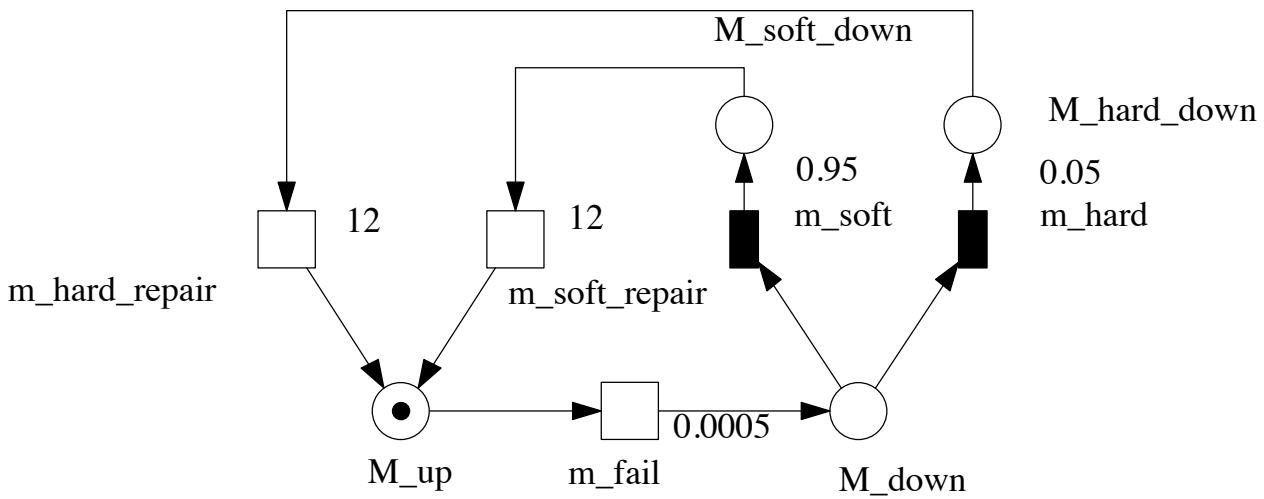
$$\pi = \begin{pmatrix} 3.958096646054E-05 \\ 9.999395869588E-01 \\ 2.083207472830E-05 \end{pmatrix}$$

- **performance and dependability analysis**

Markov Analysis (MA)

Where do they come from?

(generalized) stochastic Petri nets



Outline

Qualitative Petri Nets

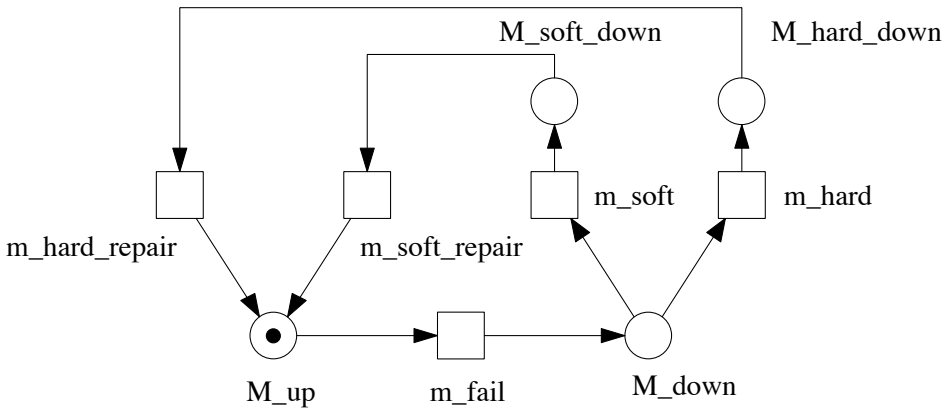
Stochastic Petri Nets

Markov Chains

Tool Support

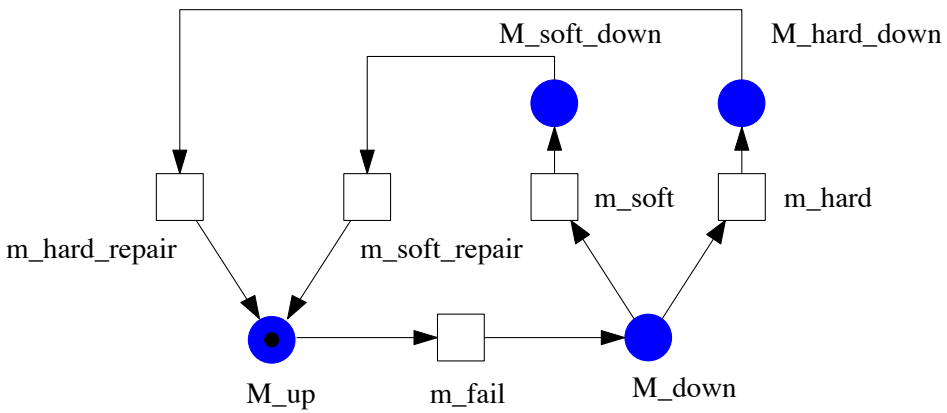
Summary

Qualitative Petri Nets (QPN)



$$\mathcal{QPN} = [P, T, V, s_0]$$

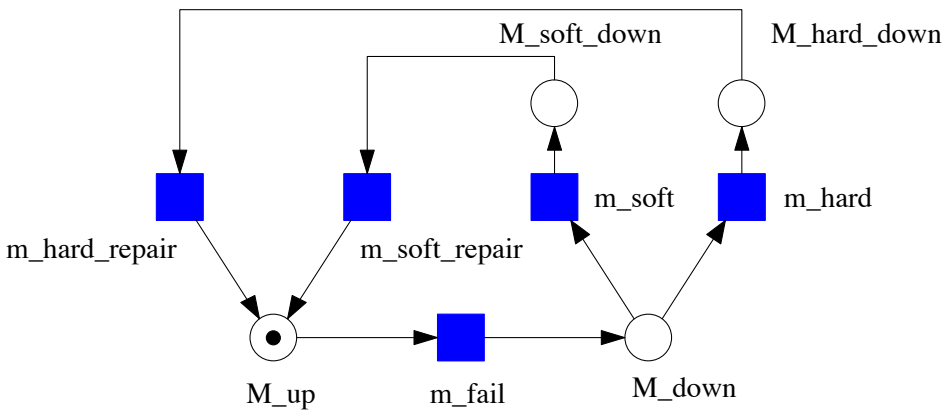
Qualitative Petri Nets



$$\mathcal{QPN} = [P, T, V, s_0]$$

- P, the finite set of places
- T, the finite set of transitions
- $V : P \times T \cup T \times P \to \mathbb{N}$, the function defining the weighted arcs
- s_0, the initial state with $s : P \to \mathbb{N}$

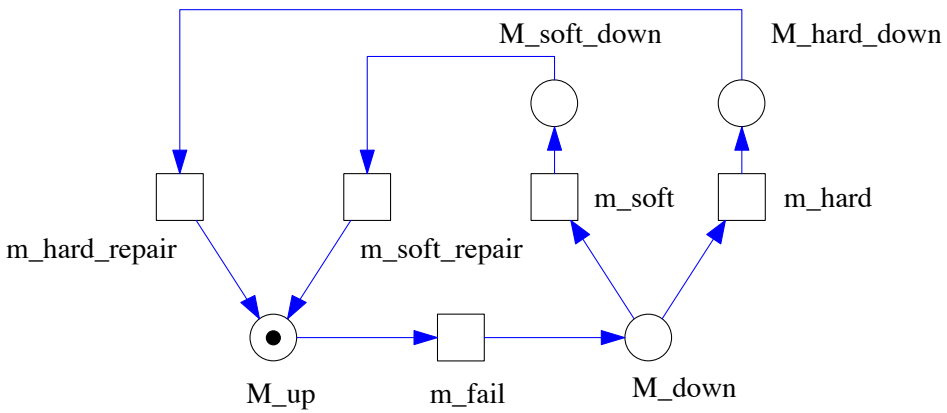
Qualitative Petri Nets



$\mathcal{QPN} = [P, T, V, s_0]$

- P, the finite set of places
- T, the finite set of transitions
- $V : P \times T \cup T \times P \to \mathbb{N}$, the function defining the weighted arcs
- s_0, the initial state with $s : P \to \mathbb{N}$

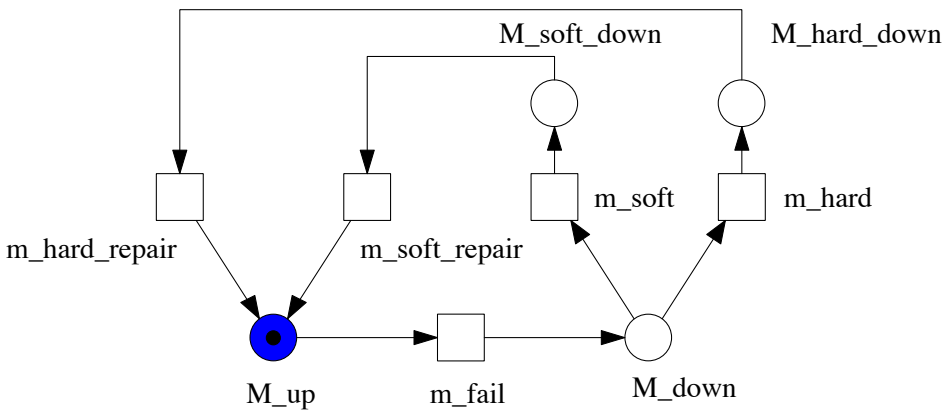
Qualitative Petri Nets



$\mathcal{QPN} = [P, T, V, s_0]$

- P, the finite set of places
- T, the finite set of transitions
- $V : P \times T \cup T \times P \to \mathbb{N}$, the function defining the weighted arcs
- s_0, the initial state with $s : P \to \mathbb{N}$

