

Phase-Type Distributions in Stochastic Automata Networks*

I. Sbeity⁽¹⁾, L. Brenner⁽¹⁾, B. Plateau⁽¹⁾ and W.J. Stewart⁽²⁾

December 20, 2005

- (1) Informatique et Distribution - 51, Av. Jean Kuntzman, 38330 Montbonnot, France.
(2) Department of Computer Science, NC State University, Raleigh, NC, USA.

Abstract

Stochastic Automata Networks (SANS) are high-level formalisms for modeling very large and complex Markov chains in a compact and structured manner. To date, the exponential distribution has been the only distribution used to model the passage of time in the evolution of the different SAN components. In this paper we show how *phase-type* distributions may be incorporated into SANS thereby providing the wherewithal by which arbitrary distributions can be used which in turn leads to an improved ability for more accurately modeling numerous real phenomena.

The approach we develop is to take a SAN model containing phase-type distributions and to translate it into another, stochastically equivalent, SAN model having only exponential distributions. In the SAN formalism, it is the events that are responsible for firing transitions and our procedure is to associate a stochastic automaton with each event having a phase-type distribution. This automaton models the distribution of time until the event occurs. Events having phase-type distributions are called phase-type events. In translating the original SAN model with phase-type events, into a new equivalent SAN model, all phase-type events are eliminated and replaced with exponential events which translate the interactions among automata associated with phase-type events and the remaining automata, as well as the scheduling policy of the phase-type distribution. In this way, the size of the elementary matrices remain small, because the size of the automata are small: the automata are either those of the original SAN, or are those associated with the phase-type events and are of size k , the number of phases in the representation of the distribution.

The major difficulty in transforming a PH-SAN into a regular SAN arises from the fact that the firing of some events may *cancel* (or disable) certain phase-type events: worse, certain types of disabling event behave differently in different circumstances, in the sense that different sets of censored events can result from the firing of a single disabling event in a given state. The first step of the transformation procedure of a PH-SAN to a SAN is to remove this difficulty. Later, a second stage converts a PH-SAN without such events into a regular SAN.

*This work is supported in part by ACI Sure-Paths project, in part by the NSF under research grant number ACI 0203971 and, in part by CAPES/Brazil.

1 Introduction

Stochastic Automata Networks (SANs) [2, 8, 13] are high-level formalisms for modeling very large and complex Markov chains in a compact and structured manner. To date, the exponential distribution has been the only distribution used to model the passage of time in the evolution of the different SAN components: a mathematical expediency that facilitates our ability to model many SAN activities concurrently. Nevertheless, the use of more general distributions remains a desirable objective for more accurately modeling numerous real phenomena. It is in this regard that *phase-type* distributions fill an important role. Such distributions are usually structured as a passage through a set of exponential phases and may be used to approximate, arbitrarily closely, any other distribution. Phase-type distributions have been used by Neuts [12] and others for modeling queueing networks, and by many authors in modeling stochastic Petri nets [1, 4–6, 9, 10].

Previous research on the use of phase-type distributions in stochastic Petri nets (SPN), have followed two different paths. The first approach [1, 5] determines the conditions under which the stochastic process of a SPN is regenerative and defines a set of regeneration points. The behavior of the system between two regeneration points is then analyzed and with the identification of an *embedded* Markov chain, the stationary distribution may be computed. However, this approach lack generality, a result of the conditions imposed upon the SPN to make it regenerative.

The second approach is to introduce phase-type distributions into the SPN and to directly generate and analyze the Markov chain without resorting to regeneration points and an embedded chain. Among the research carried out along this second direction, one possibility is that represented by the work of Molloy and others [4, 10] whereby phase-type transitions in an SPN model are replaced by sub-nets that mimics their stochastic behavior. A second possibility is to introduce phase-type distributions directly into a model, which yields a stochastic Petri nets *and* separately described phase-type distributions, and to compute required solutions from the *generated Markov chain*. In this case, the phase-type characteristics are taken into account during the generation of the reachability graph of the model [6] by considering the semantic of each phase-type transition and directly analyzing the chain. However, this method is ineffective when the state space is very large, because the chain is explicitly described.

Recent research [7, 9], instigated by the increased complexity of the Markov chain generated from a stochastic Petri net having phase-type distributions, has sought to form a compact tensor representation of the generator of the Markov chain, thereby avoiding the necessity of forming the transition rate matrix explicitly. The construction of a tensor representation is based on a partition of the state space according to what have been called *Symmetric Structural Conflicts (SSC)* between transitions. Two transitions are said to be in symmetric structural conflict if and only if the firing of one renders the other unfireable. However, this approach can result in the undesirable effect of causing the elementary matrices to become large if the partitions of the state space contain multiple phase-type transitions, especially when these phase-type transitions are in structural conflict with each other.

The approach we develop in this paper is similar to the approach above in the sense that we give a compact tensor representation of the Markov chain defined by a SAN with phase type distribution. Our approach is to take a SAN model containing phase-type distributions and to translate it into another SAN model having only exponential distributions and additional automata. In the SAN formalism, it is the events that are responsible for causing transitions to occur and we shall associate a stochastic automaton with each event having a phase-type distribution. This automaton models the distribution of time until the event occurs. Events having phase-type distributions are called *PH events*. In translating the original SAN model with

PH events into a new model, all PH events are eliminated and replaced with exponential events which translate the interactions among automata associated with PH events and the remaining automata and also take the scheduling policy of the phase-type distributions into account. In this way, the size of the elementary matrices remain small, because the size of the automata are small: the automata are either those of the original SAN, or are those associated with the PH events which are of size k , the number of phases in the representation of the distribution.

A general algorithm which implements this translation procedure is presented in this paper. A complexity analysis is also provided. The algorithm itself is currently being incorporated into the PEPS software package.

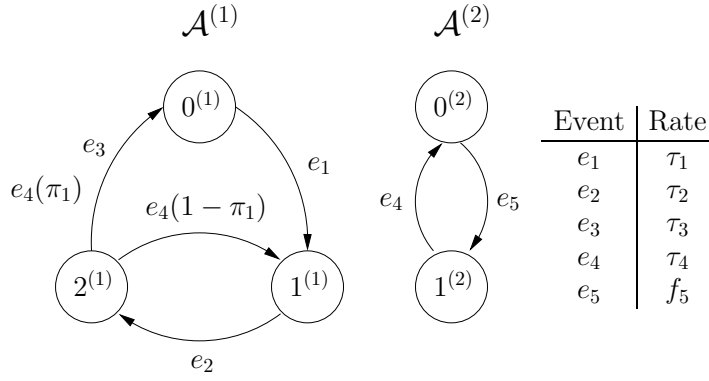
2 Stochastic Automata Networks

2.1 Informal Description

Stochastic Automata Networks, SANs, were first proposed by Plateau in 1985 [13]. The SAN formalism enables a complete system to be represented as a collection of interacting subsystems. Each subsystem is represented by an automaton which is simply a directed and labeled graph whose states are referred to as *local states*, being local to that subsystem, and whose edges, relating local states to one another, are labeled with probabilistic and event information. The different subsystems apply this label information to enable them to interact with each other and to coordinate their behavior.

