Non-Exponential Stochastic Petri Nets: an Overview of Methods and Techniques

PART 1: Semantics and Specifications

Andrea Bobbio
Dipartimento di Informatica
Università di Torino, 10149 Torino, Italy

PART 2: Solution Techniques

Miklós Telek
Department of Telecommunications
Technical University of Budapest, 1521 Budapest, Hungary

PNPM'97
International Workshop on
Petri Nets and Performance Models
Saint-Malo: June 3, 1997

Motivation and Outline

The aim of this tutorial is to present recent research work in the area of non-Markovian stochastic Petri nets with particular emphasis on the semantical implication of these models, their specification and the related solution techniques.

◇ The concept of memory in non-Markovian SPN.

◇ Semantics of non-Markovian SPN.

◇ Modeling specifications and restrictions in non-Markovian SPN.

◇ Solution techniques

- Markov Regenerative Stochastic PN;
- Supplementary variables;
- State space expansion.

◇ Transient and Steady State analysis.

◇ Examples.

Why non-exponential models:

- Non-exponential event duration:
  Process execution,
  Message transmission,
  ...

- Modeling systems with deadlines:
  Real-time systems,
  Transmission protocols,
  ...

- Computing the Cdf of cumulative measures:
  Completion time,
  Accumulated reward,
  Performability,
  ...

Why Petri nets:

- Interface language:
  PN provides a modeler's representation from which the analytical representation can be automatically derived.

Marked Petri Net: Definition

A marked *PN* is a tuple $PN = (P, T, I, O, H, M)$, where:

- $P = \{p_1, p_2, \ldots, p_{np}\}$ is the set of places;

- $T = \{t_1, t_2, \ldots, t_{nt}\}$ is the set of transitions;

- I, O and H are the input, the output and the inhibitor functions, respectively. I provides the multiplicities of the input arcs from P to T; O provides the multiplicities of the output arcs from T to P; H provides the multiplicity of the inhibitor arcs from P to T;

- $M = \{m_1, m_2, \ldots, m_{np}\}$ is the marking. m_i provides the number of tokens in place p_i, in marking M.

The reachability set $\mathcal{R}(M_0)$ is the set of all the markings that can be generated from an initial marking M_0 by repeated application of the enabling and firing rules.

Timed Execution Sequence

A timed execution sequence $\mathcal{T}_E$ is a connected path in the reachability graph $\mathcal{R}(M_0)$ augmented by a non-decreasing sequence of real non-negative values representing the epochs of firing of each transition.

Consecutive transition firings correspond to ordered time instants $\tau_i \leq \tau_{i+1}$ in $\mathcal{T}_E$.

$$\mathcal{T}_E = \{ (\tau_0, M_{(0)}) \, ; \, (\tau_1, M_{(1)}) \, ; \, \ldots \, ; \, (\tau_i, M_{(i)}) \, ; \, \ldots \}$$

The time interval $\tau_{i+1} - \tau_i$ between consecutive epochs represents the period of time that the *PN* sojourns in marking $M_{(i)}$.

Timed and Stochastic PN

A variety of timing mechanisms have been proposed in the literature.

The distinguishing features of the timing mechanisms are whether the duration of the events is modeled by deterministic or random variables, and whether the time is associated to places, transitions or tokens.

The most common assumption is that time is assigned to the duration of events represented by the transitions.

If a probability measure is assigned to the duration of the events represented by each timed transition, $\mathcal{T}_E$ is mapped into a stochastic process $\mathcal{M}(t)$, $(t \geq 0)$, called the *Marking Process.*

PN's in which the timing mechanism is stochastic are referred to as Stochastic *PN* (*SPN*).

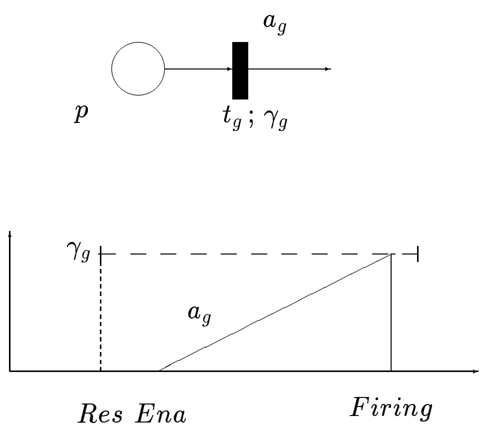
Generally Distributed Transition_SPN (GDT_SPN)

The semantics of a *SPN* with stochastic timing associated to the *PN* transitions and with generally distributed firing times was first defined by [Ajmone et al. 1989].

DEFINITION. A stochastic GDT_SPN is a marked SPN in which:

- ◇ *To any timed transition $t_k \in T$ is associated a random variable γ_k with Cdf $G_k(x)$.*
- ◇ *Each timed transition $t_k \in T$ is attached a memory variable a_k and a memory policy: the memory policy specifies how a_k is related to the past enabling time of the transition.*
- ◇ *An initial probability is given on the reachability set.*

The Individual Clock Memory Model



a)

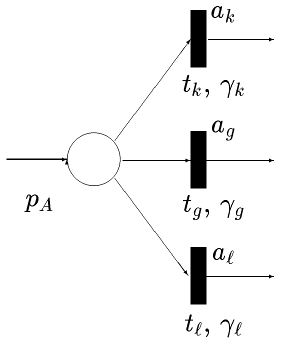
b)

An individual clock is associated to each timed PN transition.

The clock keeps track of the time in which t_g has been enabled.

The age variable a_g represents the clock count and transition t_g fires when the clock count a_g reaches the firing time level γ_g (formulation as a first passage time problem).

Preemption and Memory Model



When a generally distributed transition is disabled, and then re-enabled, a decision must be taken on the following points.

◇ How to account for the previous clock count:

- the clock is reset;
- the clock resumes its value so that only the remaining firing time must be completed.

◇ How to account for the firing time γ_g:

- the firing time is resampled;
- the firing time is maintained.

Execution Policy in a GDT_SPN

The *execution policy* comprises two specifications: the *firing policy* and the *memory policy.*

Firing policy - A natural choice for specifying how to select the next transition to fire is according to a *race policy.*

Memory Policy - The way in which a_k is related to the past history determines how the process is conditioned upon the past. We consider three different memory policies:

- *Age memory* - The age variable a_k accounts for the total time in which t_k has been enabled from its last firing.
- *Enabling memory* - The age variable a_k accounts for the time elapsed from the last epoch in which t_k has been enabled. When transition t_k is disabled (even without firing) the corresponding enabling age variable is reset.
- *Resampling* - The age variable a_k is reset to zero at any change of marking.

A Taxonomy for the Memory Policies (1)

In all the previous literature on non-Markovian SPN (originated by [Ajmone et al. 1989]) the firing time is implicitly assumed to be *resampled* each time the corresponding transition becomes *active.*

In general, the memory of a single transition can be considered as composed by two elements: an *active time* and a *resampling time.*

1. *The active time:*
   is the time during which the age variable a_g is different from 0 (i.e. the *active time* counts the time from which the transition was first enabled after being reset).

2. *The resampling time*:
   is the time during which the firing time level γ_g maintains its value (i.e. the *resampling time* counts the time from which the current value of the firing time level was set).

A Taxonomy for the Memory Policies (2)

By combining the *active time* with the *resampling time* we can construct new firing policies in which the age variable a_g can be reset, but the firing time γ_g remains effective and unchanged in the successive enabling periods.

A possible manifestation of this new firing policy, is the *preemptive repeat identical (pri)* policy, that has been formulated for the first time, in the context of *SPN*, by [Bobbio et al. PNPM95].

A Taxonomy for the Memory Policies (3)

In general, the memory policy must be formulated in terms of a combination of the *active time* and the *resampling time.*

By borrowing terminology from the queueing theory, the following taxonomy can be introduced.

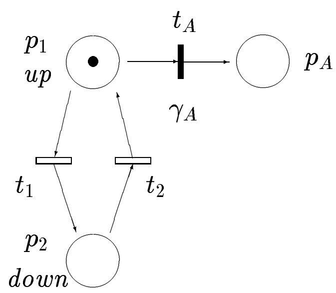
A *PN* transition can be:

◇ *Preemptive Resume (prs)*:
If the active time follows an age memory policy and the firing time resampling is synchronized with the age variable resetting.

◇ *Preemptive Repeat Different (prd)*:
If the active time follows an enabling memory policy and the firing time resampling is synchronized with the age variable resetting.

◇ *Preemptive Repeat Identical (pri)*:
If the active time follows an enabling memory policy and the firing time resampling occurs only at the transition firing.