Qualitative Petri Nets



$\mathcal{QPN} = [P, T, V, s_0]$

- P, the finite set of places
- T, the finite set of transitions
- $V : P \times T \cup T \times P \to \mathbb{N}$, the function defining the weighted arcs
- s_0, the initial state with $s : P \to \mathbb{N}$

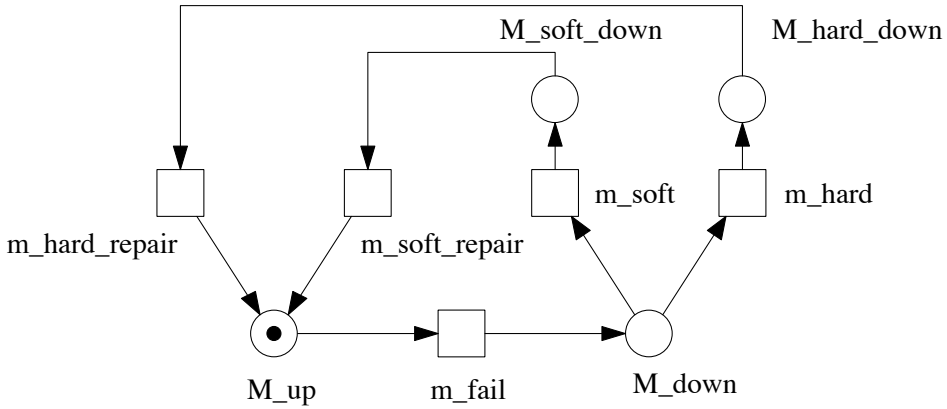
Qualitative Petri Nets

Semantics:

- state changes are caused by the firing of transitions
- firing rule:
  - enabledness
  - token consumption on pre-places, production on post-places
- exhaustive firing of transitions produces the state space
- reachability graph $\mathcal{RG} = [S, A, L, s_0]$ with
  - S, the set of reachable states (nodes)
  - A, the set of state transitions (arcs)
  - $L : S \to AP$, a labelling function
  - s_0, the initial state

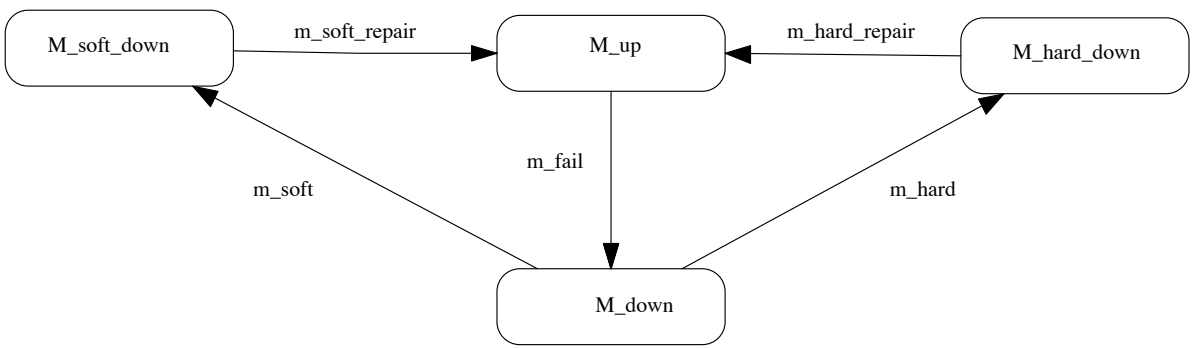
Qualitative Petri Nets – Reachability Graph

$\mathcal{QPN}$



RG construction

$\mathcal{RG}$



Qualitative Petri Nets – Behavioural Properties

- **boundedness**
  finite state space, upper bound for number of tokens on each place
- **reversibility**
  it is always possible to return to the initial state
- **weak liveness**
  it is never possible that no transition is enabled
- **liveness**
  all transitions have always the chance to become enabled

Example – Google Replicated File System (GRFS)

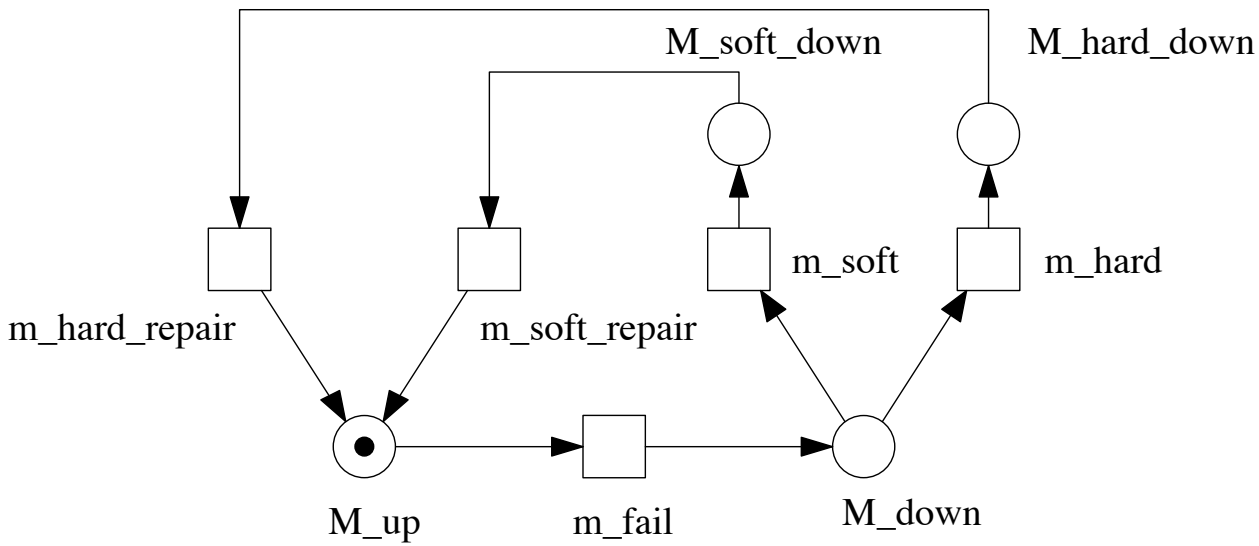
Basic facts:

- file is a composition of chunks
- several replicas for each chunk
- replicas are stored on chunk servers
- a master
  - keeps account of chunks and chunk servers
  - instantiates replica generation
  - sets up connection between clients and a chunk server

The Petri net by L. Cloth and B. Haverkort [CH05] models the life cycle of a single chunk.

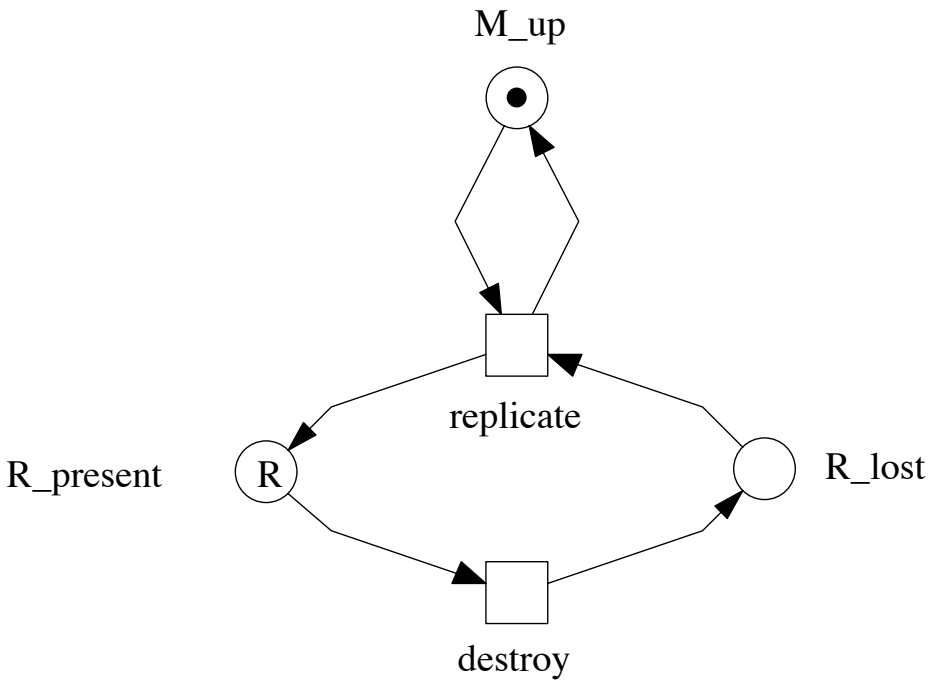
GRFS - Master

- is either up or down
- failures are due to
  - software problems - restart
  - hardware problems - repair