The states of a SAN are defined by the cartesian product of the *local states* of the automata and are called the *global states* of the SAN. Thus, a global state may be described by a vector whose i^{th} component denotes the local state occupied by the i^{th} automaton. The global state of a SAN is altered by the occurrence (referred to as the *firing*) of an *event*. Each event has a unique *identifier* and a *firing rate*. At any moment, multiple events may be enabled to fire (we shall also use the word *fireable* to describe events that are enabled): the one which actually fires is determined in a Markovian fashion, i.e., from the relative firing rates of those which are enabled. The firing of an event changes a given global *source* state into a global *destination* state. An event may be one of two different types. A *local event* causes a change in one automaton only, so that the global source and destination states differ in one component (local state) only. A *synchronizing event*, on the other hand, can cause more than one automaton to simultaneously change its state with the result that the global source and destination states may differ in multiple components. Indeed, each synchronizing event is associated with multiple automata and the occurrence of a synchronizing event forces all automata associated with the event to simultaneously change state in accordance with the dictates of this synchronizing event on each implicated automata. Naturally, a synchronizing event must be enabled in *all* of the automata on which it is defined before it can fire.

Transitions from one local state to another within a given automaton are not necessarily in one-to-one correspondence with events: several different events may occasion the same local transition. Furthermore, the firing of a single event may give rise to several possible destinations on leaving a local source state. In this case, *routing probabilities* must be associated with the different possible destinations. Routing probabilities may be omitted only if the firing of an event gives rise to a transition having a single destination. Also, automata may interact with one another by means of *functional rates*: the firing rate of any event may be expressed, not only as a constant value (a positive real number), but also as a function of the state of other automata. Functional rates are defined within a single automaton, even though their parameters involve the states of other automata.



$$f_5 = [(st(\mathcal{A}^{(1)} == 0^{(1)}) * \lambda_1] + [(st(\mathcal{A}^{(1)} == 2^{(1)}) * \lambda_2]$$

Figure 1: Example of a SAN model

As an example of the previous discussion, Figure 1 presents a SAN model with two automata, $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$, the first with 3 local states, $0^{(1)}$, $1^{(1)}$ and $2^{(1)}$ and the second with two local states, $0^{(2)}$ and $1^{(2)}$. The model contains four local events, e_1, e_2, e_3 and e_5 and one synchronizing event, e_4 . When automaton $\mathcal{A}^{(1)}$ is in local state $0^{(1)}$, and $\mathcal{A}^{(2)}$ is in local state $0^{(2)}$, (global state $[0, 0]$), two events are eligible to fire, namely e_1 and e_5 . The event e_1 fires at rate τ_1 . This is taken to mean that the random variable which describes the time t from the moment that automaton $\mathcal{A}^{(1)}$ moves into state $0^{(1)}$ until the event e_1 fires, taking it into state $1^{(1)}$, is exponentially distributed with probability density function given by $\tau_1 e^{-\tau_1 t}$. Similar remarks hold for the firing rate of the other events. The firing of e_1 when the system is in global state $[0, 0]$ moves it to global state $[1, 0]$ in which e_5 is still eligible to fire, along now with event e_2 . The event e_1 cannot fire from this new state. The synchronizing event e_4 is enabled in global state $[2, 1]$ and when it fires it changes automaton $\mathcal{A}^{(2)}$ from state $1^{(2)}$ to state $0^{(2)}$ while simultaneously changing automaton $\mathcal{A}^{(1)}$ from state $2^{(1)}$ to either state $0^{(1)}$, with probability π_1 , or to state $1^{(1)}$ with probability $1 - \pi_1$. Observe that two events are associated with the same edge in automaton $\mathcal{A}^{(1)}$, namely e_3 and e_4 . If event e_3 fires, then the first automaton will change from state $2^{(1)}$ to state $0^{(1)}$; if event e_4 fires the first automaton to change from state $2^{(1)}$ to either state $0^{(1)}$ or state $1^{(1)}$ as previously described. There is one functional rate, f_5 , the rate at which event e_5 fires, defined as

$$f_5 = \begin{cases} \lambda_1 & \text{if } \mathcal{A}^{(1)} \text{ is in state } 0^{(1)} \\ 0 & \text{if } \mathcal{A}^{(1)} \text{ is in state } 1^{(1)} \\ \lambda_2 & \text{if } \mathcal{A}^{(1)} \text{ is in state } 2^{(1)} \end{cases}$$

Thus event e_5 , which changes the state of automaton $\mathcal{A}^{(2)}$ from $0^{(2)}$ to $1^{(2)}$, fires at rate λ_1 if the first automaton is in state $0^{(1)}$ or at rate λ_2 if the first automaton is in state $2^{(1)}$. The event e_5 is prohibited from firing if the first automaton is in state $1^{(1)}$. Functional transitions are written more compactly, e.g.,

$$f_5 = [(st(\mathcal{A}^{(1)} == 0^{(1)}) * \lambda_1] + [(st(\mathcal{A}^{(1)} == 2^{(1)}) * \lambda_2]$$

in which conditions such as $st(\mathcal{A}^{(1)} == 2^{(1)})$ (which means “the state of $\mathcal{A}^{(1)}$ is $2^{(1)}$ ”) have the value 1 if the condition is true and are equal to 0 otherwise. This is the notation used to describe

functions in the PEPS software tool [3]. In this setting, the interpretation of a function may be viewed as the evaluation of an expression in the C programming language. The use of functional expressions in SANS is not limited to the rates at which events occur; indeed, probabilities also may be expressed as functions. Figure 2 shows the equivalent Markov chain transition rate diagram for this example.

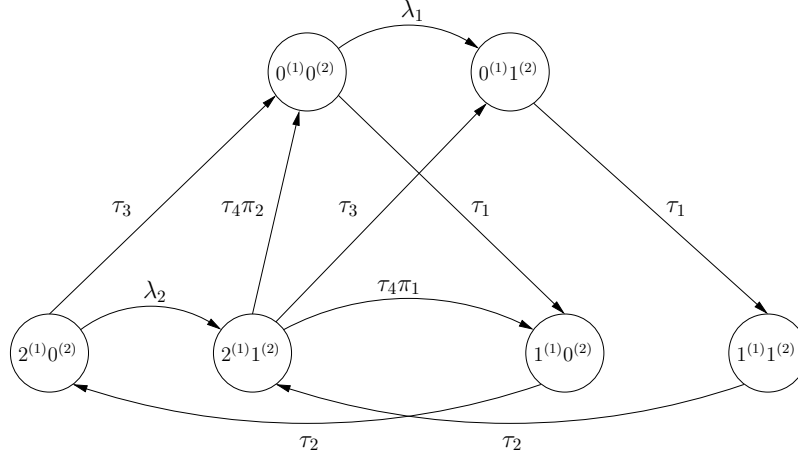


Figure 2: Transition rate diagram of corresponding Markov chain.