Response time in a dependable system: An Example (1)

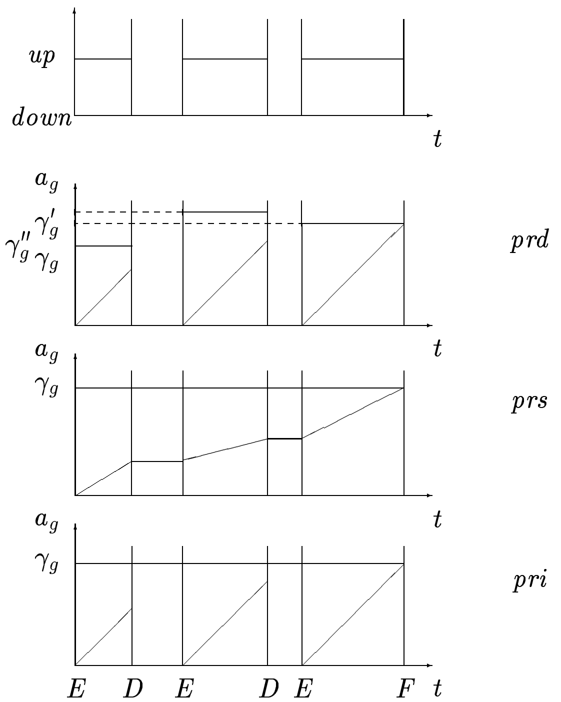


A system alternates between an *up* and a *down* state.

The system must process a job of duration γ_A, but is able to perform useful work only when it is in the *up* state.

The figure shows the corresponding *SPN*. The *Cdf* of the response time, is given by the probability of absorption in place p_A.

Response time in a dependable system: An Example (2)



Computational Restrictions for GDT_SPN

The marking process $\mathcal{M}(t)$ generated by a *GDT_SPN* does not have, in general, an analytically tractable formulation.

Various restrictions of the general model have been discussed in the literature such that $\mathcal{M}(t)$ is confined to belong to a known class of analytically tractable stochastic processes.

1. **Exponentially Distributed SPN**
2. **Semi-Markov SPN**
3. **Phase Type SPN (PHSPN)**
4. **Markov Regenerative SPN (MRSPN)**
   - **Deterministic SPN (DSPN)**
   - **Enabling Memory MRSPN***
   - **Non-overlapping Dominant Transitions MRSPN**

Exponentially Distributed SPN

All the random variables γ_k associated to the timed *PN* transitions are exponentially distributed.

The marking process $\mathcal{M}(t)$ is mapped into a *CTMC*, with state space isomorphic to the tangible subset of the reachability graph.

This restriction is the most popular in the literature, and a number of tools are built on this assumption (*GSPN, SPNP, UltraSAN, SURF, PEPNET, ...*).

Note - this model cannot accommodate *pri* policies, while *prd* and *prs* policies have the same effect (due to the memoryless property of the exponential distribution).

Semi-Markov SPN

The Semi-Markov model appeared in the first studies on *SPN* in the early '80, by assigning to all the *PN* transitions a resampling memory policy.
This model seems of little interest in applications.

A more consistent and interesting semi-Markov *SPN* can be defined by partitioning the transitions into three classes *[DUGA84]*:

- exclusive,
- competitive,
- concurrent.

Provided that the firing time of all the concurrent transitions is exponentially distributed and that non-exponential competitive transitions are resampled at the time the transition is enabled, the associated marking process becomes a semi-Markov process.

Phase Type SPN (PHSPN)

A PHSPN is a GDT_SPN in which:

- *To any timed transition $t_k \in T$ is associated a PH random variable γ_k. The PH model has ν_k stages with a single initial stage numbered stage 1 and a single final stage numbered stage ν_k.*
- *To any timed transition $t_k \in T$ is assigned a memory policy (age, enabling or resampling).*

The distinguishing feature of the *PHSPN* model, is that the non-Markovian marking process $\mathcal{M}(t)$ generated by the *GDT_SPN* over the reachability set $\mathcal{R}(M_0)$ is converted into a *CTMC* defined over an expanded state space.

The measures pertinent to the original process are defined at the *PN* level and can be evaluated by solving the expanded *CTMC*.

Advantages of the PHSPN model

ADVANTAGES

- ◇ It is possible to design a completely automated tool in which the user can assign a *PH* distribution and a *memory policy* to each timed transition *[CUMANI 1985]*.
- ◇ The expanded *CTMC* is automatically generated from the model specifications (the *PN* topology, and the *PH* models assigned to each timed transition).
- ◇ The generation algorithm is driven by the different execution policies that the user assigns to each transition.
- ◇ Each marking of the original reachability set, is mapped into a macro state in the expanded graph. This mapping allows the program to redefine the measures calculated over the expanded graph in terms of the markings of the original *PN*.

Disadvantages of the PHSPN model

DISADVANTAGES

◇ The firing time of each timed transition must be approximated by a *PH* random variable.

◇ Only *prd* and *prs* memory policies can be accommodated.

◇ Explosion of the expanded state space.

The largness problem can be alleviated by exploiting the structural properties of the expanded infinitesimal generator, by resorting to Kronecker algebra operators.

Markov Regenerative SPN (MRSPN)

Assertion - *If at time τ_n^* of entrance in a tangible marking M_n all the memory variables a_k are equal to zero, and the firing times γ_k resampled, τ_n^* is a regeneration time point for the marking process $\mathcal{M}(t)$.*

$$\mathcal{T}_E = \{ (\tau_0^*, M_{(0)}) ; (\tau_1^*, M_{(1)}) ; \ldots ; (\tau_n^*, M_{(n)}) ; \ldots \}$$

The embedded sequence of regeneration time points and associated states $(\tau_n^*, M_{(n)})$ is a Markov renewal sequence and the marking process $\mathcal{M}(t)$ is a Markov regenerative process.

Definition - *A GDT_SPN, for which an embedded Markov renewal sequence $(\tau_n^*, M_{(n)})$ exists, is a Markov Regenerative Stochastic Petri Nets (MRSPN).*

Definition - *The marking process between any two successive regeneration time points is called the subordinated process.*

A historical view on MRSPN (1)

The first model in this line was the *Deterministic and Stochastic PN (DSPN)* model introduced by Ajmone and Chiola (1987) with the aim of providing a technique for combining exponential and deterministic timings.

Choi, Kulkarni and Trivedi (1993, 1994) have shown that the marking process associated to a *DSPN* is a Markov regenerative process (*MRGP*).

In their work, the considered *MRSPN* model was defined in the following way:

- *At most, a single GEN transition is allowed to be enabled in each marking (being all the other transitions EXP).*

- *The only allowed memory policy for the GEN transitions is the* enabling memory*.*

As a consequence, the marking process between any two successive regeneration time points (the subordinated process) is a *CTMC*.

A historical view on MRSPN (2)

German and Lindemann have determined the steady state solution of the same model by resorting to the method of supplementary variables. The results have been implemented in the package *DSPNexpress* [Lindemann 95].

German has successively provided the transient analysis of the same model, and comparative results of various numerical techniques have been presented [German et al. - PNPM95].

Ciardo et al. consider the steady state analysis of *MRSPN* when multiple GEN transitions of enabling memory type are simultaneously enabled in a marking. The subordinated process becomes a semi-Markov process.

A historical view on MRSPN (3)

Bobbio and Telek have introduced the class of *MRSPN* with non-overlapping dominant transitions.

This class extends the previous model definitions in the following way:

◇ All the defined memory policies *prd*, *prs* and *pri* can be accommodated into a single *PN*;

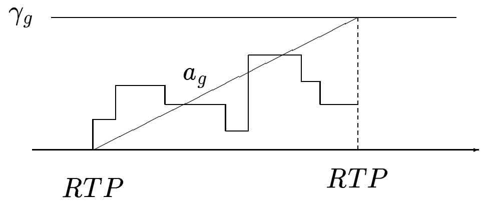
◇ Simultaneous enabling of different GEN transitions inside the same subordinated process;

◇ The subordinated process inside the non-overlapping activity cycle of any dominant transition can be a semi-Markov process.

◇ Transient and steady state results are provided.

Dominant Transition

With the aim of increasing the modeling power of *MRSPN* to any combination of *prd*, *prs* and *pri* memory policies, Bobbio and Telek have defined models with non-overlapping dominant transitions.



A transition is dominant over a regeneration period, if the initial and final regeneration time points correspond to the start and to the end of its active or resampling time.

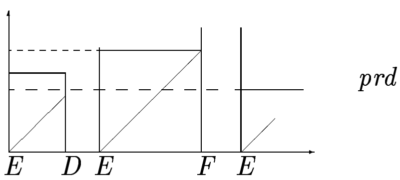
In order to completely characterize the firing conditions of a dominant transition, two elements must be known:

◇ the value of the memory variable a_g at time t;

◇ the value of the firing time γ_g.

Prd Memory Policy

The *prd* memory policy for a dominant transition is shown in the figure where E, D and F are the enabling, disabling and firing time points.