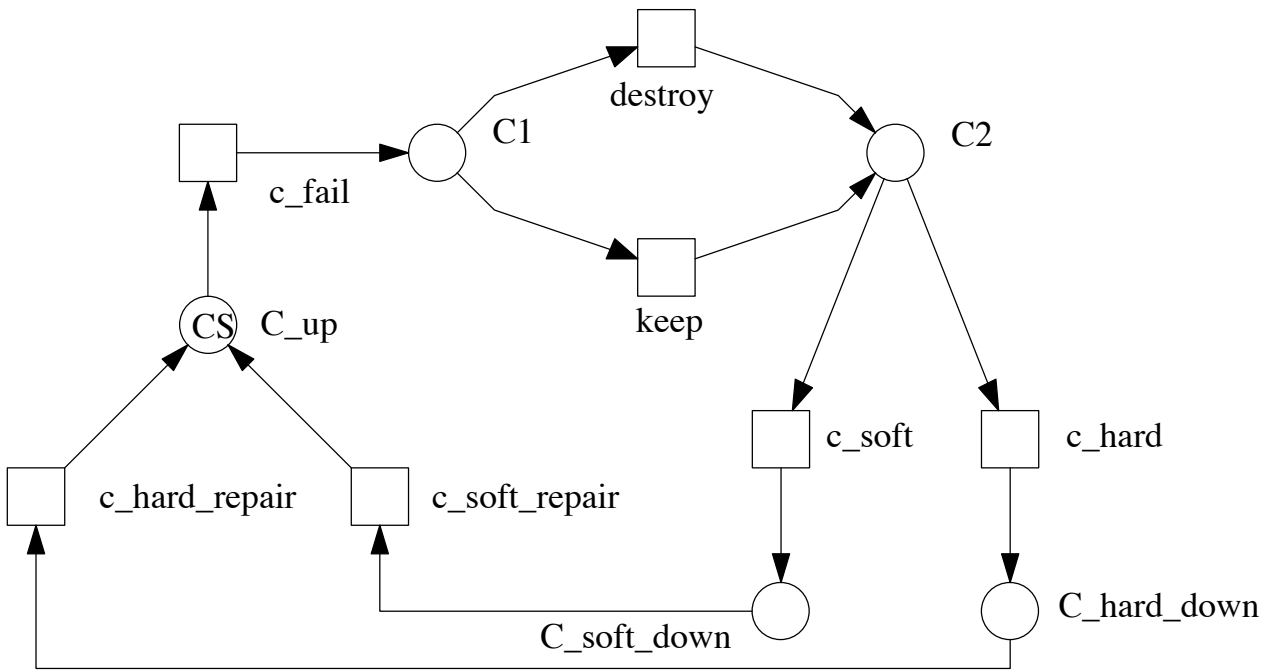
GRFS - Replicas

- a chunk can have R replicas
- replica generation is instantiated by the master

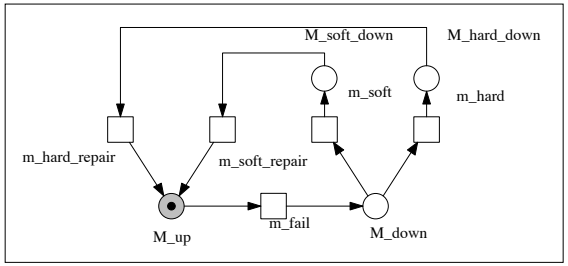


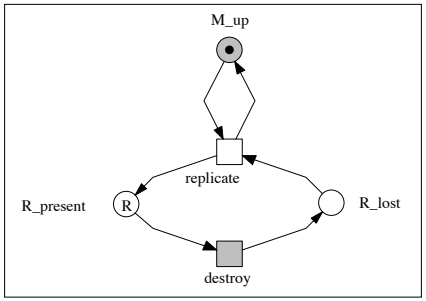
GRFS - Chunk Server

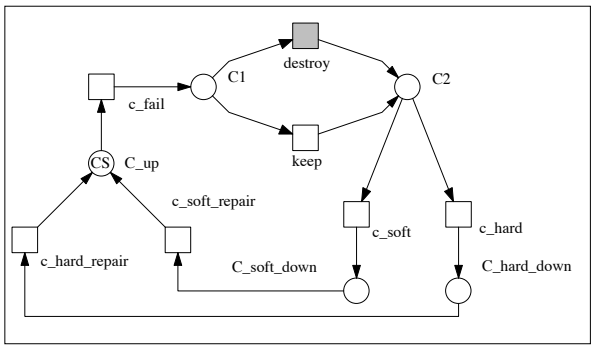
- there are *CS* chunk servers
- a chunk server may fail similar to the master
- if a chunk server fails, the investigated chunk either
  - gets lost (destroy), or
  - resides on a different chunk server (keep)
- number of chunk servers affects rates

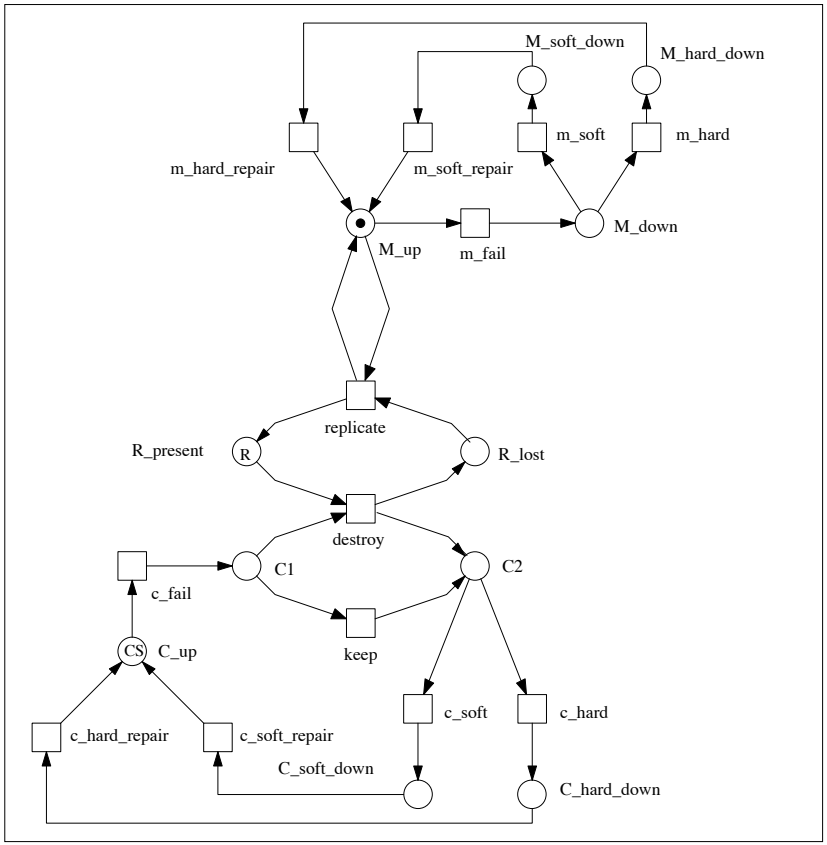


GRFS - Putting all together









RG-based Analysis

Reachability graph size for different numbers of

- chunk server (CS) and
- possible replicas (R)

R	3		4		5	
CS	|S|	|A|	|S|	|A|	|S|	|A|
20	161,604	1,113,886	196,488	1,362,307	228,312	1,588,407
40	2,139,204	15,831,252	2,650,988	19,741,338	3,148,712	23,544,753
80	30,742,404	236,938,258	38,333,988	297,114,375	45,865,512	356,826,720

|S| – number of states; |A| – number of state transitions;

In any case, the Petri nets are

- bounded
- reversible
- life

Advanced Analysis - Survivability

is the ability of a system to **recover** predefined **service levels** (in a **timely manner**) after the occurrence of **disasters** [CH05].

How can theses terms be formalized?

- **recoverability**
  existence of paths
  from disaster states to states of a required service level
- **service level n**
  master is working **and** there are at least n replicas

$$service\_level\_n \equiv M\_up = 1 \textbf{ and } R\_present \geq n$$

Survivability - Specifying Disasters

Failures

- either of software or hardware components
- of the master $\rightarrow M\_up = 0$
  - software failures $\rightarrow M\_soft\_down = 1$
  - hardware failures $\rightarrow M\_hard\_down = 1$
- of the chunk servers (software)

light	25-50%	$C\_soft\_down \in [\frac{CS}{2}, \frac{CS}{4})$ **and** $C\_hard\_down = 0$
medium	50-75%	$C\_soft\_down \in [\frac{CS}{4}, \frac{3}{4}CS)$ **and** $C\_hard\_down = 0$
severe	75 -100%	$C\_soft\_down \in [\frac{3}{4}CS, CS)$ **and** $C\_hard\_down = 0$

e.g. a *light software disaster* is characterized by

$M\_soft\_down = 1$ **and**

$C\_soft\_down \in [CS/2, CS/4)$ **and** $C\_hard\_down = 0$

GRFS - CTL Model Checking

- Is a light software disaster possible?

  EF [*light_software_disaster*]

- In the case of a light software disaster, is it possible to recover the system to service level *n*?