We now proceed to give a formal description of a SAN model.

2.2 Formal Description

Formally, a stochastic automata network [2, 8, 13] is defined as the quadruplet, $\mathcal{M} = (N, \mathcal{A}, \mathcal{E}, \hat{f})$, where

- N is the number of automata in the SAN, \mathcal{M} .
- $\mathcal{A} = \{\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(N)}\}$ is the set of automata of the model: each automaton is a directed, labeled graph. Specifically, the i^{th} automaton, $\mathcal{A}^{(i)}$, is defined as $\mathcal{A}^{(i)} = \{S^{(i)}, \text{Edge}^{(i)}, \text{Label}^{(i)}\}$ where
 - $S^{(i)}$ is the set of states of automaton $\mathcal{A}^{(i)}$. Their number is given as $n^{(i)}$, the cardinality of the set $S^{(i)}$. A state $x^{(i)} \in S^{(i)}$ is called a *local* state of automaton $\mathcal{A}^{(i)}$. It follows that the product state space (*PSS*) of the model is $S = S^{(1)} \times S^{(2)} \times \dots \times S^{(N)}$ and a global state, $x \in S$ is given by $x = (x^{(1)}, x^{(2)}, \dots, x^{(N)})$ where $x^{(i)} \in \{1, 2, \dots, n^{(i)}\}$ for $i = 1, 2, \dots, N$.
 - $\text{Edge}^{(i)}$ is a set of directed edges within automaton $\mathcal{A}^{(i)}$. An edge is a directed arc connecting a local source state $x^{(i)}$ to a local destination state $y^{(i)}$ and is written as $(x^{(i)}, y^{(i)})$. The automaton $\mathcal{A}^{(i)}$ can only change from one local state $x^{(i)}$ to another local state $y^{(i)}$ if there is an edge from $x^{(i)}$ to $y^{(i)}$.
 - $\text{Label}^{(i)}$ is the set of labels associated with automaton $\mathcal{A}^{(i)}$. Each edge must have at least one label, and may have several; labels do not exist without an associated edge. Each label on an edge $(x^{(i)}, y^{(i)})$ is written as $\text{Label}^{(i)}(x^{(i)}, y^{(i)}) = \{(e, P_e(x^{(i)}, y^{(i)}))\}$, where e is the identifier of a particular event and $P_e(x^{(i)}, y^{(i)})$ is a function from S (the state space) into $[0, 1]$ (a probability). Thus each label specifies an event e whose

firing permits a transition along the edge $(x^{(i)}, y^{(i)})$ and provides the probability that $\mathcal{A}^{(i)}$ actually changes from state $x^{(i)}$ to $y^{(i)}$ when the event fires. This does not exclude the possibility of having several labels on the edge $(x^{(i)}, y^{(i)})$, each associated with a different event.

- \mathcal{E} is the set of events that can occur in the model. As well as having a unique identifier, an element of \mathcal{E} , i.e., an event $e \in \mathcal{E}$, has three defining characteristics, its type, its master and its rate, collected together as the triple: $(type, master, rate)$. These characteristics are as follows.

- an event's type may be either local or synchronizing, i.e., $type(e) = loc|syn$.
- an event's master is the index of the automaton that is master of the event: i.e., $master(e) = \{1, 2, \dots, N\}$. Each event has a special relationship with a single automaton, called the *master* of the event.
- an event's rate is the rate at which the event e occurs and may be a function from S into \mathbb{R}^+ .

- \hat{f} is the reachability function of the SAN. Its domain is the product state space, $\{x \in S\}$, and its range is the set $\{0, 1\}$. The set of reachable states, RSS, denoted by \hat{S} , is defined as

$$\hat{S} = \{x \in S \mid \hat{f}(x) = 1\}.$$

This completes our formal description of a SAN model. We now turn our attention to transitions that result from the occurrence (or firing) of events. For all events e , we shall let $T^{(i)}(e)$, $i = 1, 2, \dots, N$, denote the set of transitions that can occur in automaton $\mathcal{A}^{(i)}$ as a result of the event e firing. Thus

$$T^{(i)}(e) = \left\{ (x^{(i)}, y^{(i)}) \in S^{(i)} \times S^{(i)} \mid \left(e, P_e(x^{(i)}, y^{(i)}) \right) \in Label^{(i)} \right\},$$

which simply says that $T^{(i)}(e)$ is the set of all edges $(x^{(i)}, y^{(i)})$ in automaton $\mathcal{A}^{(i)}$ that are labeled by the event e . It follows that the set of all automata that are affected by an event e , denoted \mathcal{O} , is given by

$$\mathcal{O}(e) = \left\{ i \in \{1, 2, \dots, N\} \mid T^{(i)}(e) \neq \emptyset \right\}.$$

Notice that, an event e is a local event if and only if $\mathcal{O}(e)$ is a singleton set; otherwise e is a synchronizing event. The set of *all* global transitions that can result from the occurrence of an event e , denoted by $T(e)$, may now be given as

$$T(e) = \left\{ (x, y) \in S \times S \mid (x^{(i)}, y^{(i)}) \in T^{(i)}(e) \text{ for } i \in \mathcal{O}(e) \right\}.$$

We have now defined the set of transitions, $T^{(i)}(e)$, within a specific automaton $\mathcal{A}^{(i)}$ that can occur as a result of an event e firing, and also the set of *all* transitions that can occur as a result of an event e firing, namely $T(e)$.

We shall distinguish events according to whether they are *local* events or *synchronizing* events. The sets of local and synchronizing events are denoted respectively by

- $\mathcal{E}_l = \{e \in \mathcal{E} \mid loc\}$ and
- $\mathcal{E}_s = \{e \in \mathcal{E} \mid syn\}$

It follows that $\mathcal{E} = \mathcal{E}_l \cup \mathcal{E}_s$. We also define the *successors* of local and global states as follows: For any event e , any local state $x^{(i)} \in S^{(i)}$ and any global state $x \in S$, we define

$$\begin{aligned} - \text{succ}_e(x^{(i)}) &= \{y^{(i)} \in S^{(i)} \mid (x^{(i)}, y^{(i)}) \in T^{(i)}(e)\} \quad \text{and} \\ - \text{succ}_e(x) &= \{y \in S \mid (x, y) \in T(e)\} \end{aligned}$$

In words, $\text{succ}_e(x^{(i)})$ is the set of local states of automaton $\mathcal{A}^{(i)}$ which can be reached from local state $x^{(i)}$ when the event e fires and $\text{succ}_e(x)$ is the set of global states that can be reached from global state x when e fires.