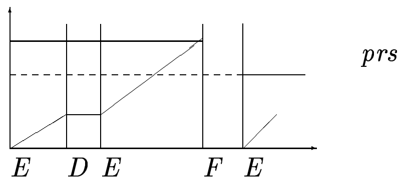
This model is based on the following assumptions:

- *The active time of the transition follows an* enabling memory *policy: the age variable a_g is reset each time t_g is disabled or fires.*
- *The firing time γ_g is resampled each time the age variable is reset (i.e. each time the transition is disabled or fires).*

This restriction is the most popular and was introduced by Ajmone and Chiola [1986] with DET transitions, then generalized by Choi-Kulkarni-Trivedi [1993,1994], Lindemann [1993], Ciardo-German-Lindemann [1994], German [1995].

Prs Memory Policy

The *prs* memory policy for a dominant transition is shown in the figure where E, D and F are the enabling, disabling and firing time points.

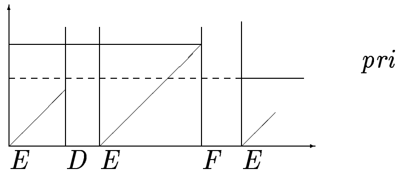


- *The active time of the GEN transition follows an* age memory *policy: the age variable a_g is reset only when t_g fires.*
- *The firing time γ_g is resampled each time the age variable is reset (i.e. when the transition fires).*

This model was introduced by Bobbio and Telek [1995].

Pri Memory Policy

The *pri* memory policy for a dominant transition is shown in the figure where E, D and F are the enabling, disabling and firing time points.



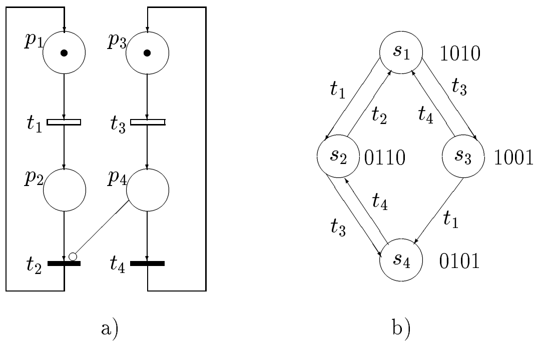
- *The active time of the GEN transition follows an* enabling memory *policy: the age variable a_g is reset each time t_g is disabled or fires.*
- *The firing time γ_g is resampled only upon firing of the transition.*

The extension to *pri* memory policies was discussed by Bobbio et al. [PNPM'95].

An example: M/D/1/2/2 queue with different classes of customers

The two customers are of different classes, and customer of class 2 preempts customer of class 1 but not viceversa.

A *prd* or a *prs* or a *pri* service policy can be realized by assigning to transitions t_2 the appropriate memory policy.



Regeneration time points and markings

When t_2 is of prd type all s_1, s_2, s_3 and s_4 can be regeneration states.

When t_2 is of prs or pri type only s_1, s_2 and s_3 can be regeneration states, while s_4 can never be a regeneration state.

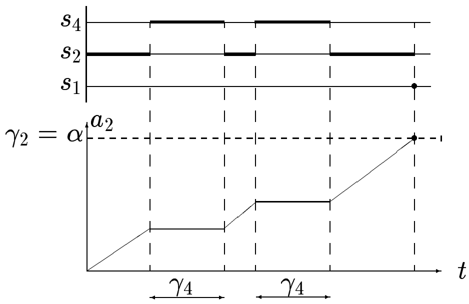
In s_1 only EXP transitions are enabled and the next regeneration states can be either s_2 or s_3.

From s_3 the next regeneration marking can be s_1 or s_2 depending whether a type 1 job does require service or not. The subordinated process is a *CTMC*.

When t_2 is of prd type from s_2 the next regeneration state can be s_1 (if t_2 fires before t_3) or s_4 (if t_3 fires before t_2). The first state transition concludes the subordinated process.

When t_2 is of prs or pri type from s_2 the next regeneration state can be only s_1, but multiple cycles (s_2 - s_4) can occur depending whether type 2 jobs arrive. The subordinated process is a *SMP*.

Process subordinated to s_2 (the policy of t_2 is prs)



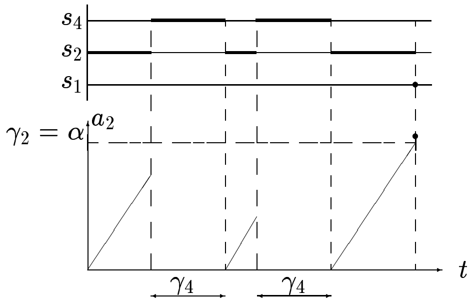
Transition t_2 is the dominant transition.

The process $\mathcal{M}^{(2)}(x)$ subordinated to state s_2 includes the states s_2, s_4 reachable from s_2 before firing of t_2.

s_2 is the only state in which t_2 is enabled during the regenerative period.

$\mathcal{M}^{(2)}(x)$ is semi-Markov process since t_4 is GEN.

Process subordinated to s_2 (the policy of t_2 is pri)



Transition t_2 is the *pri* dominant transition.

The process $\mathcal{M}^{(2)}(x)$ subordinated to state s_2 includes the states s_2, s_4 reachable from s_2 before firing of t_2.

$\mathcal{M}^{(2)}(t)$ is semi-Markov since t_4 is GEN.

PART 2: Solution techniques

The exact analysis methods of non-Markovian Petri nets presented in the literature so far are based on one of the following approaches:

◇ Markov regenerative approach (MRA)

◇ Supplementary variable approach (SVA)

The most popular approximation methods of non-Markovian Petri nets is the:

◇ Phase type approximation approach (PHA)

The standard problem considered here is:

- at most one general transition is enabled/-active/sampled at a time,
- general transitions can be prd, prs, pri type.

Extensions and generalizations of this model are possible and already published, but we do not intend to consider all of them.

Solution techniques

Applicability of approaches in transient and steady state analysis

	prd	prs	pri
MRA in time domain	X	∅	∅
SVA	X	X*	∅
MRA in transform domain	X	X	X
PHA	X	X	∅

X* possible, but never considered

Expected limits of further generalization:

- MRA: overlapping active/sampled intervals
- SVA: pri memory policy
- PHA: pri memory policy

Markov regenerative approach

The Markov regenerative approach is based on the Markov renewal theory and makes the use of the transient and steady state analysis methods available for MRGPs.

The approach consists of the steps:

1. characterization of regenerative time points
2. analysis and "limited" description of all the possible subordinated processes in isolation
   - next regeneration
   - process up to the next regeneration
3. analysis of the overall process.

Step 1:
Regenerative time points, as already defined, are instants of time when all the memory variables are reset to 0 and all the firing times are resampled.

The next two steps are different in case of transient or steady state analysis.

Transient analysis with MRA

Step 2: Analysis of subordinated processes

The transient analysis of the overall process requires only a limited description of the subordinated processes by the mean of the following matrix valued functions [Choi et al. 1993]:

$$\mathbf{K}(t) = [K_{ij}(t)] \; ; \; K_{ij}(t) = Pr\{M_{(1)} = j \, , \, \tau_1^* \leq t | \, \mathcal{M}(0) = i\}$$

$$\mathbf{E}(t) = [E_{ij}(t)] \; ; \; E_{ij}(t) = Pr\{\mathcal{M}(t) = j \, , \, \tau_1^* > t | \, \mathcal{M}(0) = i\}$$

Matrix $\mathbf{K}(t)$ is the *global kernel*; describes the next regeneration time point and state.

Matrix $\mathbf{E}(t)$ is the *local kernel*; describes the transient behaviour of subordinated processes.

$\mathbf{K}(t)$ is often determined by the state from which the next regeneration occurs and a branching probability matrix (Δ) describes the state transition at the regeneration instant.

Transient analysis with MRA

Step 2: Analysis of subordinated processes

Each row of these matrices is obtained by the isolated analysis of the subordinated process with state space $S_i \subset S$. Hence, in the worst case, *Step 2* requires the analysis of $\#S$ systems of size $\#S$.

The analysis of a single subordinated process starts from state i ($\mathcal{M}^{(i)}(t)$) often makes the use of matrices or matrix valued functions of size $\#S_i \times \#S_i$, in which useless states are eliminated and/or the states are reordered. The result obtained for the analysis of the size $\#S_i$ system has to be properly saved into the i-th row of matrices $\mathbf{K}(t)$ and $\mathbf{E}(t)$.

The Laplace Stieltjes Transforms *(LST)* of these quantities are $\mathbf{K}^{\sim}(s)$ and $\mathbf{E}^{\sim}(s)$.

Transient analysis with MRA

Subordinated process without internal state transition.

