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 \( R(M_0) \) 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 $T_E$ is a connected path in the reachability graph $R(M)$ 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 $T_E$.

$$T_E = \{(\tau_0, M(0)); (\tau_1, M(1)); \ldots; (\tau_i, M(i)); \ldots\}$$

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, $T_E$ is mapped into a stochastic process $M(t)$, $t \geq 0$, called the Marking Process.

PNs 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 [Ajamone 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 $\gamma_k$ 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

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 $\gamma_g$ (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 \( \gamma_g \):
  - the firing time is resampled;
  - the firing time is maintained.

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 \( \gamma_g \) maintains its value (i.e. the resampling time counts the time from which the current value of the firing time level was set).

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 (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 \( \gamma_g \) 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 \( \gamma_A \), 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 \( M(t) \) 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 \( M(t) \) 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 $\gamma_k$ associated to the timed $PN$ transitions are exponentially distributed.

The marking process $M(t)$ 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 $\gamma_k$. The $PH$ model has $\nu_k$ stages with a single initial stage numbered stage 1 and a single final stage numbered stage $\nu_k$.

- 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 $M(t)$ generated by the $GDT_{SPN}$ over the reachability set $R(M_0)$ 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 largeness 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 $\tau^*_n$ of entrance in a tangible marking $M_n$ all the memory variables $a_k$ are equal to zero, and the firing times $\gamma_k$ resampled, $\tau^*_n$ is a regeneration time point for the marking process $M(t)$.

$$\mathcal{T}_E = \{(\tau^*_0, M(0)); (\tau^*_1, M(1)); \ldots; (\tau^*_n, M(n)); \ldots\}$$

The embedded sequence of regeneration time points and associated states $(\tau^*_n, M(n))$ is a Markov renewal sequence and the marking process $M(t)$ 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. - PNPM 95].

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 $\gamma_g$.

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 $\gamma_g$ 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 $\gamma_g$ 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 $a_2$ is reset each time $t_2$ is disabled or fires.
- The firing time $\gamma_2$ is resampled only upon firing of the transition.

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

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 $s_4$ 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.

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 vice versa.

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

Process subordinated to $s_2$

(the policy of $t_2$ is prs)

Transition $t_2$ is the dominant transition.

The process $M^{(2)}(x)$ 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.

$M^{(2)}(x)$ is semi-Markov process since $t_4$ is GEN.
Process subordinated to $s_2$
(the policy of $t_2$ is pri)

\[ \begin{array}{c}
  s_1 \\
  s_2 \\
  s_3 \\
  s_4 \\
  t
\end{array} \]

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.

Solution techniques

Applicability of approaches in transient and steady state analysis

<table>
<thead>
<tr>
<th></th>
<th>prd</th>
<th>prs</th>
<th>pri</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRA in time domain</td>
<td>X</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>SVA</td>
<td>X</td>
<td>$X^*$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>MRA in transform domain</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PHA</td>
<td>X</td>
<td>X</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

$X^*$ possible, but never considered

Expected limits of further generalization:
- MRA: overlapping active/sampled intervals
- SVA: pri memory policy
- PHA: pri memory policy

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.

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]:

\[ K(t) - [K_{ij}(t)]; K_{ij}(t) = Pr\{M(t) = j, \tau_i^r \leq t \}; M(0) = i \]

\[ E(t) = [E_{ij}(t)]; E_{ij}(t) = Pr\{M(t) = j, \tau_i^r > t \}; M(0) = i \]

Matrix \( K(t) \) is the global kernel; describes the next regeneration time point and state.
Matrix \( E(t) \) is the local kernel; describes the transient behaviour of subordinated processes.

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

Subordinated process without internal state transition

When only exponential transitions are enabled in a marking \( i \):

\[ K_{ij}(t) = \sum_{d \text{ enabled in } i} \frac{\lambda^d}{\lambda^i} (1 - e^{-\lambda^i t}) \Delta^d_{ij} \]

\[ E_{ij}(t) = \delta_{ij} e^{-\lambda^i t} \]

where \( \lambda^i = \sum_{d \text{ enabled in } i} \lambda^d \) and \( \delta_{ij} \) 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

\[ K_{ij}(t) = \sum_{d \text{ enabled in } i} \int_{t_{ij}}^{T} \prod_{e \text{ in } i; e \neq d} (1 - F_e(t)) \, dF^e(t) \Delta^d_{ij} \]

\[ E_{ij}(t) = \delta_{ij} (1 - F^i(t)) \]

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.