Finally, a SAN is said to be *coherent* if the following conditions hold

- For all $i \in \{1, 2, \dots, N\}$ and for all $x^{(i)} \in S^{(i)}$:

$$\sum_{y^{(i)} \in \text{succ}_e(x^{(i)})} P_e(x^{(i)}, y^{(i)}) = 1.$$

The probabilities for choosing local destination states, $y^{(i)} \in \text{succ}_e(x^{(i)})$, when leaving a local state, $x^{(i)}$, must sum to one.

- For all $e \in \mathcal{E}$, there exists a unique $i \in \mathcal{O}(e)$ such that $\text{master}(e) = i$ (by definition) and $T^{(i)}(e)$ is not empty. In other words, each event e has a unique master automaton, $\mathcal{A}^{(i)}$, and the set of transitions generated by e in that master automaton is not empty.
- For all $i \in \{1, 2, \dots, N\}$, for all $x^{(i)}, y^{(i)} \in S^{(i)}$ and for all $e_1, e_2 \in \mathcal{E}$ such that $(e_1, P_{e_1}(x^{(i)}, y^{(i)}))$, $(e_2, P_{e_2}(x^{(i)}, y^{(i)})) \in \text{Label}(x^{(i)}, y^{(i)})$, we have $e_1 \neq e_2$. Events must not be multiply defined.
- \hat{S} is strongly connected. This last condition places a restriction on the reachability function of the SAN.

When, as is generally assumed to be the case, the distribution of firing time of all events in a SAN is exponentially distributed, then the underlying Markovian infinitesimal generator matrix of the SAN, Q , may be written as a sum of Kronecker products [2, 8, 13]. We have

$$Q = \bigoplus_{i=1}^N Q^{(i)} + \sum_{e \in \mathcal{E}_s} \bigotimes_{i=1}^N Q_{e^+}^{(i)} + \bigotimes_{i=1}^N Q_{e^-}^{(i)}.$$

In this representation, $Q^{(i)}$ is the local transition matrix of the automaton $\mathcal{A}^{(i)}$ and is of size $n^{(i)}$. Each matrix $Q^{(i)}$ is formed from \mathcal{E}_l and the corresponding sets $\text{Label}^{(i)}$ and $S^{(i)}$. The value of the element $Q^{(i)}(x^{(i)}, y^{(i)})$ of the matrix $Q^{(i)}$ is given by:

$$Q^{(i)}(x^{(i)}, y^{(i)}) = \sum_{(e, P_e(x^{(i)}, y^{(i)})) \in \text{Label}^{(i)}} \text{rate}(e) \cdot P_e(x^{(i)}, y^{(i)}).$$

For a synchronizing event, $e \in \mathcal{E}_s$, the elements, $Q_{e^+}^{(i)}(x^{(i)}, y^{(i)})$ of the matrix $Q_{e^+}^{(i)}$ of size $n^{(i)}$ are formed as follows. If $i \in \mathcal{O}(e)$, in other words if automaton $\mathcal{A}^{(i)}$ is affected by event e , then

$$Q_{e^+}^{(i)}(x^{(i)}, y^{(i)}) = \begin{cases} \text{rate}(e) \cdot P_e(x^{(i)}, y^{(i)}) & \text{if } \text{master}(e) = i \\ P_e(x^{(i)}, y^{(i)}) & \text{if } \text{master}(e) \neq i \end{cases}$$

In all other cases, i.e., when $i \notin \mathcal{O}(e)$, then the matrix $Q_{e^+}^{(i)}$ is the identity matrix:

$$Q_{e^+}^{(i)} = I_{n^{(i)}}.$$

The (negative) matrix $Q_{e^-}^{(i)}$ is a diagonal matrix whose function is that of normalization. Indeed, for all $x^{(i)} \in S^{(i)}$, we must have

$$Q_{e^-}^{(i)}(x^{(i)}, x^{(i)}) + \sum_{y^{(i)} \in \text{succ}_e(x^{(i)})} Q_{e^+}^{(i)}(x^{(i)}, y^{(i)}) = 0.$$

It therefore follows that

$$Q_{e^-}^{(i)}(x^{(i)}, y^{(i)}) = \begin{cases} -\sum_{z^{(i)} \in S^{(i)}} Q_{e^+}^{(i)}(x^{(i)}, z^{(i)}) & \text{if } y^{(i)} = x^{(i)} \\ 0 & \text{if } y^{(i)} \neq x^{(i)} \end{cases}$$

3 Phase-Type Distributions

The exponential distribution is widely used in performance modeling. The reason, of course, is the exceptional mathematical tractability that flows from the memoryless property of this distribution. But sometimes, mathematical tractability is not sufficient to overcome the need to model processes for which the exponential distribution is simply not adequate. This leads us to explore ways in which we can model more general distributions while maintaining some of the mathematical advantages of the exponential. This is precisely what phase-type distributions permit us to do. Additionally, phase type distributions prove to be very useful when it is necessary to form a distribution having some given expectation and variance.

Phase-type distributions (the word “stage” is also used) get their name from the fact that they can be represented as the passage through a succession of exponential phases or stages. More precisely, a phase-type distribution is defined as the time, from some initial distribution, to absorption in a Markov chain in which all states except one are transient. The states of the Markov chain constitute the phases of the phase-type distribution. Different distributions are obtained by varying the number of phases, by varying the transition structure of the Markov chain and/or by varying the initial distribution. Popular examples of phase-type distributions include the Erlang distribution, the hyperexponential distribution and the Coxian distribution. Indeed, any distribution having a rational Laplace transform can be modeled arbitrarily closely with a Coxian distribution. In representing phase-type distributions, the single absorbing state is frequently omitted; it is implicitly understood that a departure from any state towards the exterior is meant to indicate a transition to the absorbing state. Graphically, an r-phase Coxian distribution is usually represented as in Figure 3.

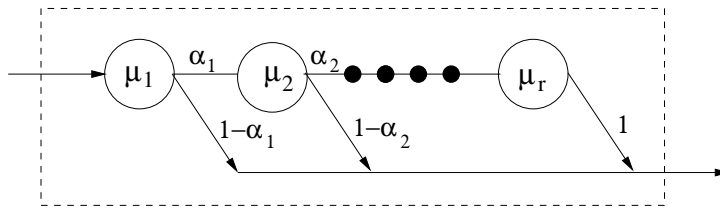


Figure 3: The Coxian Distribution

If this distribution represents a service process, then after providing service at phase i , $i = 1, 2, \dots, r - 1$, at rate μ_i , the process, with probability α_i , continues on to phase $i + 1$ to continue the service process, or with probability $1 - \alpha_i$, the service process terminates which simply means that the process enters the (implicitly represented) single absorbing state. In terms of the general description of a phase-type distribution, the initial distribution for this process is defined such that with probability 1, the process begins in phase 1. However, this is not exactly the representation we shall use. For reasons that will become clear later, when modeling phase-type distributions in a SAN, we shall number the phases of the distribution from 0 through $r - 1$. In one sense, phase 0 represents the exterior because a transition from any phase to phase 0 represents the termination of the process. Thus, when a phase-type distribution is in phase 0, it may mean that the process has terminated and is waiting to begin again. However, this is not the only meaning that may be associated with a phase-type distribution being in phase 0. Additionally, as with every other phase in the representation, the distribution of time spent in phase zero is exponential and is an integral part of the distribution. It is therefore necessary to understand which of the two possibilities applies, whether the process is waiting to begin or whether it is already underway. In our SAN model, a function is used to differentiate these possibilities on the basis of the context in which they are found.