When only exponential transitions are enabled in a marking i

$$K_{ij}(t) = \sum_{d \text{ enabled in } i} \frac{\lambda_d}{\lambda^i} (1 - e^{-\lambda^i t}) \, \Delta_{ij}^d$$

$$E_{ij}(t) = \delta_{ij} \, e^{-\lambda^i t}$$

where $\lambda^i = \sum_{d \text{ enabled in } i} \lambda_d$ and δ_{ij} is the Kronecker delta.

When general transitions are also enabled in a marking i, but the first state transition results in a new regeneration time point then

$$K_{ij}(t) = \sum_{d \text{ en. in } i} \int_{u=0}^{t} \prod_{\substack{e \text{ en. in } i; \\ e \neq d}} (1 - F_e(t)) \;\; d\, F^i(t) \;\; \Delta_{ij}^d$$

$$E_{ij}(t) = \delta_{ij} \, (1 - F^i(t))$$

where $F_d(t)$ is the CDF of the firing time distribution of transition d, and

$$F^i(t) = 1 - \prod_{d \text{ enabled in } i} (1 - F_d(t)) \; .$$

Transient analysis with MRA

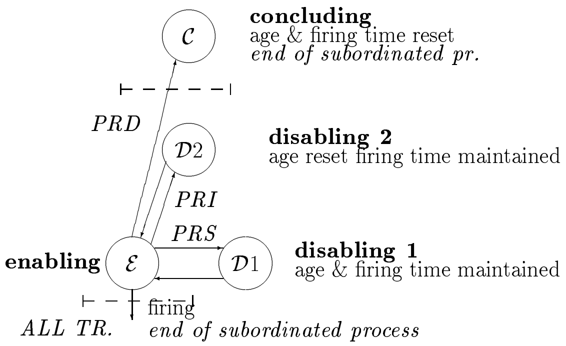
State space of the subordinated process with state transitions.

The state space of the stochastic process subordinated to a dominant general transition can be partitioned as follows:

- $\mathcal{E}$ - enabling set: the dominant transition is enabled
- $\mathcal{D}$ - disabling set: the dominant transition is disabled
  - $\mathcal{D}1$ - the age variable is maintained
  - $\mathcal{D}2$ - the age variable is reset to 0
- $\mathcal{C}$ - concluding set: states are not reachable in the subordinated process (a state transition to $\mathcal{C}$ concludes the subordinated process)

Transient analysis with MRA

State space of the subordinated process with state transitions.



Structure of the state space with different dominant transitions:

transition	structure of state space
prd	no transition to $\mathcal{D}1$ and $\mathcal{D}2$
pri	no transition to $\mathcal{D}1$ and $\mathcal{C}$
prs	no transition to $\mathcal{D}2$ and $\mathcal{C}$

Transient analysis with MRA

Measures of the subordinated process over the partitioned state space.

Let us fix the value of the firing time $\gamma_g = w$.

Matrices of measures describe the stochastic process over the parts of the state space are [Telek et al. 1996]:

- $\mathbf{P1}(t,w)$ - state transition probabilities inside $\mathcal{E}$ before leaving $\mathcal{E}$
- $\mathbf{F1}(t,w)$ - probability of firing before t before leaving $\mathcal{E}$
- $\mathbf{C1}(t,w)$ - probability of concluding before t before leaving $\mathcal{E}$
- $\mathbf{P12}(t,w)$ - probability of transition to $\mathcal{D}$ before t before leaving $\mathcal{E}$

- $\mathbf{P2}(t)$ - state transition probabilities inside $\mathcal{D}$ before leaving $\mathcal{D}$
- $\mathbf{P21}(t)$ - probability of transition to $\mathcal{E}$ before t

Transient analysis with MRA

Analysis of the subordinated process over the partitioned state space.

Suppose the subordinated process is a *CTMC* with generator $\mathbf{A}$ with state numbers ordered according to $\mathcal{E}$, $\mathcal{D}$, $\mathcal{C}$ we have:

$$\mathbf{A} = \begin{array}{|c|c|c|}\hline \mathbf{A}_{\mathcal{E}} & \mathbf{A}_{\mathcal{E}\mathcal{D}} & \mathbf{A}_{\mathcal{E}\mathcal{C}} \\ \hline \mathbf{A}_{\mathcal{D}\mathcal{E}} & \mathbf{A}_{\mathcal{D}} & \emptyset \\ \hline \bullet & \bullet & \bullet \\ \hline \end{array}$$

By which:

$$\mathbf{P1}^{\sim *}(s,v) = \frac{s}{v}\left((s+v)\mathbf{I} - \mathbf{A}_{\mathcal{E}}\right)^{-1}$$

$$\mathbf{F1}^{\sim *}(s,v) = \left((s+v)\mathbf{I} - \mathbf{A}_{\mathcal{E}}\right)^{-1}$$

$$\mathbf{C1}^{\sim *}(s,v) = \frac{1}{v}\left((s+v)\mathbf{I} - \mathbf{A}_{\mathcal{E}}\right)^{-1}\mathbf{A}_{\mathcal{E}\mathcal{C}}$$

$$\mathbf{P12}^{\sim *}(s,v) = \frac{1}{v}\left((s+v)\mathbf{I} - \mathbf{A}_{\mathcal{E}}\right)^{-1}\mathbf{A}_{\mathcal{E}\mathcal{D}}$$

$$\mathbf{P2}^{\sim}(s) = s\ (s\mathbf{I} - \mathbf{A}_{\mathcal{D}})^{-1}$$

$$\mathbf{P21}^{\sim}(s) = (s\mathbf{I} - \mathbf{A}_{\mathcal{D}})^{-1}\mathbf{A}_{\mathcal{D}\mathcal{E}}$$

Transient analysis with MRA

Analysis of the subordinated process over the partitioned state space.

An inverse Laplace transformation with respect to v yields:

$$\mathbf{P1}^{\sim}(s,w) = s\int_0^w e^{x(-s\mathbf{I}+\mathbf{A}_{\mathcal{E}})}dx$$

$$\mathbf{F1}^{\sim}(s,w) = e^{(-s\mathbf{I}+\mathbf{A}_{\mathcal{E}})w}$$

$$\mathbf{C1}^{\sim}(s,w) = \int_0^w e^{x(-s\mathbf{I}+\mathbf{A}_{\mathcal{E}})}dx\ \mathbf{A}_{\mathcal{E}\mathcal{C}}$$

$$\mathbf{P12}^{\sim}(s,w) = \int_0^w e^{x(-s\mathbf{I}+\mathbf{A}_{\mathcal{E}})}dx\ \mathbf{A}_{\mathcal{E}\mathcal{D}}$$

Transient analysis with MRA

Analysis of the overall subordinated process

Matrices of measures describe the state of the stochastic process are [Bobbio and Telek 1995]:

- $\mathbf{P}(t,w)$ - state transition probabilities inside the subordinated process
- $\mathbf{F}(t,w)$ - probability of firing before t
- $\mathbf{C}(t,w)$ - probability of concluding before t

In case of unique numbering of states in S these measures are related with the kernel elements as follows:

$$K_{ij}(t) = \int_0^\infty \sum_{k\in S^i} F^i_{ik}(t,w)\,\Delta^d_{kj} + C^i_{ik}(t,w) dF_d(w)$$

$$E_{ij}(t) = \int_0^\infty P^i_{ij}(t,w) dF_d(w)$$

where $F_d(\cdot)$ is CDF of the firing time of the dominant transition.

Transient analysis with MRA

Analysis of the overall subordinated process

prd type dominant transition

$$\mathbf{P}^\sim(s,w) = \mathbf{P1}^\sim(s,w)$$

$$\mathbf{F}^\sim(s,w) = \mathbf{F1}^\sim(s,w)$$

$$\mathbf{C}^\sim(s,w) = \mathbf{C1}^\sim(s,w)$$

pri type dominant transition

$$\mathbf{P}^\sim(s,w) = [\mathbf{I} - \mathbf{P12}^\sim(s,w)\,\mathbf{P21}^\sim(s)]^{-1} [\mathbf{P1}^\sim(s,w) + \mathbf{P12}^\sim(s,w)\,\mathbf{P2}^\sim(s)]$$

$$\mathbf{F}^\sim(s,w) = [\mathbf{I} - \mathbf{P12}^\sim(s,w)\,\mathbf{P21}^\sim(s)]^{-1}\,\mathbf{F1}^\sim(s,w)$$

$$\mathbf{C}^\sim(s,w) = 0$$

prs type dominant transition

$$\mathbf{P}^{\sim\sim}(s,v) = [\mathbf{I} - \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P21}^\sim(s)]^{-1} [\mathbf{P1}^{\sim\sim}(s,v) \mid \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P2}^\sim(s)]$$

$$\mathbf{F}^{\sim\sim}(s,v) = [\mathbf{I} - \mathbf{P12}^{\sim\sim}(s,v)\,\mathbf{P21}^\sim(s)]^{-1}\,\mathbf{F1}^{\sim\sim}(s,v)$$

$$\mathbf{C}^{\sim\sim}(s,v) = 0$$

Transient analysis with MRA

Step 3: Analysis of the overall marking process

The state transition probability matrix is denoted as [Choi et al. 1993] :

$$\mathbf{V}(t) = [V_{ij}(t)] \; ; \quad V_{ij}(t) = Pr\{\mathcal{M}(t) = j \mid \mathcal{M}(0) = i\}$$

In **time domain** the transient behavior of the *MRSPN* is given by the Volterra integral equation:

$$V_{ij}(t) = E_{ij}(t) + \sum_k \int_0^t d\,K_{ik}(y)\,V_{kj}(t-y)$$

The usual way of the **time domain** analysis is to adopt a numerical technique to solve the Volterra integral equation. An algorithm based on the fix size discretization of time is discussed by German et al. [PNPM95].