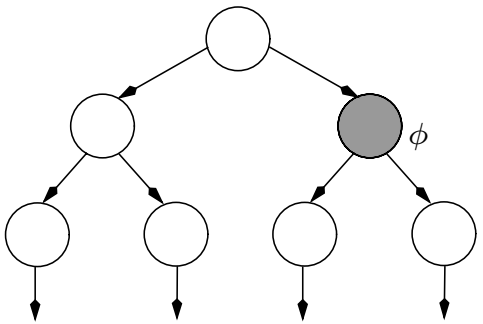
  AG [*light_software_disaster* $\Rightarrow$ **EF** [*service_level_n*]]

- In the case of a light software disaster, is it ensured that the system will be recovered to service level *n*?

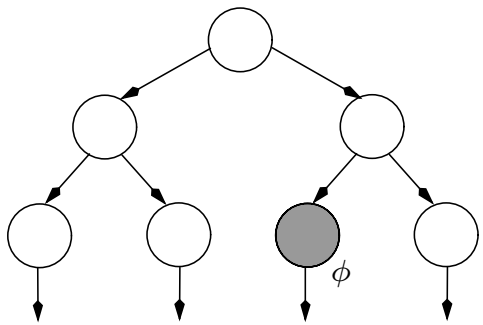
  AG [*light_software_disaster* $\Rightarrow$ **AF** [*service_level_n*]]

Computation Tree Logic (CTL)

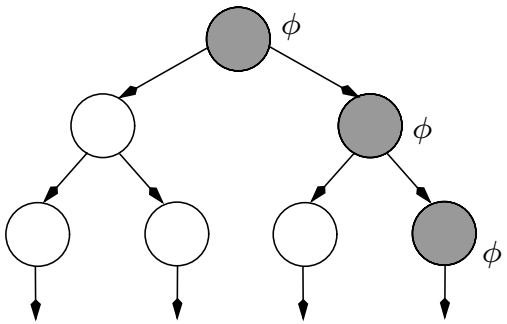
qualitative reasoning on the existence/reachability of states/paths



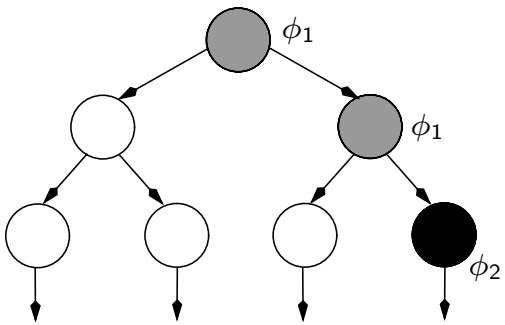
EXϕ



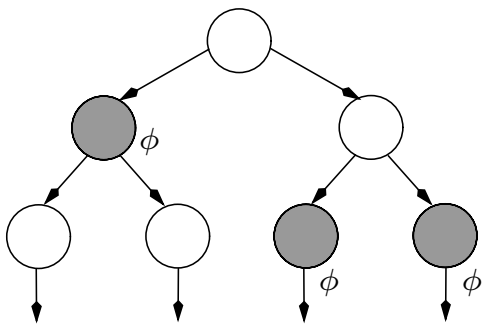
EFϕ



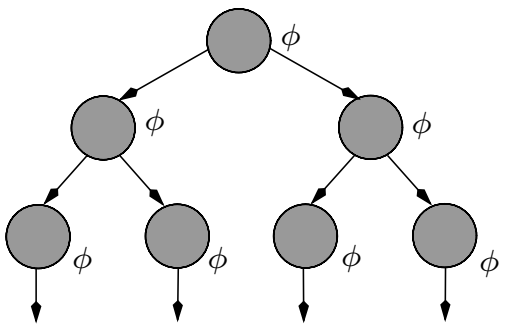
EGϕ



E$[\phi_1 \mathbf{U} \phi_2]$



AFϕ



AGϕ

Model checking [BK08]

automatic procedure to determine for a model
the fulfillment of a given property specification

- model specification, e.g.
  - QPN
  - SPN
  - ...
- property specification in temporal propositional logics, e.g.
  - Computation Tree Logic (CTL)
  - Linear Temporal Logic (LTL)
  - Continuous Stochastic Logic (CSL)
  - Continuous Stochastic Reward Logic (CSRL)
  - ...

How to investigate
in a timely manner?

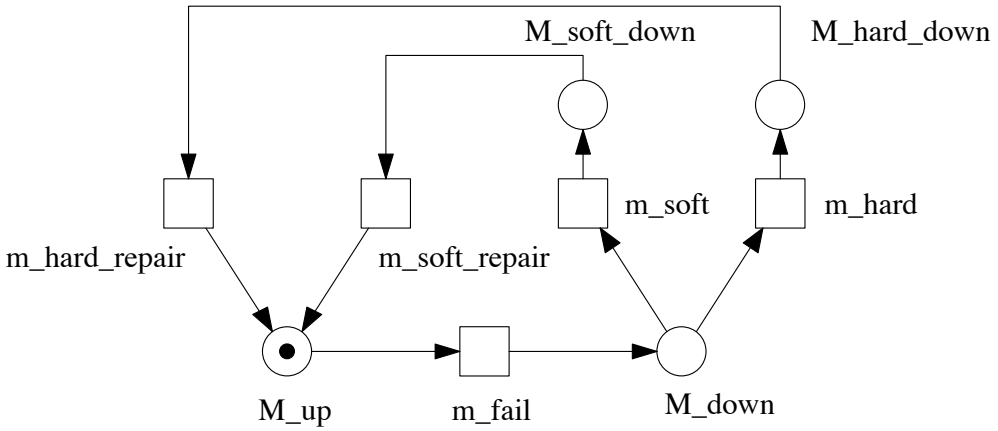
Stochastic Petri Nets [BK02]

Introduction of time by defining **transition firing rates**

- average of observable firings of a transition per time unit and state
- time spent in states (sojourn time δ) is a negative exponentially distributed random variable if past does not matter (memoryless/Markov property)

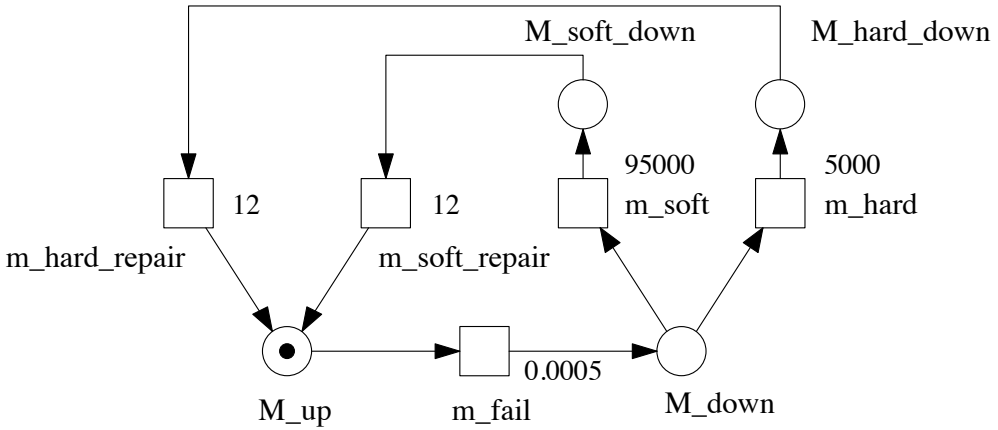
Stochastic Petri Nets

$\mathcal{QPN}$



firing rates

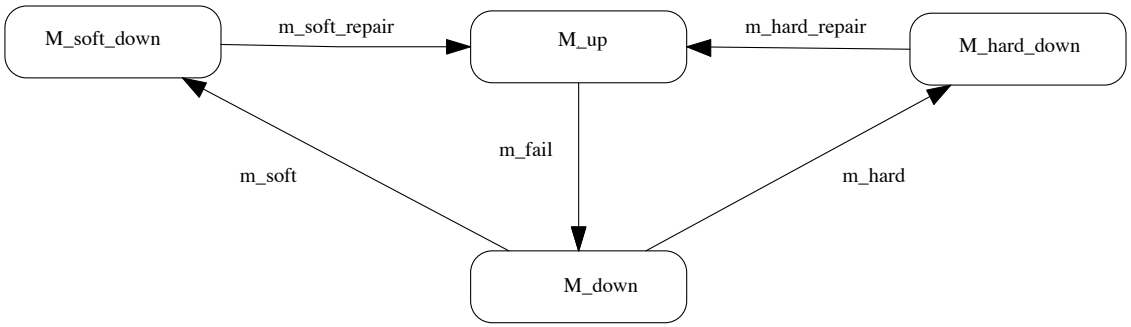
$\mathcal{SPN}$



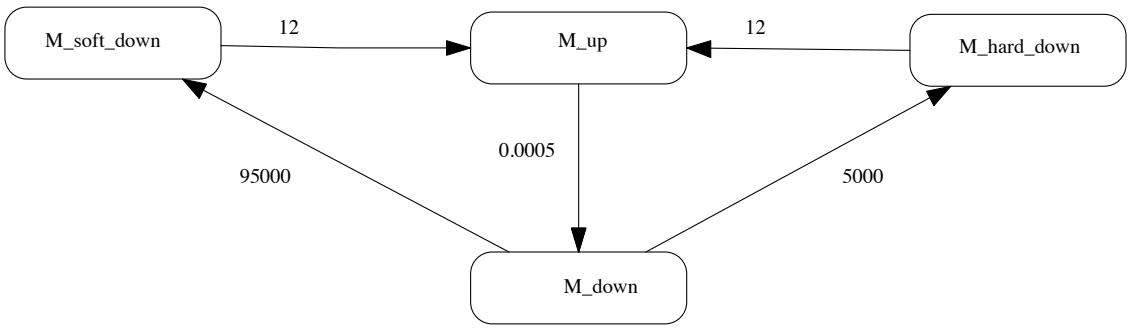
Stochastic Petri Nets

Semantics is a Continuous-time Markov Chain (CTMC)
$\rightarrow$ reachability graph augmented by firing rates