In many modeling scenarios, it is necessary to interrupt certain processes at certain times and to allow these same processes to continue at a later time. Different preemption policies make the decision as to whether an interrupted process can continue from the point at which it was interrupted (called Preempt-Resume) while others force the process to begin all over again (called Preempt-Restart). The memoryless property of the exponential distribution facilitates the modeling of these policies, since at any point of time, the remaining lifetime in an exponential distribution is equal to the distribution itself and hence both policies are identical. Phase-type distributions on the other hand, do not possess this memoryless property and therefore it becomes necessary to take elapsed time into account when computing the remaining lifetime of a distribution. However, each exponential phase of a phase-type distribution does possess this property and so during an interruption of a phase-type service, it suffices to remember the particular phase in which the interruption occurred. If the scheduling policy permits the process to continue from the point of interruption, then the phase-type distribution is restarted in the phase at which it was interrupted. If the preemption policy forces a process to begin from the beginning again, then the phase-type distribution is begun from the initial distribution.

4 SANs and Phase-Type Distributions.

Having thus described both regular SAN and phase-type distributions, we are now in a position to introduce phase-type distributions into SANS, the result of which we shall refer to as a PH-SAN. It is important to remember that phase-type distributions apply only to events: they define the probability distributions of the random variables that represent the time between the moment certain events become enabled and their firing. This means that we must now distinguish between events with exponentially distributed firing rates and those with phase-type firing rates. Thus, in a PH-SAN, events have an additional attribute, a probability distribution characteristic denoted by *dist*, which is equal to M if the associated firing distribution is exponential, and is equal to PH if it is phase-type. It follows that the set of events in a PH-SAN, \mathcal{E} , may be partitioned into two sets, those having exponential firing distributions, called \mathcal{E}_M , and those with PH distributions, called \mathcal{E}_{PH} . We have $\mathcal{E} = \mathcal{E}_M \cup \mathcal{E}_{PH}$. Since events may also be local or synchronizing, this gives rise to subsets, \mathcal{E}_{M-l} : local events having an exponential distribution; \mathcal{E}_{M-s} : synchronizing events having an exponential distribution, and similarly for \mathcal{E}_{PH-l} and

\mathcal{E}_{PH-s} : local and synchronizing events having a PH distribution. Thus, $\mathcal{E} = \mathcal{E}_{M-l} \cup \mathcal{E}_{M-s} \cup \mathcal{E}_{PH-l} \cup \mathcal{E}_{PH-s}$.

A PH-SAN must contain a description of each phase-type event. This description must include the number of phases needed to represent the distribution, the service rate at each of these phases and the probabilities of transition among the phases. *In other words, an automaton complete with its own set of events.* These events are internal to the representation of the phase-type event, and their firing moves the representation from phase to phase. The description of a PH event must also specify the policy to be adopted in the case of an interruption: either preempt-resume or preempt-restart. This will be denoted by PP , for preemption policy, with $PP = B$ for begin again (i.e., preempt-restart), or $PP = C$ for continue on (i.e., preempt-resume). To present this more formally, let h be a PH event, i.e., $h \in \mathcal{E}_{PH}$. Then h is described by $\mathcal{D}^{(h)}$ where

$$\mathcal{D}^{(h)} = \left(\mathcal{A}^{(h)}, \mathcal{E}^{(h)}, PP^{(h)} \right).$$

In this description, $\mathcal{A}^{(h)}$ is an automaton with all the attributes of automata as defined formally in Section 2.2, i.e., $\mathcal{A}^{(h)} = (S^{(h)}, Edge^{(h)}, Label^{(h)})$; the set $\mathcal{E}^{(h)}$ is a set of events, again with all the attributes of events as defined in Section 2.2, and $PP^{(h)}$ specifies the preemption policy of h , as just outlined. The set of states of $\mathcal{A}^{(h)}$, i.e., $S^{(h)}$, contains one special state, state 0, which is taken to correspond to the first phase. Nothing happens when $\mathcal{A}^{(h)}$ is in this state and the PH event h is disabled. Once h is enabled, the exponential sojourn time to be spent in this phase can begin. At the end of the phase-type distribution, the automaton returns to this state. The automaton also returns to this state when the event h is interrupted and the preemption policy is B , preempt-restart.

The events that are internal to a PH event h , i.e., the elements of the set $\mathcal{E}^{(h)}$, are *local* events and have firing distributions that are exponentially distributed. Since they are local and all belong to the same automaton, the master of each is the automaton $\mathcal{A}^{(h)}$ itself. Furthermore, each $e^{(h)} \in \mathcal{E}^{(h)}$ has its own rate, given by $rate(e^{(h)})$. Therefore the triple that defines all $e^{(h)} \in \mathcal{E}^{(h)}$, the events internal to a PH event, may be written as $(loc, \mathcal{A}^{(h)}, rate(e^{(h)}))$. Let us now consider some of the implications of introducing PH events into a SAN.

- Since each PH event gives rise to a new automaton, it behooves us to differentiate between the automata of the base SAN, those automata whose purpose is to represent the components of the system being modeled, and automata whose purpose is to represent phase-type distributions. We shall call the former *base* automata, and the latter *induced* automata.
- In the formal description of a regular SAN, events were defined as having three components, namely type, master and rate: type being either local or synchronizing, master being the identity of a particular base automaton and rate being a constant or a function of the state of several base automata. A PH event must have these same three components. The type and master of a PH event are defined identically to those of exponential events. The rate of a phase-type event on the other hand, is given, not as a constant nor as a function, but instead is specified through the behavior of an automaton. We point out that it remains possible to associate a function with the rate of a PH event, by multiplying the rate of each phase in its representation by the value of the function. To differentiate between phase-type and exponential events, events in a PH-SAN are defined with four components: [(exponential or phase-type), (local or synchronizing), master, rate].