Transient analysis with MRA

Step 3: Analysis of the overall marking process

In the **Laplace Stieltjes Transform domain** the state transition probabilities are given by:

$$\mathbf{V}^\sim(s) = [\mathbf{I} - \mathbf{K}^\sim(s)]^{-1}\,\mathbf{E}^\sim(s)$$

In **LST domain** an analytical solution of $\mathbf{V}^\sim(s)$ is feasible, but

- the symbolic evaluation of $[\mathbf{I} - \mathbf{K}^\sim(s)]^{-1}$ is computationally hard,
- the symbolic inverse transformation of $\mathbf{V}^\sim(s)$ is usually impossible,
- and the numerical inverse transformation of $\mathbf{V}^\sim(s)$ is also computationally hard.

The Steady-State Solution with MRA

Step 2: Analysis of the subordinated processes

The steady-state evaluation of the overall process requires less information on the subordinated processes than the transient analysis.

The required measures are:

$$\alpha_{ij} = \mathbb{E}[\int_0^\infty I_{\mathcal{M}^{(i)}(t)=j} dt]$$

α_{ij} is the expected time a subordinated process starting from state i spends in state j.

$$\pi_{ij} = Pr\{M_{(1)} = j \mid \mathcal{M}(0) = i\}$$

π_{ij} is the probability that the subordinated process starting from state i is followed by a subordinated process starting from state j.

The matrix $\mathbf{\Pi} = \{\pi_{ij}\}$ is the transition probability matrix of the $DTMC$ embedded into the regeneration time points.

The Steady-State Solution with MRA

Step 2: Analysis of the subordinated processes

These measures can be obtained from the global and local kernels both in time and transform domain:

$$\alpha_{ij} = \int_{t=0}^\infty E_{ij}(t)dt = \lim_{s\to 0} 1/s\ E_{ij}^\sim(s)$$

$$\pi_{ij} = \lim_{t\to\infty} K_{ij}(t) = \lim_{s\to 0} K_{ij}^\sim(s)$$

By the partitioned generator of the subordinated $CTMC$ the steady state measures can be effectively computed as follow:

prd type dominant transition

$$\alpha_i \mid_{\gamma_g=w} = \mathbf{L}(w)$$

$$\pi_i \mid_{\gamma_g=w} = e^{w\ \mathbf{A}_\mathcal{E}}\ \mathbf{\Delta} + [\ \mathbf{0} \mid \mathbf{L}(w)\ \mathbf{A}_{\mathcal{E}\mathcal{C}}]$$

where $\mathbf{L}(w) = \int_{x=0}^w e^{x\ \mathbf{A}_\mathcal{E}}\ dx$.

The Steady-State Solution with MRA

Step 2: Analysis of the subordinated processes

prs type dominant transition

$$\alpha_i \mid_{\gamma_g=w} = [\ \mathbf{L}_\beta(w)\ \big|\ -\mathbf{L}_\beta(w)\ \mathbf{A}_{\mathcal{E}\mathcal{D}}\ \mathbf{A}_\mathcal{D}^{-1}\ ]$$

$$\pi_i \mid_{\gamma_g=w} = [\ e^{w\beta}\ ]\ \ \mathbf{\Delta}$$

where

$\beta = \mathbf{A}_\mathcal{E} - \mathbf{A}_{\mathcal{E}\mathcal{D}}\ \mathbf{A}_\mathcal{D}^{-1}\ \mathbf{A}_{\mathcal{D}\mathcal{E}}$ and $\mathrm{L}_\beta(w) = \int_0^w\ e^{x\beta}\ dx$

pri type dominant transition

$$\alpha_i \mid_{\gamma_g=w} = [I - \mathbf{L}(w)\mathbf{A}_{\mathcal{E}\mathcal{D}}\ \mathbf{A}_\mathcal{D}^{-1}\ \mathbf{A}_{\mathcal{D}\mathcal{E}}]^{-1}\ [\mathbf{L}(w) \mid -\mathbf{L}(w)\ \mathbf{A}_{\mathcal{E}\mathcal{D}}\ \mathbf{A}_\mathcal{D}^{-1}]$$

$$\pi_i \mid_{\gamma_g=w} = [I - \mathbf{L}(w)\ \mathbf{A}_{\mathcal{E}\mathcal{D}}\ \mathbf{A}_\mathcal{D}^{-1}\ \mathbf{A}_{\mathcal{D}\mathcal{E}}]^{-1}[e^{w\mathbf{A}_\mathcal{E}}]$$

The Steady-State Solution with MRA

Step 3: Analysis of the overall process

Let $P = \{p_i\}$ be the unique solution of:

$$P = P\mathbf{\Pi}\quad ;\qquad \sum_i p_i = 1$$

The steady-state probabilities of the $MRGP$ become:

$$v_j = \lim_{t\to\infty} Pr\{\mathcal{M}(t) = j\} = \frac{\sum_k p_k\ \alpha_{kj}}{\sum_k p_k\ \sum_\ell \alpha_{k\ell}}$$

The computational complexity of *Step 3* is determined by the solution of the linear system to obtain P.

The numerical solution of $\lim_{t\to\infty}\mathbf{V}(t)$ and the symbolic solution of $\lim_{s\to 0}\mathbf{V}^\sim(s)$ gives the same result for v_j, but in a computationally much harder way.

However, an "automatic" analytical solution can be obtained by $\lim_{s\to 0}\mathbf{V}^\sim(s)$.

The Supplementary Variable Approach

The **marking process** ($\mathcal{M}(t)$) together with the **age variable** (A) of the active transition of a non-Markovian SPN with at most one active prd or prs general transition is a **Markov process** over the state space $S \times R$, where

- S is the set of reachable tangible markings and
- R is the (sub)set positive real numbers

The joint process can be analyzed by the method of supplementary variable [COX].

The Supplementary Variable Approach

Concept and notations (from German et al. [PNPM95]):

- The state space is divided in two parts
  - S^E set of states in which no general transition is active ($A \doteq 0$)
  - S^g set of states in which one general transition is active
- superscript ^{E} refers to states in S^E and superscript ^{g} (or ^{h}) refers to states in S^g
- state probability

$$\pi_n(t) = Pr[\mathcal{M}(t) = n]$$

- "age rate"

$$p_n(t,x) = \frac{Pr[\mathcal{M}(t) = n, x < A \leq x + dx]}{dx} \cdot \frac{1}{1 - F^g(x)}$$

- firing time distribution $F^g(x)$
- branching probability $\Delta_{i,j}^{g,h}$

Transient analysis with SVA

With the of use proper vectors and matrices a system with prd transitions is characterized as follow:

Partial differential equation describes the process evolution in S^g

$$\frac{\partial}{\partial t}\mathbf{p^g}(t,x) + \frac{\partial}{\partial x}\mathbf{p^g}(t,x) = \mathbf{p^g}(t,x)\mathbf{Q^g}$$

Ordinary differential equation describes the process evolution in S^E.

$$\begin{aligned}\tfrac{d}{dt}\pi^{\mathbf{E}}(t) \; = \; & \pi^{\mathbf{E}}(t)\mathbf{Q}^{\mathbf{E},\mathbf{E}} + \\ & \textstyle\sum_g \int_0^\infty \mathbf{p^g}(t,x)dF^g(x)\mathbf{\Delta}^{\mathbf{g},\mathbf{E}} + \\ & \textstyle\sum_g \pi^{\mathbf{g}}(t)\mathbf{Q}^{\mathbf{g},\mathbf{E}}\end{aligned}$$

State probabilities in S^E changes by

- firing of an exponential transition (1st term),
- after firing of a general transition only exponential transitions are enabled (2nd term),
- the same after disabling a general transition (3rd term).