$\mathcal{RG}$
$(\mathcal{QPN})$



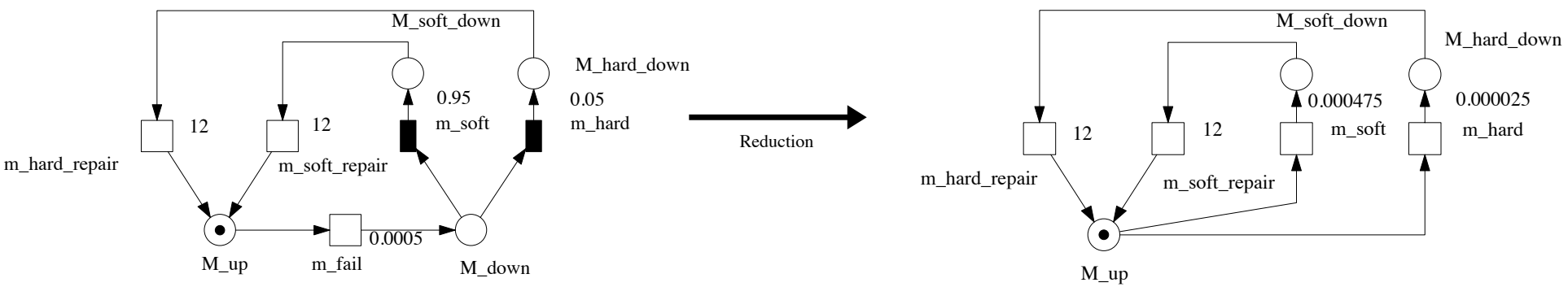
$\mathcal{CTMC}$
$(\mathcal{SPN})$



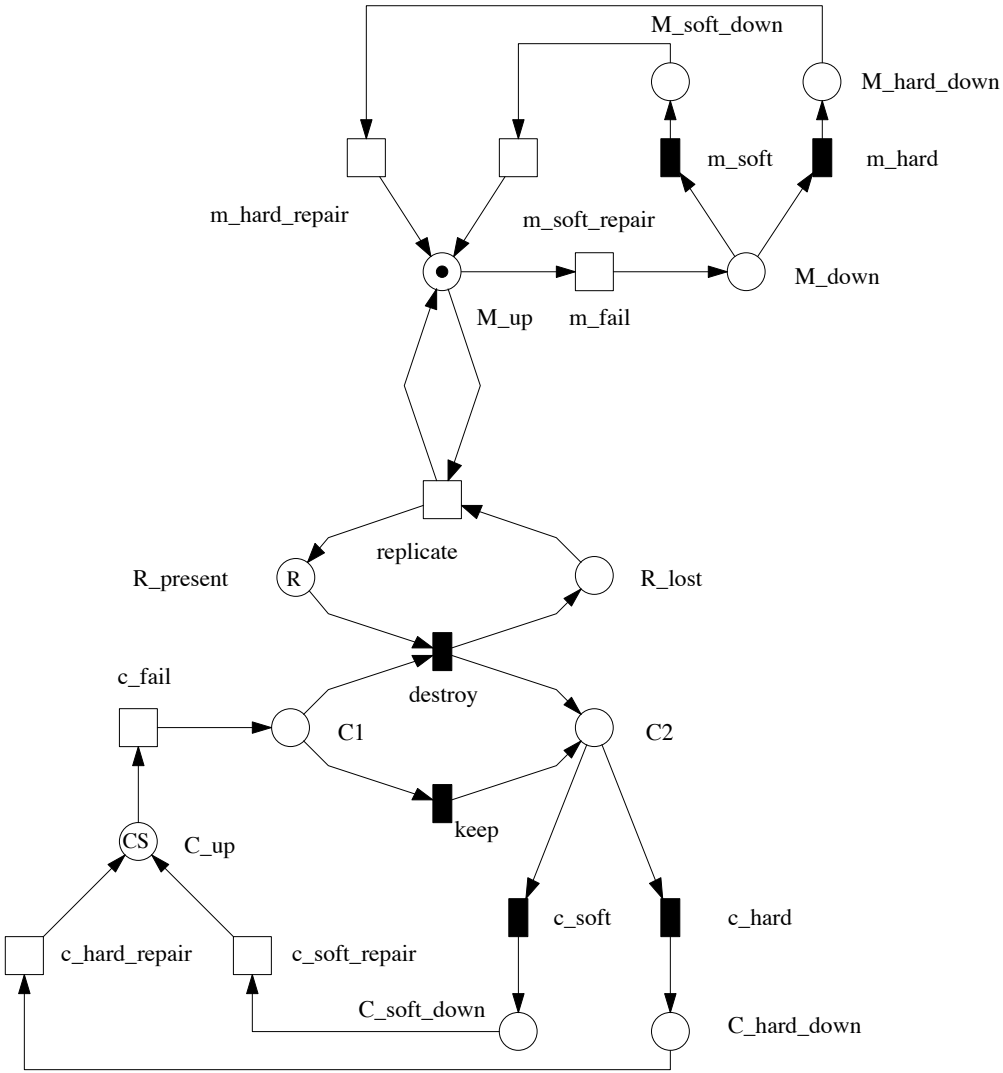
Generalized SPN (GSPN) [MBC+95]

- immediate transitions with zero delay
- weights to treat conflicts
- reduction to SPN possible
- semantics is still a CTMC

$\mathcal{GSPN}$ $\mathcal{SPN}$



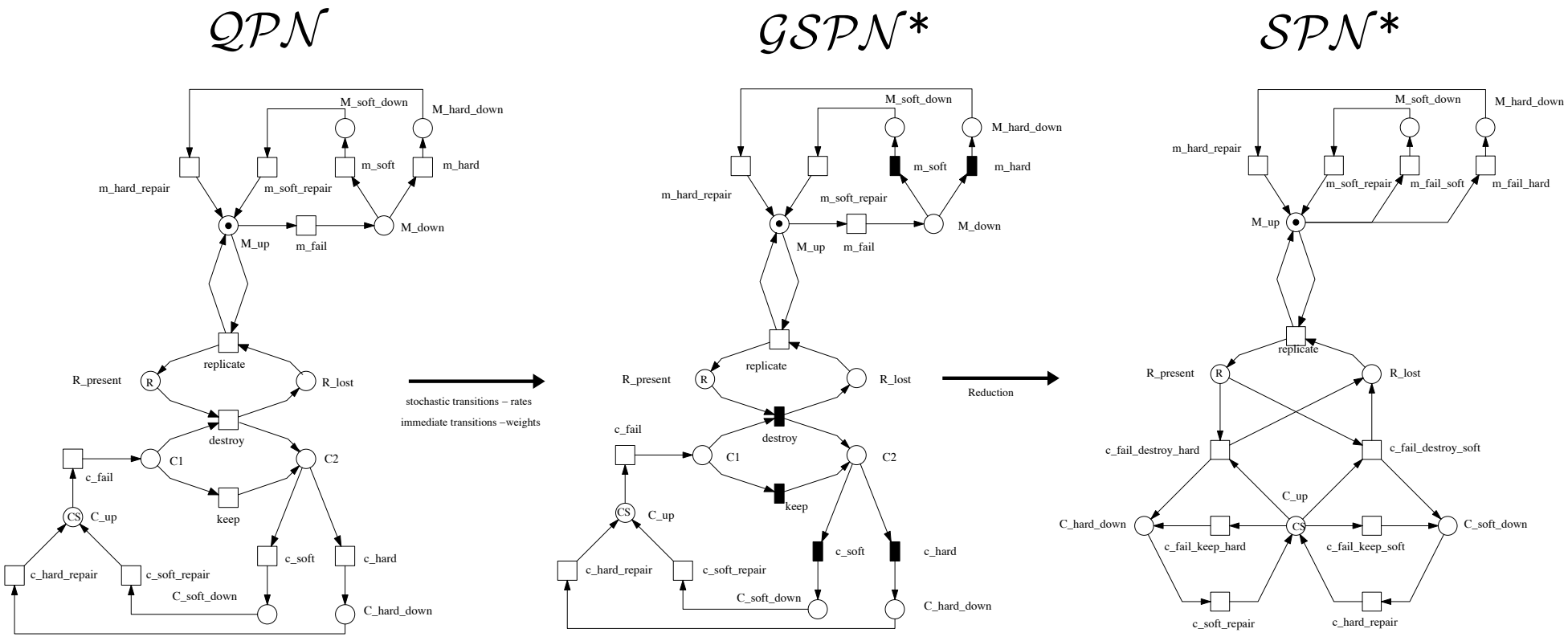
GSPN for Google Replicated File System



stochastic	rates
m_fail	0.0005
m_soft_repair	12
m_hard_repair	12
c_fail	0.05
c_soft_repair	$C\_soft\_down \cdot 12$
c_hard_repair	1.0
replicate	R_present $>$ 0: 20.0 R_present $=$ 0: 2.0

immediate	weights
m_soft	0.95
m_hard	0.05
c_soft	0.95
c_hard	0.05
destroy	$\frac{R\_present}{C\_up}$
keep	$1 - \frac{R\_present}{C\_up}$