Formally, a PH-SAN may be defined as $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$, where

- N is the number of base automata in the model.
- $\mathcal{A} = (\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(N)})$ is the set of base automata. The product state space generated by this set of base automata is denoted by S .
- \mathcal{E} is the set of events defined on the base automata. It includes both exponentially distributed and phase-type events.
- \hat{f} is a reachability function defined on the product state space S .
- K is the number of PH events in the model. They are all defined on the set of base automata and are denoted by h_1, h_2, \dots, h_K .
- $\mathcal{D} = (\mathcal{D}^{(h_1)}, \mathcal{D}^{(h_2)}, \dots, \mathcal{D}^{(h_K)})$ is the set of phase-type event definitions, one for each PH event.

Observe that when $K = 0$, this definition reverts to that of a normal SAN as defined in Section 2.2. In a manner similar to that used to define the product state space S and a global state x of a regular SAN, we define the product state space of a PH-SAN, denoted by \tilde{S} , as

$$\begin{aligned}\tilde{S} &= S^{(1)} \times S^{(2)} \times \dots \times S^{(N)} \times S^{(h_1)} \times S^{(h_2)} \times \dots \times S^{(h_K)} \\ &= S \times S^{(h_1)} \times S^{(h_2)} \times \dots \times S^{(h_K)}\end{aligned}$$

where $S^{(h_i)}$ is the state space of the automata induced by the i^{th} PH event, $i = 1, 2, \dots, K$; and a global state of a PH-SAN, as

$$\begin{aligned}\tilde{x} &= (x^{(1)}, x^{(2)}, \dots, x^{(N)}, x^{(h_1)}, x^{(h_2)}, \dots, x^{(h_K)}) \\ &= (x, x^{(h_1)}, x^{(h_2)}, \dots, x^{(h_K)})\end{aligned}$$

where $x^{(h_i)}$ is the local state of the i^{th} induced automaton, the automaton that represents the i^{th} PH event. Consider the example of the PH-SAN shown in Figure 4.

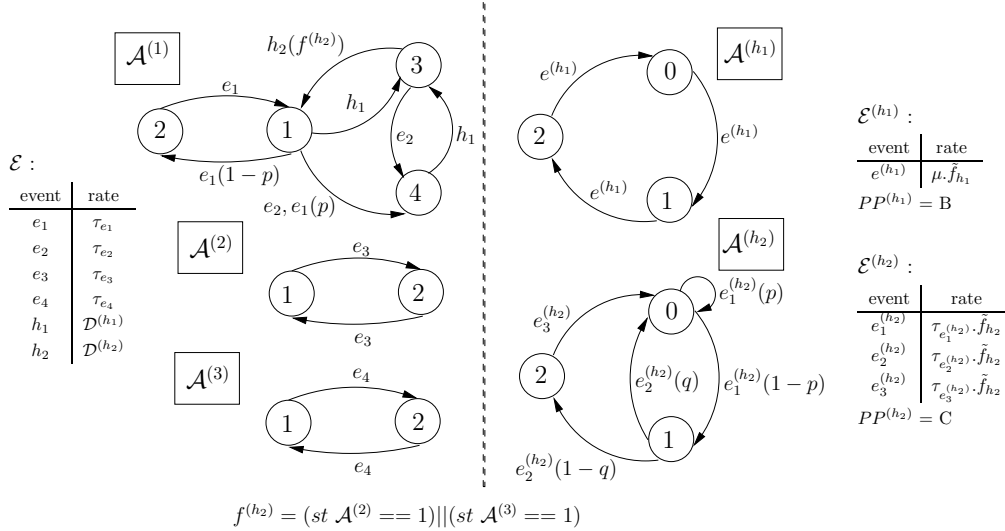


Figure 4: Example of a PH-SAN model

It consists of 3 base automata, $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$, the first having four local states and the second and third having 2 local states each. A total of 6 events, all of which are local events, are defined on these three base automata: e_1 , e_2 , e_3 and e_4 are exponential events while h_1 and h_2 are PH events. The firing rates of these events are specified within the table denoted \mathcal{E} . The rates associated with the two PH events give rise to two induced automata, $\mathcal{A}^{(h_1)}$ and $\mathcal{A}^{(h_2)}$. The first of these represents an Erlang-3 distribution in which the rate at each of the three phases is equal to μ . If the event h_1 is interrupted, the preemption policy is $PP^{(h_1)} = B$, i.e., preempt-restart. The second, $\mathcal{A}^{(h_2)}$, represents a 3-phase Coxian distribution with rate λ_i at phase i , $i = 0, 1, 2$ and probabilities of exiting at the end of phases 0 and 1 given by p and q respectively. The preemption policy of the PH event h_2 is $PP^{(h_2)} = C$, i.e., preempt-resume. Additionally, the induced events associated with each PH event have a functional rate (functions \tilde{f}_{h_1} and \tilde{f}_{h_2}). These induced events are eligible to fire only if the corresponding PH event is fireable. Finally, this model contains one functional transition: event h_2 can fire only if one or both of $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ are in local state 1.

To proceed, it is important to understand the manner in which events fire and the effect that these events have on one another. At any instant of time, events may be divided into those that are able to fire (enabled events) and those that are not (disabled events). In any given state x , one of the enabled events will eventually fire and this can have a number of consequences on the other events. After an event e fires:

- some events that were previous disabled may now be enabled.
- other events that were previously disabled will continue to be disabled.
- some events that were previously enabled will continue to be enabled.
- other events that were previously enabled may now be disabled.

In the example just presented, suppose that the base automaton $\mathcal{A}^{(1)}$ is in state 2. At this point we need not be concerned about the other two base automata since the events defined on them, e_3 and e_4 , affect $\mathcal{A}^{(1)}$ only through the function $f^{(h_2)}$ which is applicable only when $\mathcal{A}^{(1)}$ is in local state 3. Therefore, given that $\mathcal{A}^{(1)}$ is in state 2, only one event in $\mathcal{A}^{(1)}$ can fire, namely e_1 and so, after an exponentially distributed amount of time spent in state 2, event e_1 fires and automaton $\mathcal{A}^{(1)}$ moves to state 1. The moment that this happens, three events in $\mathcal{A}^{(1)}$ become enabled, i.e., eligible to fire, namely e_1 , e_2 and h_1 . This sets up a race condition among these three events and the one which fires first will move $\mathcal{A}^{(1)}$ into a state in which the other two may no longer be enabled. If e_1 fires first, $\mathcal{A}^{(1)}$ has the possibility of moving either back to state 2 or on to state 4 with probabilities p and $1 - p$ respectively. If $\mathcal{A}^{(1)}$ moves back to state 2 then e_2 and h_1 are no longer enabled. On the other hand, if e_2 fires first, then $\mathcal{A}^{(1)}$ moves to state 4. In this case e_1 is disabled but h_1 continues to be eligible to fire. Since the interruption policy of h_1 is preempt-restart, its firing process must begin again from its first phase, no matter which phase it was in when event e_2 fired. Now however, since h_1 is the only event that is enabled when $\mathcal{A}^{(1)}$ is in state 4, it will not be interrupted but instead will proceed to complete all 3 phases in its Erlang representation. Finally, if h_1 is the first to fire, $\mathcal{A}^{(1)}$ moves to state 3, a state in which e_1 is prohibited from firing, e_2 is enabled and h_2 is enabled if one or both of the automata $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ is in state 1. If this condition is true, then event h_2 continues from the point at which it was previously interrupted, since its preemption policy is preempt-resume.