Transient analysis with SVA

Boundary condition:

$$\begin{aligned}\mathbf{p^g}(t,0) \; = \; & \pi^{\mathbf{E}}(t)\mathbf{Q}^{\mathbf{E},\mathbf{g}} + \\ & \textstyle\sum_h \int_0^\infty \mathbf{p^h}(t,x)dF^h(x)\mathbf{\Delta}^{\mathbf{h},\mathbf{g}} + \\ & \textstyle\sum_h \pi^{\mathbf{h}}(t)\mathbf{Q}^{\mathbf{h},\mathbf{g}}\end{aligned}$$

General transition g can be activated by the firing of an exponential

- transition in S^E (1st term),
- by firing of a general transition (2nd term)
- or by disabling the active general transition (3rd term).

State probabilities in S^g are:

$$\pi^{\mathbf{g}}(t) = \int_0^\infty \mathbf{p^g}(t,x)(1 - F^g(x))dx$$

Initial conditions are

- $\pi^{\mathbf{E}}(0)$
- $\mathbf{p^g}(0,x) = \pi^{\mathbf{g}}(0)\delta(x)$

Transient analysis with SVA

The analysis of the transient behaviour is based on a numerical evaluation of the system of equations. An iterative algorithm based on the fix size (h) discretization of the continuous variables proposed by German et al. [PNPM95] consists of the steps:

1. compute age rates in the next time instant
$$\mathbf{p}^{\mathbf{g}}(ih, jh) = \mathbf{p}^{\mathbf{g}}((i-1)h, (j-1)h)e^{\mathbf{Q}^{\mathbf{g}}h}$$
and set $\mathbf{p}^{\mathbf{g}}(ih, 0) = 0$
2. compute the state probabilities $\pi^{\mathbf{g}}(ih)$ by $\mathbf{p}^{\mathbf{g}}(ih, jh),\ j = 0, 1, \ldots$
3. compute the state probabilities $\pi^{\mathbf{E}}(ih)$ by the ordinary differential equation
4. compute the activation rate of general transitions $\mathbf{p}^{\mathbf{g}}(ih, 0)$ by the boundary condition
5. check the convergence and go back to Step 2 or start with the next time instant $(i+1)h$

Steady state analysis with SVA

Let the steady state measures be: $\mathbf{p}^{\mathbf{g}}(x)$, $\pi^{\mathbf{E}}$

The steady state description of the system (if exist) is obtained by eliminating the time dependent behaviour:

$$\frac{d}{dx}\mathbf{p}^{\mathbf{g}}(x) = \mathbf{p}^{\mathbf{g}}(x)\mathbf{Q}^{\mathbf{g}}$$

$$\begin{aligned} 0 = {} & \pi^{\mathbf{E}}\mathbf{Q}^{\mathbf{E},\mathbf{E}} + \\ & \sum_g \int_0^\infty \mathbf{p}^{\mathbf{g}}(x)dF^g(x)\mathbf{\Delta}^{\mathbf{g},\mathbf{E}} + \\ & \sum_g \pi^{\mathbf{g}}\mathbf{Q}^{\mathbf{g},\mathbf{E}} \end{aligned}$$

$$\begin{aligned} \mathbf{p}^{\mathbf{g}}(0) = {} & \pi^{\mathbf{E}}\mathbf{Q}^{\mathbf{E},\mathbf{g}} + \\ & \sum_h \int_0^\infty \mathbf{p}^{\mathbf{h}}(x)dF^h(x)\mathbf{\Delta}^{\mathbf{h},\mathbf{g}} + \\ & \sum_h \pi^{\mathbf{h}}\mathbf{Q}^{\mathbf{h},\mathbf{g}} \end{aligned}$$

$$\pi^{\mathbf{g}} = \int_0^\infty \mathbf{p}^{\mathbf{g}}(x)(1 - F^g(x))dx$$

Steady state analysis with SVA

The solution of the differential equation is

$$\mathbf{p}^{\mathbf{g}}(x) = \mathbf{p}^{\mathbf{g}}(0)e^{x\mathbf{Q}^{\mathbf{g}}}$$

Introduce matrices

$$\Omega^g = \int_0^\infty e^{x\mathbf{Q}^{\mathbf{g}}}dF^g(x), \quad \Psi^g = \int_0^\infty e^{x\mathbf{Q}^{\mathbf{g}}}(1-F^g(x))dx.$$

Where:

Ω^g_{ij} is the probability that the subordinated process of general transition g which starts from state i completes from state j, and

Ψ^g_{ij} is the mean time this subordinated process spends in state j.

Relevant row(s) of matrices π and α of the Markov regenerative approach are associated with Ω^g and Ψ^g as follows:

$$\pi_{ij} = \sum_k \Omega^g_{ik}\Delta^g_{kj}, \qquad \alpha_{ij} = \Psi^g_{ij}$$

Steady state analysis with SVA

By these notations the steady state system simplifies to the system of linear equations:

$$\begin{aligned} 0 = {} & \pi^{\mathbf{E}}\mathbf{Q}^{\mathbf{E},\mathbf{E}} + \\ & \sum_g \mathbf{p}^{\mathbf{g}}(0)\mathbf{\Omega}^{\mathbf{g}}\mathbf{\Delta}^{\mathbf{g},\mathbf{E}} + \\ & \sum_g \mathbf{p}^{\mathbf{g}}(0)\mathbf{\Psi}^{\mathbf{g}}\mathbf{Q}^{\mathbf{g},\mathbf{E}} \end{aligned}$$

$$\begin{aligned} \mathbf{p}^{\mathbf{g}}(0) = {} & \pi^{\mathbf{E}}\mathbf{Q}^{\mathbf{E},\mathbf{g}} + \\ & \sum_h \mathbf{p}^{\mathbf{h}}(0)\mathbf{\Omega}^{\mathbf{h}}\mathbf{\Delta}^{\mathbf{h},\mathbf{g}} + \\ & \sum_h \mathbf{p}^{\mathbf{h}}(0)\mathbf{\Psi}^{\mathbf{h}}\mathbf{Q}^{\mathbf{h},\mathbf{g}} \end{aligned}$$

By the solution of the linear system the steady state probabilities of states in S^g are obtained by

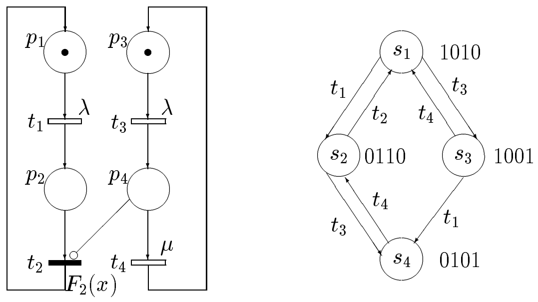
$$\pi^{\mathbf{g}} = \mathbf{p}^{\mathbf{g}}(0)\mathbf{\Psi}^{\mathbf{g}}$$

An example: M/G/1/2/2 queue with different classes of customers

The two customers are of different classes, and customer of class 2 preempts customer of class 1 but not viceversa.

A *prd* or a *prs* or a *pri* service policy can be realized by assigning to transitions t_2 the appropriate memory policy.

The high priority customer has an exponential service time (t_4) and the low priority customer has a general service time (t_2)



a) b)

An example: M/G/1/2/2 queue with different classes of customers

PRD service policy - transient analysis

state	type of reg. per.	int. st.	next reg. st.
s_1	Exp.	s_1	s_2, s_3
s_2	*PRD - CTMC*	s_2	s_1, s_4
s_3	Exp.	s_3	s_1, s_4
s_4	Exp.	s_4	s_2

Exponential regeneration period starting from s_1

$$E^{\sim}_{11}(s) = \frac{s}{s+2\lambda} \; ; \quad K^{\sim}_{12}(s) = \frac{\lambda}{s+2\lambda} \; ; \quad K^{\sim}_{13}(s) = \frac{\lambda}{s+2\lambda}$$

PRD-CTMC regeneration period starting from s_2

Generator matrix of the subordinated process:

$$\mathbf{A} = \begin{array}{|c|c|} \hline \mathbf{A}_{\mathcal{E}} & \mathbf{A}_{\mathcal{EC}} \\ \hline \bullet & \bullet \\ \hline \end{array} = \begin{array}{|c|c|} \hline -\lambda & \lambda \\ \hline \bullet & \bullet \\ \hline \end{array}$$

An example: M/G/1/2/2 queue with different classes of customers

PRD service policy - transient analysis

Measures of the subordinated process:

$$\mathbf{P}^{\sim}(s,w) = \mathbf{P1}^{\sim}(s,w) = s \int_0^w e^{-x(s+\lambda)} dx$$

$$\mathbf{F}^{\sim}(s,w) = \mathbf{F1}^{\sim}(s,w) = e^{-w(s+\lambda)}$$

$$\mathbf{C}^{\sim}(s,w) = \mathbf{C1}^{\sim}(s,w) = \lambda \int_0^w e^{-x(s+\lambda)} dx$$