Structure of the state space with different dominant transitions:

<table>
<thead>
<tr>
<th>transition</th>
<th>structure of state space</th>
</tr>
</thead>
<tbody>
<tr>
<td>prd</td>
<td>no transition to $D1$ and $D2$</td>
</tr>
<tr>
<td>pri</td>
<td>no transition to $D1$ and $C$</td>
</tr>
<tr>
<td>prs</td>
<td>no transition to $D2$ and $C$</td>
</tr>
</tbody>
</table>

Transient analysis with MRA

Analysis of the subordinated process over the partitioned state space.

Suppose the subordinated process is a CTMC with generator $A$ with state numbers ordered according to $E$, $D$, $C$ we have:

$$A = \begin{bmatrix} \dot{A}_E & \dot{A}_{ED} & \dot{A}_{EC} \\ \dot{A}_{DE} & \dot{A}_D & 0 \\ \cdot & \cdot & \cdot \end{bmatrix}$$

By which:

- $P1^-(s, v) = \frac{s}{v} ((s + v)I - \dot{A}_E)^{-1}$
- $F1^-(s, v) = ((s + v)I - \dot{A}_E)^{-1}$
- $C1^-(s, v) = \frac{1}{v} ((s + v)I - \dot{A}_C)^{-1} \dot{A}_{EC}$
- $P12^-(s, v) = \frac{1}{v} ((s + v)I - \dot{A}_E)^{-1} \dot{A}_{ED}$
- $P2^-(s) = s (sI - \dot{A}_D)^{-1}$
- $P21^-(s) = (sI - \dot{A}_D)^{-1} \dot{A}_{D\ell}$

Transient analysis with MRA

Measures of the subordinated process over the partitioned state space.

Let us fix the value of the firing time $\gamma_0 = w$.

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

- $P1(t, w)$ - state transition probabilities inside $E$ before leaving $E$
- $F1(t, w)$ - probability of firing before $t$ before leaving $E$
- $C1(t, w)$ - probability of concluding before $t$ before leaving $E$
- $P12(t, w)$ - probability of transition to $D$ before $t$ before leaving $E$
- $P2(t)$ - state transition probabilities inside $D$ before leaving $D$
- $P21(t)$ - probability of transition to $E$ before $t$
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:

\[
K_{ij}(t) = \int_{0}^{\infty} \sum_{k \in S} F_{ik}(t, w) \Delta_{kj} + C_{ik}(t, w) dF_{d}(w)
\]

\[
E_{ij}(t) = \int_{0}^{\infty} P_{ij}(t, w) dF_{d}(w)
\]

where \( F_{d}(\cdot) \) is CDF of the firing time of the dominant transition.

Transient analysis with MRA

Step 3: Analysis of the overall marking process

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

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

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

Analysis of the overall subordinated process

\( \text{prd} \) type dominant transition

\[
\mathbf{P}^\sim(s, w) = \mathbf{P}_1^\sim(s, w) \quad \mathbf{F}^\sim(s, w) = \mathbf{F}_1^\sim(s, w) \quad \mathbf{C}^\sim(s, w) = \mathbf{C}_1^\sim(s, w)
\]

\( \text{pri} \) type dominant transition

\[
\mathbf{P}^\sim(s, w) = [\mathbf{I} - \mathbf{P}^{12\sim}(s, w) \mathbf{P}^{21\sim}(s)]^{-1} \quad \mathbf{F}^\sim(s, w) = [\mathbf{I} - \mathbf{P}^{12\sim}(s, w) \mathbf{P}^{21\sim}(s)]^{-1} \mathbf{F}_1^\sim(s, w) \quad \mathbf{C}^\sim(s, w) = 0
\]

\( \text{prs} \) type dominant transition

\[
\mathbf{P}^{\sim\sim}(s, v) = [\mathbf{I} - \mathbf{P}^{12\sim\sim}(s, v) \mathbf{P}^{21\sim}(s)]^{-1} \quad \mathbf{F}^{\sim\sim}(s, v) = [\mathbf{I} - \mathbf{P}^{12\sim\sim}(s, v) \mathbf{P}^{21\sim}(s)]^{-1} \mathbf{F}_1^{\sim\sim}(s, v) \quad \mathbf{C}^{\sim\sim}(s, v) = 0
\]

In the Laplace Stieltjes Transform domain

Step 3: Analysis of the overall marking process

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:

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

\( \alpha_{ij} \) is the expected time a subordinated process starting from state \( i \) spends in state \( j \).

\[ \pi_{ij} = \Pr \{ M(1) = j \mid M(0) = i \} \]

\( \pi_{ij} \) is the probability that the subordinated process starting from state \( i \) is followed by a subordinated process starting from state \( j \).