Due to the memoryless property of the exponential distribution, the only situation requiring special attention occurs when the event or events that are disabled are PH events. Hence we shall use the term “disabling event” to designate an event e whose firing disables phase-type events that were enabled prior to the firing of e . The term “disabling event” will *not* be applied

to events which disable only exponential events. An enabled phase-type event which is disabled by the firing of an event e is said to be a *censored* event. The set of all phase-type events disabled by the firing of e in state x is called the censored set of the event e in state x . This censored set will be denoted $\perp_x(e)$, where the current state of the system is specifically identified.

In certain contexts, events such as e_1 and e_2 in $\mathcal{A}^{(1)}$ are said to be in *symmetric structural conflict*: structural since the disabling condition (or conflict) arises from the diagrammatic (or structural) layout of the edges on which the events are defined, and symmetric since if e_1 fires first then e_2 becomes disabled, and vice versa, if e_2 fires first, then e_1 becomes disabled. Observe that in SANS, it is not necessary for disabling conditions to be symmetric. It is possible to have a state in which two events a and b are both enabled and from which the firing of a leads to a state in which b is disabled, but the firing of b leads to a state in which a is still enabled. In this situation we could say that a is in conflict with b , but b is not in conflict with a . However, the word conflict itself has an inherent concept of symmetry and for this reason, we avoid using it, choosing instead to use the terminology of disabling events and censored sets. When an event e fires in some state x and phase-type events are subsequently disabled as a result of this firing and the structural layout of the edges, we shall say that e is a *structurally-disabling* event. The set of events disabled by the firing of e is called the set of structurally-censored events of e in the state x and is denoted by $\perp_{(s)x}(e)$.

Definition 4.1 Let $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$ be a PH-SAN. Let $e \in \mathcal{E}$ be an event, $h \in \mathcal{E}$ a PH event and $x \in S$ a global state of the base automata. Then e is said to be a *structurally-disabling* event in x and $h \in \perp_{(s)x}(e)$ a *structurally-censored* event in x if and only if there exists an $i \in \{1, 2, \dots, N\}$ with $a^{(i)}, b^{(i)} \in S^{(i)}$ such that $(x^{(i)}, a^{(i)}) \in T^{(i)}(e)$, $(x^{(i)}, b^{(i)}) \in T^{(i)}(h)$ and for all $c^{(i)} \in S^{(i)}$, $(a^{(i)}, c^{(i)}) \notin T^{(i)}(h)$.

In words, this means that it is possible for the PH-SAN to arrive at a state x such that within automata $\mathcal{A}^{(i)}$, there are events e and h which are both enabled in the same local state $x^{(i)}$ and that if e fires first and moves $\mathcal{A}^{(i)}$ to state $a^{(i)}$, there is no edge $(a^{(i)}, c^{(i)})$ labeled with the event h , i.e., the event h is disabled after the firing of e . Returning to the example, we see that

$$\begin{aligned} h_1 &\in \perp_{(s)x}(e_1) && \text{for } x = (1, x^{(2)}, x^{(3)}) \\ h_2 &\in \perp_{(s)x}(e_2) && \text{for } x = (3, x^{(2)}, x^{(3)}) \\ h_1 &\notin \perp_{(s)x}(e_2) && \text{for all } x. \end{aligned}$$

The event e_1 also (structurally) disables event e_2 but we do not write $e_2 \in \perp_{(s)x}(e_1)$ since e_2 is not a PH event.

With SANS, structurally is not the only way in which the firing of certain events may disable other events: events may also be *functionally-disabling*. Let us return to the example and consider the situation in which $\mathcal{A}^{(1)}$ is in local state 3, $\mathcal{A}^{(2)}$ is in local state 1 and $\mathcal{A}^{(3)}$ is in local state 2. In this case events e_3 and h_2 are enabled. The event e_2 is also enabled but this is not relevant to the situation we are studying. If the event e_3 fires before the event h_2 , then the function $f^{(h_2)}$ assumes the value 0 and the event h_2 is interrupted. It is no longer enabled. The event e_3 is said to *functionally-disable* the event h_2 : h_2 is a functionally-censored event. The firing of the event e_3 has the effect of changing the value of the function $f^{(h_2)}$ and consequently to imposing a restriction on h_2 . More formally, we have:

Definition 4.2 Let $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$ be a PH-SAN. Let $e \in \mathcal{E}$ be an event, $h \in \mathcal{E}$ a PH event and $x \in S$ a global state of the base automata for which $f^{(h)}(x) \neq 0$. Then e is said to be

a functionally-disabling event in x and $h \in \perp_{(f)x}(e)$ a functionally-censored event in x if and only if there exists a $y \in \text{succ}_e(x)$ such that $f^{(h)}(y) = 0$.

In words, this means that it is possible for the PH-SAN to enter a state x in which events e and h are defined and for which there is no functional impediment to the firing of the phase-type event h (i.e., $f^{(h)}(x) \neq 0$). If the firing of event e leads to a global state y of the base automata in which the event h is disabled due to the fact that the function in this state y now has a value 0, then we say that the event e functionally-disables the event h . The event e is a functionally disabling event and h , a functionally-censored event. The set of events that are functionally-censored by the firing of e in state x is denoted by $\perp_{(f)x}(e)$.

We now wish to define the sets containing all the phase-type events in a PH-SAN, $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$, that are structurally or functionally disabled by an event $e \in \mathcal{E}$ in some global state x of the base automata. We have the following:

- $\perp_x(e) = \perp_{(s)x}(e) \cup \perp_{(f)x}(e)$
- $\perp_{(s)}(e) = \cup_{x \in S} \perp_{(s)x}(e)$; $\perp_{(f)}(e) = \cup_{x \in S} \perp_{(f)x}(e)$
- $\perp(e) = \cup_{x \in S} \perp_x(e) = \perp_{(s)}(e) \cup \perp_{(f)}(e)$

Given that the set $\perp_{(s)x}(e)$ contains all PH events that are *structurally-censored* by the event e in x and that the set $\perp_{(f)x}(e)$ contains all PH events that are *functionally-censored* by event e in x , it follows that the set of all PH events that are censored (either structurally or functionally) by e in x is the union of these and is denoted by $\perp_x(e)$. The set $\perp_{(s)}(e)$, (respectively $\perp_{(f)}(e)$) is the set of all PH events which are structurally- (respectively functionally-) censored by e in *any* global state defined on the base automata, while the set $\perp(e)$ contains all PH events that are disabled by e in any global state of the base automata. In the previous example, the non-empty censored sets are

$$\perp_{(s)}(e_1) = \{h_1\}, \quad \perp_{(s)}(e_2) = \{h_2\}, \quad \perp_{(f)}(e_3) = \{h_2\}, \quad \perp_{(f)}(e_4) = \{h_2\}.$$

Additionally, within each of these sets, we distinguish their elements, the phase-type events, based on whether their preemption policy is preempt-restart, $PP = B$, or preempt resume, $PP = C$. For example the set $\perp_{(s)x}(e)|B$ (respectively $\perp_{(s)x}(e)|C$) is the set of all PH events which are structurally-censored by the firing of e in x and whose preemption policy is preempt-restart (respectively preempt-resume).