Kernel elements:

$$E^{\sim}_{22}(s) = \int_0^\infty P^{\sim}_{11}(s,w) dF_2(w)$$

$$K^{\sim}_{21}(s) = \int_0^\infty F^{\sim}_{11}(s,w) dF_2(w)$$

$$K^{\sim}_{24}(s) = \int_0^\infty C^{\sim}_{11}(s,w) dF_2(w)$$

An example: M/G/1/2/2 queue with different classes of customers

PRS service policy - transient analysis

state	type of reg. per.	int. st.	next reg. st.
s_1	Exp.	s_1	s_2, s_3
s_2	*PRS - CTMC*	s_2, s_4	s_1
s_3	Exp.	s_3	s_1, s_4
s_4	Exp.	s_4	s_2

PRS-CTMC regeneration period starting from s_2:

$$\mathbf{A} = \begin{array}{|c|c|} \hline \mathbf{A}_{\mathcal{E}} & \mathbf{A}_{\mathcal{ED}} \\ \hline \mathbf{A}_{\mathcal{DE}} & \mathbf{A}_{\mathcal{D}} \\ \hline \end{array} = \begin{array}{|c|c|} \hline -\lambda & \lambda \\ \hline \mu & -\mu \\ \hline \end{array}$$

Partitioned state space related measures (after an LT → LST with respect to v):

$$\mathbf{P1}^{\sim\sim}(s,v) = \frac{s}{s+v+\lambda}$$

$$\mathbf{F1}^{\sim\sim}(s,v) = \frac{v}{s+v+\lambda}$$

$$\mathbf{P12}^{\sim\sim}(s,v) = \frac{\lambda}{s+v+\lambda}$$

$$\mathbf{P2}^{\sim}(s) = \frac{s}{s+\mu}$$

$$\mathbf{P21}^{\sim}(s) = \frac{\mu}{s+\mu}$$

An example: M/G/1/2/2 queue with different classes of customers

PRS service policy - transient analysis

Measures of the subordinated process:

$$P_{11}^{\sim\sim}(s,v) = \frac{s(s+\mu)}{(s+v+\lambda)(s+\mu)-\lambda\mu}$$
$$P_{12}^{\sim\sim}(s,v) = \frac{s\lambda}{(s+v+\lambda)(s+\mu)-\lambda\mu}$$
$$F_{11}^{\sim\sim}(s,v) = \frac{v(s+\mu)}{(s+v+\lambda)(s+\mu)-\lambda\mu}$$
$$\mathbf{C}^{\sim\sim}(s,v) = 0$$

Kernel elements are obtained after a symbolic inverse transformation with respect to v and unconditioning with respect to the firing time.

An example: M/G/1/2/2 queue with different classes of customers

PRI service policy - transient analysis

state	type of reg. per.	int. st.	next reg. st.
s_1	Exp.	s_1	s_2, s_3
s_2	*PRI - CTMC*	s_2, s_4	s_1
s_3	Exp.	s_3	s_1, s_4
s_4	Exp.	s_4	s_2

PRI-CTMC regeneration period starting from s_2: **A** is the same as for the PRS case.

Partitioned state space related measures of the subordinated process:

$$\mathbf{P1}^{\sim}(s,w) = s\int_0^w e^{-x(s+\lambda)}dx = \frac{s}{s+\lambda}(1-e^{-w(s+\lambda)})$$
$$\mathbf{F1}^{\sim}(s,w) = e^{-w(s+\lambda)}$$
$$\mathbf{P12}^{\sim}(s,w) = \lambda\int_0^w e^{-x(s+\lambda)}dx = \frac{\lambda}{s+\lambda}(1-e^{-w(s+\lambda)})$$
$$\mathbf{P2}^{\sim}(s) = \frac{s}{s+\mu}$$
$$\mathbf{P21}^{\sim}(s) = \frac{\mu}{s+\mu}$$

An example: M/G/1/2/2 queue with different classes of customers

PRI service policy - transient analysis

Measures of the subordinated process:

$$P_{11}^{\sim}(s,w) = \frac{s(s+\mu)(1-e^{-w(s+\lambda)})}{(s+\lambda)(s+\mu)-\lambda\mu}$$
$$P_{12}^{\sim}(s,w) = \frac{s\lambda(1-e^{-w(s+\lambda)})}{(s+\lambda)(s+\mu)-\lambda\mu}$$
$$F_{11}^{\sim}(s,w) = \frac{(s+\mu)(s+\lambda)e^{-w(s+\lambda)}}{(s+\lambda)(s+\mu)-\lambda\mu}$$
$$\mathbf{C}^{\sim}(s,w) = 0$$

Kernel elements are obtained after unconditioning with respect to the firing time w.

An example: M/G/1/2/2 queue with different classes of customers

PRD service policy - steady state analysis

Exponential regeneration period starting from s_1

$$\alpha_{11} = \frac{1}{2\lambda}\ ; \quad \pi_{12} = \frac{1}{2}\ ; \quad \pi_{13} = \frac{1}{2}$$

PRD-CTMC regeneration period starting from s_2

$$\alpha_{22}|_{\gamma_2=w} = \int_0^w e^{-x\lambda}dx = \frac{1}{\lambda}(1-e^{-w\lambda})$$

t_2 fires

$$\pi_{21}|_{\gamma_2=w} = e^{-w\lambda}$$

t_2 becomes disabled (t_3 fires before)

$$\pi_{24}|_{\gamma_2=w} = 1-e^{-w\lambda}$$

An example: M/G/1/2/2 queue with different classes of customers

PRS service policy - steady state analysis

PRS-CTMC regeneration period starting from s_2
$\beta = 0$ and $\mathrm{L}_\beta(w) = w$

$$\alpha_{22}|_{\gamma_2 = w} = w$$

$$\alpha_{24}|_{\gamma_2 = w} = w\lambda/\mu$$

$$\pi_{21}|_{\gamma_2 = w} = e^{-w\beta} = 1$$

PRI service policy - steady state analysis

PRI-CTMC regeneration period starting from s_2

$$\alpha_{22}|_{\gamma_2 = w} = [e^{-w\lambda}]^{-1}\ 1/\lambda(1 - e^{-w\lambda}) = 1/\lambda(e^{w\lambda} - 1)$$

$$\alpha_{24}|_{\gamma_2 = w} = [e^{-w\lambda}]^{-1}\ 1/\mu(1 - e^{-w\lambda}) = 1/\mu(e^{w\lambda} - 1)$$

$$\pi_{21}|_{\gamma_2 = w} = [e^{-w\lambda}]^{-1} e^{-w\lambda} = 1$$

Exponential service case with *prd* or *prs* preemption

If λ is the EXP thinking time of t_1 and t_3 and μ the EXP service time of t_4 and t_2, the *prd* and the *prs* service discipline have the same effect.

The marking process is an ordinary CTMC with generator:

$$\mathbf{A} = \left| \begin{array}{cccc} -2\lambda & \lambda & \lambda & 0 \\ \mu & -(\lambda+\mu) & 0 & \lambda \\ \mu & 0 & -(\lambda+\mu) & \lambda \\ 0 & \mu & 0 & -\mu \end{array} \right|$$

The steady state distribution is given by:

$$v_1 = \frac{\mu}{\mu + 2\lambda + 2\lambda^2}$$

$$v_2 = v_1 \left(\frac{2\lambda}{\mu} - \frac{\lambda}{\lambda + \mu} \right)$$

$$v_3 = v_1 \frac{\lambda}{\lambda + \mu}$$

$$v_4 = v_1 \frac{2\lambda^2}{\mu}$$

Exponential service case with *pri* preemption

λ is the thinking time of t_1 and t_3 and μ the service time of t_4 and t_2. The subordinated process starting from state s_2 is a CTMC with generator:

$$\mathbf{A}^{s_2} = \begin{array}{|c|c|} \hline -\lambda & \lambda \\ \hline \mu & -\mu \\ \hline \end{array}$$

By applying the suitable formulas, we get:

$$\alpha_{s_2,s_2}(w) = \frac{1 - e^{-\lambda w}}{\lambda\, e^{-\lambda w}} \qquad \alpha_{s_2,s_4}(w) = \frac{1 - e^{-\lambda w}}{\mu\, e^{-\lambda w}}$$

By unconditioning with respect to w:

$$\alpha_{s_2,s_2} = \frac{1}{\mu - \lambda} \qquad \alpha_{s_2,s_4} = \frac{\lambda}{\mu(\mu - \lambda)}$$

The complete α values are:

$$\alpha = \left| \begin{array}{cccc} 1/2\lambda & 0 & 0 & 0 \\ 0 & 1/(\mu-\lambda) & 0 & \lambda/(\mu^2 - \lambda\mu) \\ 0 & 0 & 1/(\lambda+\mu) & 0 \\ 0 & 0 & 0 & 1/\mu \end{array} \right|$$