Petri Net Modelling



* rates and weights have been omitted for the sake of readability

Size of the CTMC

$$\mathcal{RG}(\mathcal{QPN}) \equiv \mathcal{CTMC}(\mathcal{GSPN})$$

R	3		4		5	
CS	|S|	|A|	|S|	|A|	|S|	|A|
20	161,604	1,113,886	196,488	1,362,307	228,312	1,588,407
40	2,139,204	15,831,252	2,650,988	19,741,338	3,148,712	23,544,753
80	30,742,404	236,938,258	38,333,988	297,114,375	45,865,512	356,826,720

$$\mathcal{CTMC}(\mathcal{SPN})$$

R	3		4		5	
CS	|S|	|A|	|S|	|A|	|S|	|A|
20	2,406	15,323	2,865	18,485	3,273	21,285
40	9,606	63,614	11,715	78,636	13,713	92,856
80	38,406	260,885	47,415	326,028	56,193	389,488

|S| – number of states; |A| – number of state transitions

Continuous-time Markov Chains (CTMC) [Ste94]

$$C = [S, \mathbf{R}, L, s_0]$$

- S – finite set of states
- $\mathbf{R}$ – transition rate relation (usually a $|S| \times |S|$ matrix)
- $L : S \to AP$ – a labelling function
- s_0 – the initial state

CTMC - Basic Measures

- exit rate

$$E(s) = \sum_{s \neq s'} \mathbf{R}(s, s')$$

- probability to leave s within τ time units

$$Pr\{\delta_s < \tau\} = 1 - e^{-E(s)\cdot\tau}$$

- probability of a given state transition $s \to s'$

$$Pr\{s \to s'\} = \mathbf{P}(s, s') = \mathbf{R}(s, s')/E(s)$$

  within τ time units is

$$Pr\{s \xrightarrow{\delta_s < \tau} s'\} = \mathbf{P}(s, s') \cdot (1 - e^{-E(s)\cdot\tau})$$

CTMC - Standard Measures

Let π, Π be state vectors.

- transient probabilities $\pi(\tau)$
  probability distribution at time instant τ
- steady state probabilities π
  $\lim_{\tau \to \infty} \pi(\tau)$ - probability distribution on the long run
- cumulative state probabilities $\Pi(\tau)$
  $\int_0^\tau \pi(u)du$

CTMC - Dependability Measures

Let be

- $S_{up} \subseteq S$
  – the set of states providing expected service
- $S_{down} \subseteq S$
  – the set of states **not** providing expected service
- $S_{down} \cap S_{up} = \emptyset$
- ap_{up}
  – atomic proposition such that $ap_{up} \in L(s) \Leftrightarrow s \in S_{up}$
- ap_{down}
  – atomic proposition such that $ap_{down} \in L(s) \Leftrightarrow s \in S_{down}$

CTMC - Availability

- probability that the system is *up* at time τ

$$\sum_{s \in S_{up}} \pi(\tau)$$

- in Continuous Stochastic Logic (CSL)

$$\mathcal{P}_{=?} \left[\mathbf{F}^{[\tau,\tau]} \; ap_{up} \right]$$

CTMC - Reliability

- probability that the system is continuously *up* until time τ
- in Continuous Stochastic Logic (CSL)

$$\mathcal{P}_{=?} \left[\mathbf{G}^{[0,\tau]} \; ap_{up} \right]$$

- transform the CTMC by making all S_{down} states absorbing

$$\sum_{s \in S_{up}} \pi(\tau) \text{ with } \forall s \in S_{down} : E(s) = 0$$

CTMC - Survivability

- probability that the system will reach
  for each *down* state an *up* state within τ time units
- in Continuous Stochastic Logic (CSL)

$$\mathcal{P}_{=?}\ [\ \mathbf{F}^{[0,\tau]}\, ap_{up}\ ]\ \{ap_{down}\}$$

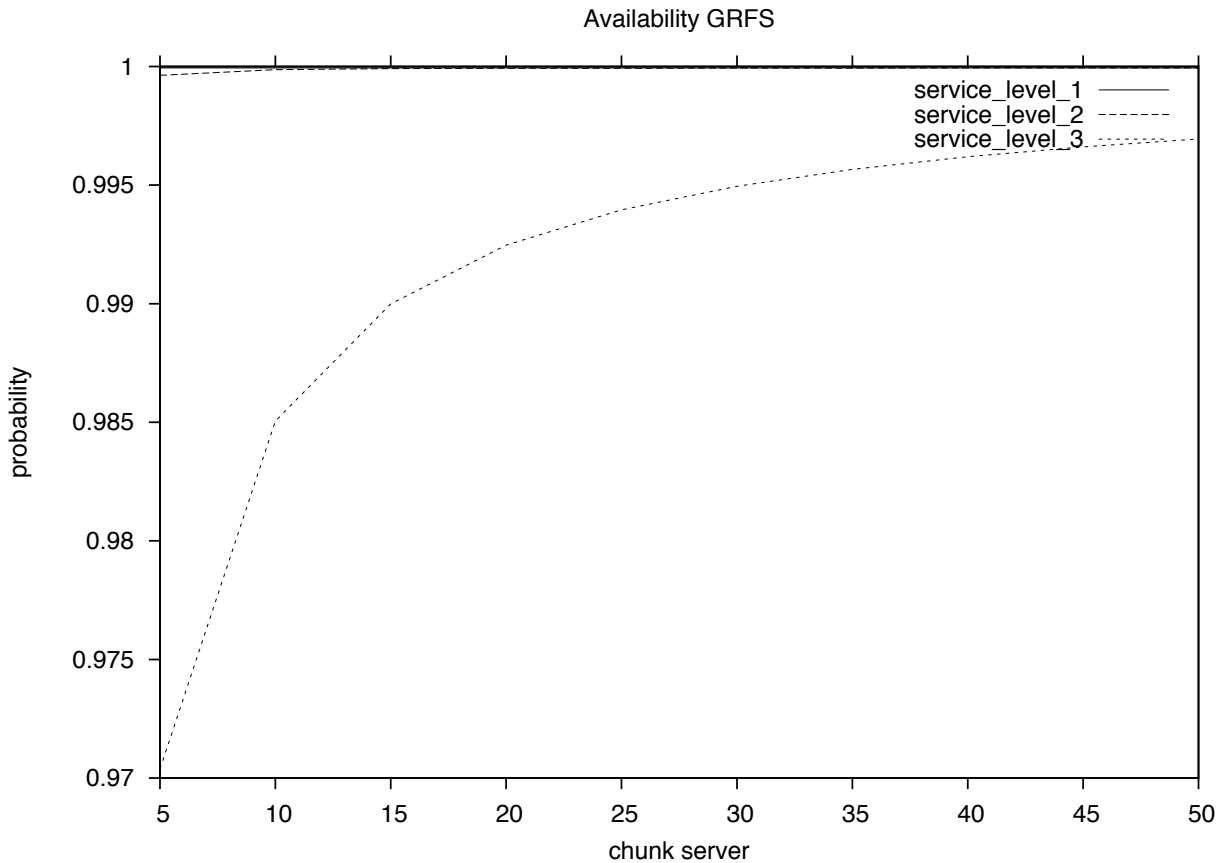
- transform the CTMC by making all S_{up} states absorbing

$$\forall s \in S_{down} \text{ as initial state } : \sum_{s' \in S_{up}} \pi(\tau) \text{ with } \forall s' \in S_{up} : E(s') = 0$$

GRFS - Availability

What is the probability that the system is at time point τ at service level n?