The matrix \( \Pi = \{ \pi_{ij} \} \) 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:

\[ \alpha_{ij} = \int_{-\infty}^\infty E_{ij}(t) dt = \lim_{s \to 0} 1/s \, E_{ij}^s(s) \]

\[ \pi_{ij} = \lim_{s \to 0} K_{ij}(t) = \lim_{s \to 0} K_{ij}^s(s) \]

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

\( \text{prd type dominant transition} \)

\[ \alpha_i \big|_{g \rightarrow w} = L(w) \]

\[ \pi_i \big|_{g \rightarrow w} = e^w A_i \Delta + [0 \mid L(w) \, A_{\Delta \Delta}] \]

where \( L(w) = \int_0^w e^x A_i \, dx \).

The Steady-State Solution with MRA

Step 3: Analysis of the overall process

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

\[ P = P \Pi \quad \sum_i p_i = 1 \]

The steady-state probabilities of the MRGP become:

\[ \nu_j = \lim_{l \to \infty} \Pr \{ M(t) = j \} = \frac{\sum_{i} p_{i} \alpha_{ij}}{\sum_{k} p_{k} \sum_{l} \alpha_{kl}} \]

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

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

However, an "automatic" analytical solution can be obtained by \( \lim_{s \to 0} V^-(s) \).
The Supplementary Variable Approach

The marking process \((M(t))\) 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 \([COX]\).

Transient analysis with SVA

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

Partial differential equation describes the process evolution in \(S^g\)

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

Ordinary differential equation describes the process evolution in \(S^E\)

\[
\frac{d}{dt} E(t) = \pi_E(t)Q_EE + \sum_j \int_0^\infty p_g^h(t, x)dF^h(x)\Delta_{h,j}E + \sum_j \pi_g^h(t)Q_{h,h}E
\]

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).

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 \((A = 0)\)
  - \(S^g\) 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
  \[
  \pi_n(t) = \Pr[M(t) = n]
  \]
- "age rate"
  \[
  p_n(t, x) = \frac{\Pr[M(t) = n, x < A \leq x + dx]}{dx}, \frac{1}{1 - F^g(x)}
  \]
- firing time distribution \(F^g(x)\)
- branching probability \(\Delta_{h,j}^g\)

Transient analysis with SVA

Boundary condition:

\[
p^g(t, 0) = \pi_E(t)Q_Eg^+ + \sum_h \int_0^\infty p^h(t, x)dF^h(x)\Delta_{h,g}^+ + \sum_h \pi^h(t)Q_{h,g}^+
\]

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:

\[
\pi^g(t) = \int_0^\infty p^g(t, x)(1 - F^g(x))dx
\]

Initial conditions are

- \(\pi_E(0)\)
- \(p^g(0, x) = \pi^g(0)\delta(x)\)
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

\[ p^g(ih, jh) = p^g((i - 1)h, (j - 1)h) e^{Q^g h} \]

and set \( p^g(ih, 0) = 0 \)

2. compute the state probabilities \( \pi^g(ih) \) by \( p^g(ih, jh), \ j = 0, 1, \ldots \)

3. compute the state probabilities \( \pi^E(ih) \) by the ordinary differential equation

4. compute the activation rate of general transitions \( p^g(ih, 0) \) 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: \( p^g(x), \pi^E \)

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

\[ \frac{d}{dx} p^g(x) = p^g(x) Q^g \]

\[ 0 = \pi^E Q^E E_+ + \sum_g \int_0^\infty p^g(x) dF^g(x) \Delta^g E_+ \]

\[ \sum_g \pi^g Q^g E \]

\[ p^g(0) = \pi^E Q^E g_+ + \sum_g \int_0^\infty p^h(x) dF^h(x) \Delta^h g_+ \]

\[ \sum_g \pi^h Q^h g \]

\[ \pi^g = \int_0^\infty p^g(x) (1 - F^g(x)) dx \]

Steady state analysis with SVA

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

\[ 0 = \pi^E Q^E E_+ + \sum_g \int_0^\infty p^g(0) \Omega^g \Delta^g E_+ \]

\[ \sum_g \pi^g Q^g E \]

\[ p^g(0) = \pi^E Q^E g_+ + \sum_g \int_0^\infty p^h(0) \Omega^h \Delta^h g_+ \]

\[ \sum_g \pi^h Q^h g \]

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

\[ \pi^g = p^g(0) \Psi^g \]
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 vice versa.