Exponential service case with *pri* preemption

$$\mathbf{\Pi} = \lim_{t \to \infty} \mathbf{K}(t) = \left| \begin{array}{cccc} 0 & 0.5 & 0.5 & 0 \\ 1 & 0 & 0 & 0 \\ \mu/(\lambda+\mu) & 0 & 0 & \lambda/(\lambda+\mu) \\ 0 & 1 & 0 & 0 \end{array} \right|$$

By solving the embedded DTMC we get:

$$p_{s_1} = \frac{\lambda + \mu}{3\lambda + 2\mu} \quad ; \quad p_{s_2} = \frac{2\lambda + \mu}{2(3\lambda + 2\mu)}$$

$$p_{s_3} = \frac{\lambda + \mu}{2(3\lambda + 2\mu)} \quad ; \quad p_{s_4} = \frac{\lambda}{2(3\lambda + 2\mu)}$$

and finally v_i can be evaluated:

$$v_{s_1} = \frac{\mu(\mu - \lambda)}{(\mu^2 + \lambda\mu + \lambda^2)}$$

$$v_{s_2} = \frac{\lambda\mu(2\lambda + \mu)}{(\lambda + \mu)(\mu^2 + \lambda\mu + \lambda^2)}$$

$$v_{s_3} = \frac{\lambda\mu(\mu - \lambda)}{(\lambda + \mu)(\mu^2 + \lambda\mu + \lambda^2)}$$

$$v_{s_4} = \frac{\lambda^2(\lambda + 2\mu)}{(\lambda + \mu)(\mu^2 + \lambda\mu + \lambda^2)}$$

Comparative Results

Assuming a constant value for the service time $\mu = 1$, the following values as a function of λ are obtained:

	$v_{s1} + v_{s3}$		$v_{s2} + v_{s4}$	
λ	*prs*	*pri*	*prs*	*pri*
0.01	0.980	0.980	0.020	0.020
0.1	0.894	0.885	0.105	0.115
0.5	0.533	0.381	0.466	0.619
0.99	0.302	0.005	0.698	0.995

Conclusion

The preemption policy has to be carefully considered in stochastic modelling. It significantly effects the stochastic behaviour of systems.

References

[1] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2:93–122, 1984.

[2] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *Lecture Notes in Computer Science*, volume 266, pages 132–145. Springer Verlag, 1987.

[3] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, SE-15:832–846, 1989.

[4] A. Bertoni and M. Torelli. Probabilistic Petri nets and semi Markov processes. In *Proceedings 2-nd European Workshop on Petri Nets*, 1981.

[5] A. Bobbio and M. Telek. A benchmark for PH estimation algorithms: results for Acyclic-PH. *Stochastic Models*, 10:661–677, 1994.

[6] A. Bobbio and M. Telek. Computational restrictions for SPN with generally distributed transition times. In D. Hammer K. Echtle and D. Powell, editors, *First European Dependable Computing Conference (EDCC-1), Lecture Notes in Computer Science*, volume 852, Springer-Verlag, pages 131–148, 1994.

[7] A. Bobbio and M. Telek. Markov regenerative SPN with non-overlapping activity cycles. In *International Computer Performance and Dependability Symposium - IPDS95*, pages 124–133. IEEE CS Press, 1995.

[8] A. Bobbio, V.G. Kulkarni, A. Puliafito, M. Telek and K. Trivedi. Preemptive repeat identical transitions in Markov Regenerative Stochastic Petri Nets. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, pages 113–124. IEEE CS Press, 1995.

[9] P. Chen, S.C. Bruell, and G. Balbo. Alternative methods for incorporating non-exponential distributions into stochastic timed Petri nets. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM89*, pages 187–197. IEEE Computer Society, 1989.

[10] G. Chiola. *GreatSPN* 1.5 Software architecture. In G. Balbo and G. Serazzi, editors, *Computer Performance Evaluation*, pages 121–136. Elsevier Science Publishers, 1992.

[11] Hoon Choi, V.G. Kulkarni, and K. Trivedi. Transient analysis of deterministic and stochastic Petri nets. In *Proceedings of the 14-th International Conference on Application and Theory of Petri Nets*, Chicago, June 1993.

[12] H. Choi, V.G. Kulkarni, and K. Trivedi. Markov regenerative stochastic Petri nets. *Performance Evaluation*, 20:337–357, 1994.

[13] G. Ciardo, J. Muppala, and K.S. Trivedi. SPNP: stochastic Petri net package. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM89*, pages 142–151. IEEE Computer Society, 1989.

[14] G. Ciardo and C. Lindemann. Analysis of deterministic and stochastic Petri nets. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM93*, pages 160–169. IEEE Computer Society, 1993.

[15] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Transactions on Software Engineering*, 20:506–515, 1994.

[16] E. Cinlar. *Introduction to Stochastic Processes.* Prentice-Hall, Englewood Cliffs, 1975.

[17] J.A. Couvillon, R. Freire, R. Johnson, W.D. Obal, M.A. Qureshi, M. Rai, W. Sanders, and J.E. Tvedt. Performability modeling with UltraSAN. *IEEE Software*, 8:69–80, September 1991.

[18] D.R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. *Proceedings of the Cambridge Phylosophical Society*, 51:433–440, 1955.

[19] A. Cumani. Esp - A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In *Proceedings International Workshop Timed Petri Nets*, pages 144–151, Torino (Italy), 1985. IEEE Computer Society Press no. 674.

[20] J. Bechta Dugan, K. Trivedi, R. Geist, and V.F. Nicola. Extended stochastic Petri nets: applications and analysis. In *Proceedings PERFORMANCE '84*, Paris, 1984.

[21] G. Florin and S. Natkin. Les reseaux de Petri stochastiques. *Technique et Science Informatique*, 4:143–160, 1985.

[22] R. German, C. Kelling, A. Zimmermann, and G. Hommel. *TimeNET - A toolkit for evaluating non-markovian stochastic Petri nets.* Report No. 19 - Technische Universität Berlin, 1994.

[23] R. German and C. Lindemann. Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation*, 20:317–335, 1994.

[24] R. German. New results for the analysis of deterministic and stochastic Petri nets. In *International Computer Performance and Dependability Symposium - IPDS95.* IEEE CS Press, pages 114–123, 1995.

[25] R. German, D. Logothetis and K. Trivedi. Transient analysis of Markov Regenerative Stochastic Petri Nets: a comparison of approaches. In *6-th International Conference on Petri Nets and Performance Models - PNPM95*, IEEE CS Press, pages 103–112, 1995.

[26] P.J. Haas and G.S. Shedler. Regenerative stochastic Petri nets. *Performance Evaluation*, 6:189–204, 1986.

[27] P.J. Haas and G.S. Shedler. Stochastic Petri nets with simultaneous transition firings. In *Proceedings International Workshop on Petri Nets and Performance Models - PNPM87*, pages 24–32. IEEE Computer Society, 1987.

[28] P.J. Haas and G.S. Shedler. Stochastic Petri nets with timed and immediate transitions. *Stochastic Models*, 5:563–600, 1989.

[29] R. Lepold. PEPNET: A new approach to performability modelling using stochastic Petri nets. In *Proceedings 1st International Workshop on Performability Modelling of Computer and Communication Systems*, pages 3–17, University of Twente - Enschede (NL), 1991.

[30] C. Lindemann. An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. *Performance Evaluation*, 18:75–95, 1993.

[31] C. Lindemann. DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri nets. *Performance Evaluation*, 22:3–21, 1995.

[32] M.K. Molloy. On the integration of delay and throughput measures in distributed processing models. Technical report, Phd Thesis, UCLA, 1981.

[33] S. Natkin. Les reseaux de Petri stochastiques et leur application a l'evaluation des systemes informatiques. Technical report, These de Docteur Ingegneur, CNAM, Paris, 1980.

[34] M.F. Neuts. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, 1981.

[35] A. Reibman, R. Smith, and K.S. Trivedi. Markov and Markov reward model transient analysis: an overview of numerical approaches. *European Journal of Operational Research*, 40:257–267, 1989.

[36] M. Telek and A. Bobbio. Markov regenerative stochastic Petri nets with age type general transitions. In G. De Michelis and M. Diaz, editors, *Application and Theory of Petri Nets (16-th International Conference)*, Lecture Notes in Computer Science, volume 935, Springer-Verlag, pages 471–489, 1995.

[37] M. Telek, A. Bobbio, L. Jereb, A. Puliafito and K. Trivedi. Steady state analysis of Markov regenerative SPN with age memory policy. In H. Beilner and F. Bause, editors, *8-th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, Lecture Notes in Computer Science, volume 977, Springer-Verlag, pages 165–179, 1995.

[38] M. Telek, A. Bobbio and A. Puliafito. Steady State Solution of MRSPN with mixed preemption policies. In *International Computer Performance and Dependability Symposium - IPDS96*, IEEE Computer Society Press, pages 106–115, 1996.