$$\mathcal{P}_{=?}\ [\ \mathbf{F}^{[\tau,\tau]} service\_level\_n\ ]$$

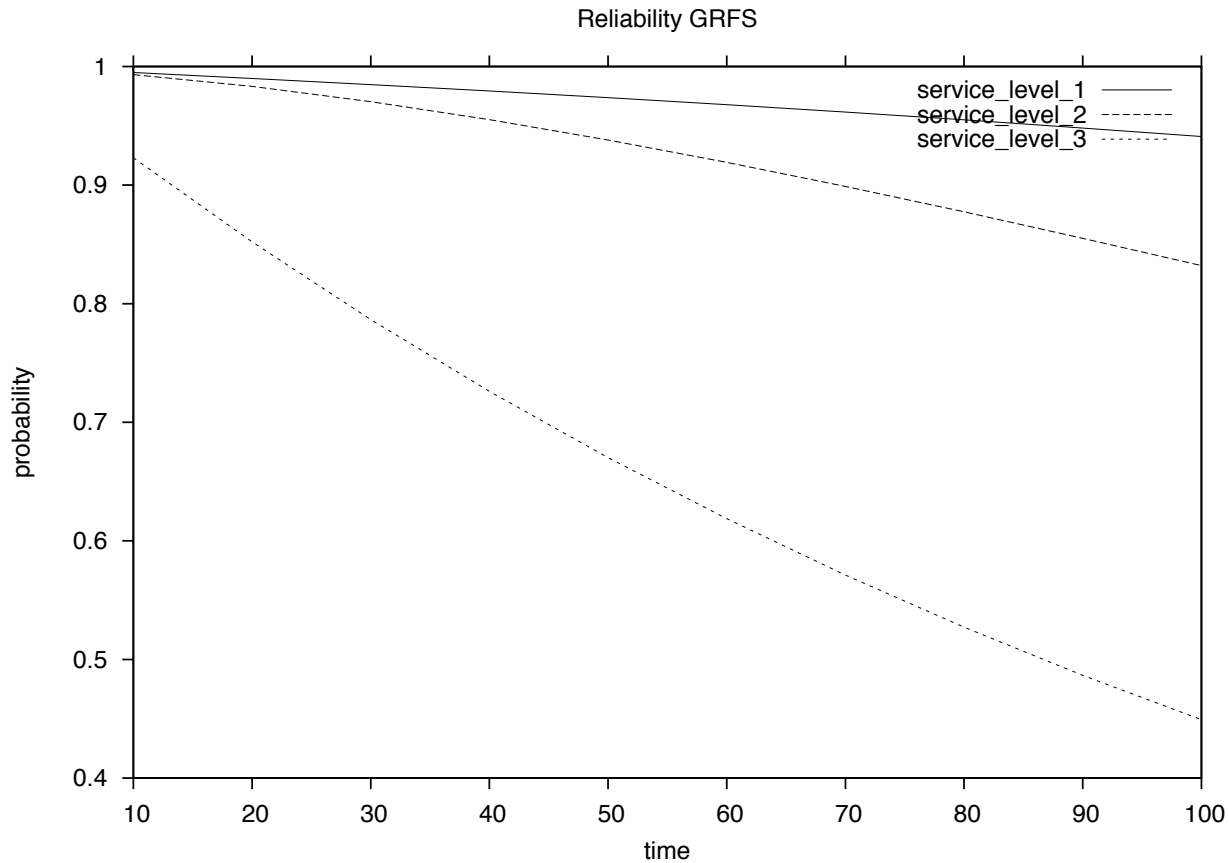


$R = 3, \tau = 1$

GRFS - Reliability

What is the probability that the system
remains the first τ time units continuously at service level n?

$$\mathcal{P}_{=?}\ [\ \mathbf{G}^{[0,\tau]} service\_level\_n\ ]$$

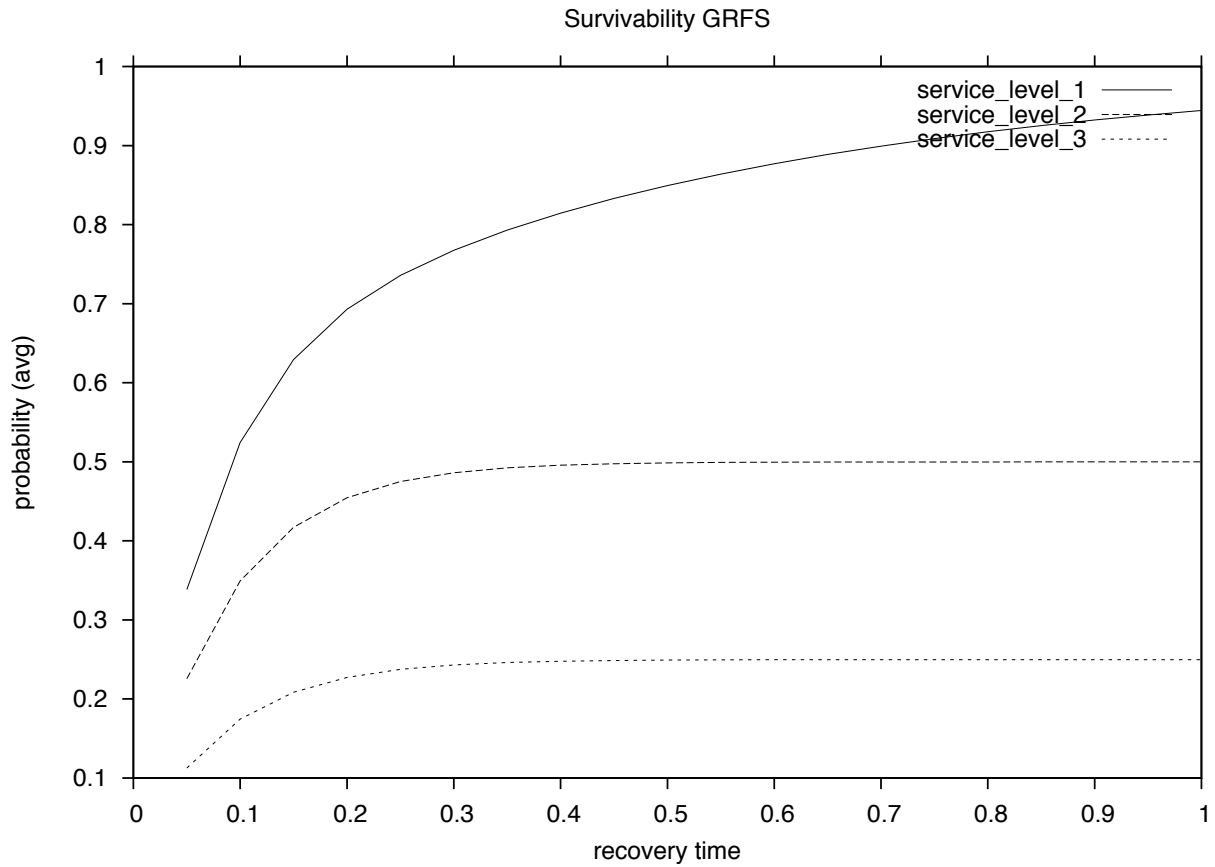


$R = 3, CS = 20$

GRFS - Survivability

What is the probability for states representing a light software disaster that the system will be recovered within τ time units?

$$\mathcal{P}_{=?}\ [\ \mathbf{F}^{[0,\tau]} service\_level\_n\ ]\{light\_software\_disaster\}$$



$R = 3, CS = 20$

CSL - Syntax

state formulas

$$\phi ::= \mathit{true} \mid \mathit{ap} \mid \neg\phi \mid \phi \vee \phi \mid \mathcal{P}_{\bowtie p}[\psi] \mid \mathcal{P}_{=?}[\psi] \mid \mathcal{S}_{\bowtie p}[\phi]$$

path formulas

$$\psi ::= \mathbf{X}^I\phi \mid \mathbf{F}^I\phi \mid \mathbf{G}^I\phi \mid \phi\,\mathbf{U}^I\phi$$

with
$\mathit{ap} \in \mathit{AP}$, $\bowtie \in \{<, \leq, \geq, >\}$, $p \in [0,1]$, and $I \subseteq \mathbb{R}_+$

CSL - Semantics

state formulas:

- $s \models ap \Leftrightarrow ap \in L(s)$
- $s \models \neg\Phi \Leftrightarrow s \not\models \Phi$
- $s \models \Phi \vee \Psi \Leftrightarrow s \models \Phi \vee s \models \Psi$
- $s \models \mathcal{P}_{\bowtie p}[\psi] \Leftrightarrow Prob_s^M(\psi) \bowtie p$
- $s \models \mathcal{S}_{\bowtie p}[\psi] \Leftrightarrow Prob_s^M(\psi) \bowtie p,$

path formulas:

- $\sigma \models \mathbf{X}^I\Phi \Leftrightarrow |\sigma| \geq 1 \wedge \tau_0 \in I \wedge \sigma[1] \models \Phi$
- $\sigma \models \mathbf{F}^I\Phi \Leftrightarrow \exists\tau \in I : \sigma(\tau) \models \Phi$
- $\sigma \models \mathbf{G}^I\Phi \Leftrightarrow \forall\tau \in I : \sigma(\tau) \models \Phi$
- $\sigma \models \Phi\mathbf{U}^I\Psi \Leftrightarrow \exists\tau \in I : \sigma(\tau) \models \Psi \wedge \forall\tau' < \tau : \sigma(\tau') \models \Phi$

Beyond CTMCs - Rewards

- reward functions for states

$$\varrho : S \to \mathbb{R}^+$$

- can be interpreted as costs
- CTMC + reward function $\to$ Markov Reward Model (MRM)
- CSL $\to$ CSRL (reward constraints concerning paths)
- Survivability with recovery costs

$$\mathcal{P}_{=?} \left[\mathsf{F}^{[0,\tau]}_{[0,r]} \mathit{service\_level\_n} \right] \{\mathit{light\_software\_disaster}\}$$

CTMC - Expected up-time

- expected time in which the system is *up* within τ time units

$$\sum_{s \in S_{up}} \Pi(\tau)$$

- in Continuous Stochastic Reward Logic (CSRL)