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 ($t_2$)

PRD service policy - transient analysis

Measures of the subordinated process:

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

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

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

Kernel elements:

$E_{22}^{-}(s) = \int_0^\infty P_{11}^{-}(s, w)dF_2(w)$

$K_{21}^{-}(s) = \int_0^\infty F_{11}^{-}(s, w)dF_2(w)$

$K_{24}^{-}(s) = \int_0^\infty C_{11}^{-}(s, w)dF_2(w)$

PRS service policy - transient analysis

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

PRS-CTMC regeneration period starting from $s_2$:

$A = \begin{bmatrix} \bar{A}_F & \bar{A}_{FD} \\ \bar{A}_{DG} & \bar{A}_{D} \end{bmatrix} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$

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

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

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

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

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

$P21^{-}(s) = \frac{\mu}{s + \mu}$
An example: M/G/1/2/2 queue with different classes of customers

**PRS service policy - transient analysis**

Measures of the subordinated process:

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

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**

Measures of the subordinated process:

\[ P_{11}^{\text{PRI}}(s, v) = \frac{s(s + \mu)}{(s + v + \lambda)(s + \mu) - \lambda \mu} \]
\[ P_{12}^{\text{PRI}}(s, v) = \frac{\lambda}{(s + v + \lambda)(s + \mu) - \lambda \mu} \]
\[ F_{11}^{\text{PRI}}(s, v) = \frac{v(s + \mu)}{(s + v + \lambda)(s + \mu) - \lambda \mu} \]
\[ C^{\text{PRI}}(s, v) = 0 \]

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:

\[ P_{11}^{\text{PRI}}(s, w) = s \int_0^w e^{-x(s+\lambda)} \, dx = \frac{s}{s+\lambda} \left( 1 - e^{-w(s+\lambda)} \right) \]
\[ P_{12}^{\text{PRI}}(s, w) = e^{-w(s+\lambda)} \]
\[ P_{11}^{\text{PRI}}(s, w) = \lambda \int_0^w e^{-x(s+\lambda)} \, dx = \frac{\lambda}{s+\lambda} \left( 1 - e^{-w(s+\lambda)} \right) \]
\[ P_{21}^{\text{PRI}}(s) = \frac{s}{s+\mu} \]

PRI 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 service policy - steady state analysis**

Exponential regeneration period starting from \( s_2 \)

\[ \alpha_{22} = \int_0^w e^{-x\lambda} \, dx = \frac{1}{\lambda} (1 - e^{-w\lambda}) \]
\[ t_{2 \text{ fires}} \]
\[ \pi_{21}^{t_{2 \text{ disabled}}} = e^{-w\lambda} \]
\[ t_{2 \text{ becomes disabled (t}_{3 \text{ fires before})}} \]
\[ \pi_{21}^{t_{3 \text{ fires before}}} = 1 - e^{-w\lambda} \]
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 $L_\beta(w) = w$

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

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

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

**PRI service policy - steady state analysis**

**PRI-CTMC** regeneration period starting from $s_2$

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

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

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

**Exponential service case with pri preemption**

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

$$A^{\alpha} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

By applying the suitable formulas, we get:

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

$$\alpha_{s_1,s_2} |_{\gamma_w = w} = \frac{1 - e^{-\lambda w}}{\lambda e^{-\lambda w}}$$

By unconditioning with respect to $w$:

$$\alpha_{s_1,s_2} = \frac{1}{\mu - \lambda}$$

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

The complete $\alpha$ values are:

$$\alpha = \begin{bmatrix} 1/2\lambda & 0 & 0 & 0 \\ 0 & 1/(\mu - \lambda) & 0 & \lambda/\mu(\mu^2 - \lambda\mu) \\ 0 & 0 & 1/(\lambda + \mu) & 0 \\ 0 & 0 & 0 & 1/\mu \end{bmatrix}$$

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

If $\lambda$ is the EXP thinking time of $t_1$ and $t_3$ and $\mu$ 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:

$$A = \begin{bmatrix} -2\lambda & \lambda & \lambda & 0 \\ \mu & -/(\lambda + \mu) & 0 & \lambda \\ \mu & 0 & -/(\lambda + \mu) & \lambda \\ 0 & \mu & 0 & -\mu \end{bmatrix}$$

The steady state distribution is given by:

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

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

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

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

By solving the embedded DTMC we get:

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

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

and finally $v_1$ can be evaluated:

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

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

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

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

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

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$v_{p1} + v_{p3}$</th>
<th>$v_{p2} + v_{p4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.980</td>
<td>0.020</td>
</tr>
<tr>
<td>0.1</td>
<td>0.894</td>
<td>0.105</td>
</tr>
<tr>
<td>0.5</td>
<td>0.533</td>
<td>0.466</td>
</tr>
<tr>
<td>0.99</td>
<td>0.302</td>
<td>0.698</td>
</tr>
</tbody>
</table>

Conclusion

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

References