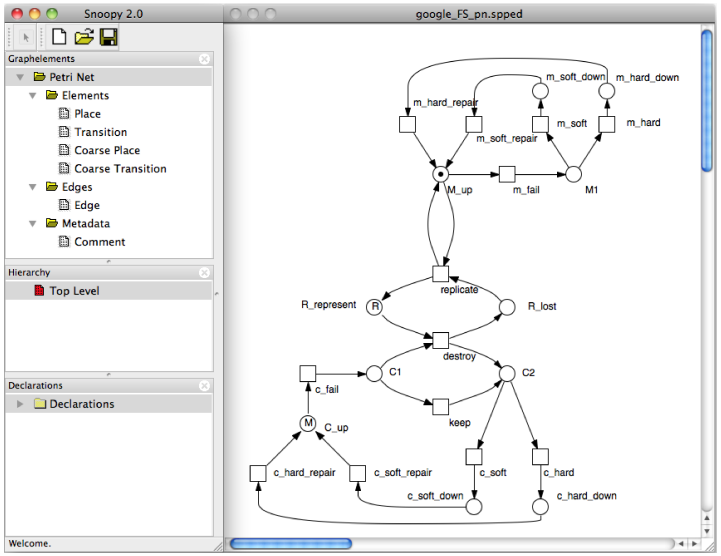
$$\mathcal{R}_{=?}\,[\,\mathsf{C}\ \leq t\,]$$

given the reward function

$$\varrho(s) = \begin{cases} 1 & \text{if } s \in S_{up} \\ 0 & \text{otherwise} \end{cases}$$

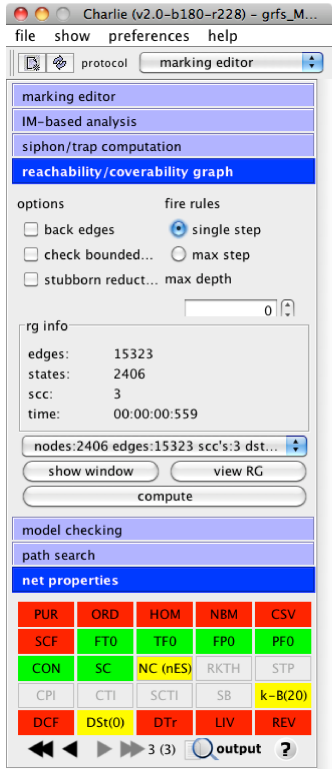
Tool Support

Snoopy [HHL+12]



- modeling/animation
  - QPN, SPN, GSPN
- stochastic simulation

Charlie [Fra09]

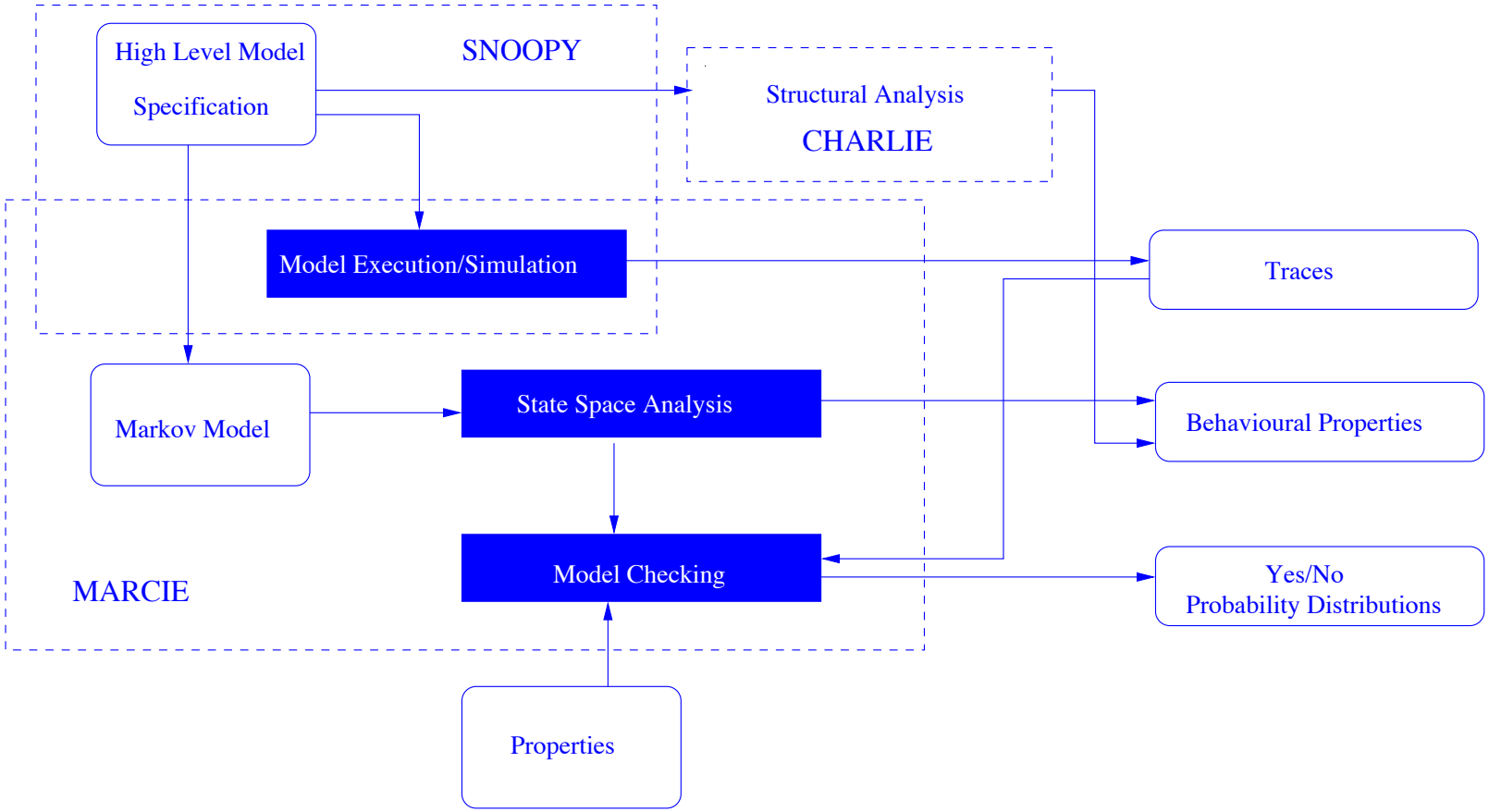


- structural analysis
- RG visualization
- model checking
  - CTL/LTL - RG_{exp}

MARCIE [SRH11]

- standard properties
- model checking
  - CTL - RG_{sym}
  - CS(R)L - $CTMC_{otf}$
  - PLTLc - $CTMC_{sim}$

Tool Support



Related Markov Analysis Tools

Popular tools for symbolic state space analysis & model checking:

- PRISM (CSL) - University of Oxford
  `http://www.prismmodelchecker.org`
- SMART (CTL) - University of California at Riverside
  `http://www.cs.ucr.edu/~ciardo/SMART/`

MARCIE outperforms these tools re

- treatable state space size
- performance

thanks to its multi-threaded (simulative and symbolic) engines [HST09, SH09, ST10, SRH11].

Summary

Basic ingredients

- dependability model of the system to be assessed
- dependability properties of interest
  - good/bad system states
  - patterns for typical properties
- powerfull toolkit
- knowledgeable staff/collaborators
- time/money

INTERESTED IN A CASE STUDY ?

References I

[ARP96] *ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems*.
SAE Inc., 1996.

[BK02] F. Bause and P.S. Kritzinger.
Stochastic Petri Nets.
Vieweg, 2002.

[BK08] Christel Baier and Joost-Pieter Katoen.
Principles of Model Checking (Representation and Mind Series).
The MIT Press, 2008.

[CH05] Lucia Cloth and Boudewijn R. Haverkort.
Model checking for survivability!
In *Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, 2005*, pages 145–154. IEEE, 2005.

[Fra09] Andreas Franzke.
Charlie 2.0 – a multi-threaded petri net analyzer.
Diploma thesis, BTU Cottbus, Dep. of CS, December 2009.

[HHL+12] M Heiner, M Herajy, F Liu, C Rohr, and M Schwarick.
Snoopy - a unifying Petri net tool.
In *Proc. PETRI NETS 2012*, volume 7347 of *LNCS*, pages 398âĂŞ–407. Springer, June 2012.

[HST09] M. Heiner, M. Schwarick, and A. Tovchigrechko.
DSSZ-MC-A Tool for Symbolic Analysis of Extended Petri Nets.
In *Proc. Petri Nets*, pages 323–332. LNCS 5606, Springer, 2009.

[MBC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis.
Modelling with Generalized Stochastic Petri Nets.
Wiley Series in Parallel Computing, John Wiley and Sons, 1995.
2nd Edition.

References II

[SH09] M. Schwarick and M. Heiner.
CSL model checking of biochemical networks with interval decision diagrams.
In *Proc. CMSB 2009*, pages 296–312. LNCS/LNBI 5688, Springer, 2009.

[SRH11] M Schwarick, C Rohr, and M Heiner.
MARCIE - Model checking And Reachability analysis done effiCIEntly.
In *Proc. 8th International Conference on Quantitative Evaluation of SysTems (QEST 2011)*, pages 91 – 100. IEEE CS Press, September 2011.

[ST10] M. Schwarick and A. Tovchigrechko.
IDD-based model validation of biochemical networks.
TCS 412, pages 2884–2908, 2010.

[Ste94] W.J. Stewart.
Introduction to the Numerical Solution of Markov Chains.
Princeton Univ. Press, 1994.