We now turn our attention to the definition of the Markov chain described by a PH-SAN. We shall denote the transition of the Markov chain from a global state $\tilde{x} = (x, x^{(h_1)}, x^{(h_2)}, \dots, x^{(h_K)})$ of the PH-SAN to another global state $\tilde{y} = (y, y^{(h_1)}, y^{(h_2)}, \dots, y^{(h_K)})$ by (\tilde{x}, \tilde{y}) . When it becomes necessary to identify the individual automata that change when this transition occurs, we shall use the notation

$$(\tilde{x}, \tilde{y}) = \left(x^{(i_1)}/y^{(i_1)}, \dots, x^{(i_n)}/y^{(i_n)}, x^{(h_1)}/y^{(h_1)}, \dots, x^{(h_k)}/y^{(h_k)} \right)$$

to signify that changes have occurred in n of the N base automata, namely i_1, i_2, \dots, i_n and in k of the K induced automata, namely those corresponding to PH events h_1, h_2, \dots, h_k . We consider first, transitions that occur due to the firing of an *exponential* event, e . We identify these by the letter M for Markovian. Later we shall address the case of transitions that result from the firing of a PH event h . When the exponential event e is a local, as opposed to synchronizing, event, the only possibilities are the following:

$M_l.(a)$ Event e is a local event for which the set $\perp_x(e)$ is empty.

In other words, no PH event is disabled when e fires. Then the only change in moving from state \tilde{x} to \tilde{y} occurs in the automaton on which the event e is defined. We write this change as $(x^{(i)}/y^{(i)})$ where $\{i\} = \mathcal{O}(e)$.

$M_l.(b)$ Event e is a local event for which the set $\perp_x(e)|B$ is empty but $\perp_x(e)|C$ is not.

In this case there are PH events which are disabled when e fires but these PH events all have the preempt-resume policy. This gives rise to exactly the same change of state as before, namely $(x^{(i)}/y^{(i)})$ where $\{i\} = \mathcal{O}(e)$, since the firing of e will leave the status of the PH events unaltered.

$M_l.(c)$ Event e is a local event for which the set $\perp_x(e)|B$ is not empty, and there exist preempt-restart PH events h_1, h_2, \dots, h_k ($\{h_1, h_2, \dots, h_k\} \subseteq \perp_x(e)|B$) enabled in x but disabled in the destination state y . When e fires, it will force the induced automata corresponding to these PH events to move to state 0, i.e., to restart. This transition is written as $(x^{(i)}/y^{(i)}, x^{(h_1)}/0, \dots, x^{(h_k)}/0)$ where $\{i\} = \mathcal{O}(e)$.

In all three cases, the rate at which the transition occurs from state \tilde{x} to \tilde{y} is the same and is given by

$$Q_l^e(\tilde{x}, \tilde{y}) = \text{rate}(e) \times P_e(x^{(i)}, y^{(i)}).$$

Obviously, if a local event e is not responsible of the occurrence of the transition (\tilde{x}, \tilde{y}) then $Q_l^e(\tilde{x}, \tilde{y}) = 0$.

The same three situations arise when e is a synchronizing event with an exponential firing time, but now additional automata, those concerned by the synchronization, must become involved. We follow the same procedure as before:

$M_s.(a)$ When the set $\perp_x(e)$ is empty, no PH events are disabled by the firing of e so the only automata affected are the base automata involved in the synchronization. This gives the transition $(x^{(i_1)}/y^{(i_1)}, x^{(i_2)}/y^{(i_2)}, \dots, x^{(i_n)}/y^{(i_n)})$ where $n \geq 1$ and $\{i_1, i_2, \dots, i_n\} \subseteq \mathcal{O}(e)$.

$M_s.(b)$ When $\perp_x(e)|B$ is empty but $\perp_x(e)|C$ is not, we obtain exactly the same situation as $M_s.(a)$ for the same reasons as cited for local exponential events above, i.e., case $M_l.(b)$. We have $(x^{(i_1)}/y^{(i_1)}, x^{(i_2)}/y^{(i_2)}, \dots, x^{(i_n)}/y^{(i_n)})$ where $n \geq 1$ and $\{i_1, i_2, \dots, i_n\} \subseteq \mathcal{O}(e)$.

$M_s.(c)$ The final case to consider occurs when $\perp_x(e)|B$ is not empty, and there exist preempt-restart PH events h_1, h_2, \dots, h_k ($\{h_1, h_2, \dots, h_k\} \subseteq \perp_x(e)|B$) enabled in x but disabled in the destination state y .

Following a similar reasoning to that used in the local event case, $M_l.(c)$, we obtain $(x^{(i_1)}/y^{(i_1)}, \dots, x^{(i_n)}/y^{(i_n)}, x^{(h_1)}/0, \dots, x^{(h_k)}/0)$ where $\{i_1, i_2, \dots, i_n\} \subseteq \mathcal{O}(e)$ and $n \geq 1$.

Again, in all three cases, the rate at which the transition occurs from state \tilde{x} to \tilde{y} is the same and is given by

$$Q_s^e(\tilde{x}, \tilde{y}) = \text{rate}(e) \times \prod_{i \in \mathcal{O}(e)} P_e(x^{(i)}, y^{(i)}).$$

Once again, it is apparent that, if a synchronising event e is not responsible of the occurrence of the transition (\tilde{x}, \tilde{y}) then $Q_s^e(\tilde{x}, \tilde{y}) = 0$.

Consider now the case in which the event which fires is a PH event. Like an exponential event, a PH event can be a local event or a synchronizing event and when it fires, it can give rise to exactly the same types of situation described above for an exponential event. As we have

seen, a PH event h defined on the set of base automata, is represented by an induced automaton, $\mathcal{A}^{(h)}$, which contains its own internal events, denoted $e^{(h)}$, and one special state, state 0. Only one type of internal event can possibly affect the base automata, those internal events which take $\mathcal{A}^{(h)}$ into the special state 0. We shall refer to this kind of event as an “End of Service” (EoS) event for it is precisely this event that signals the completion of the firing time distribution and invokes changes in the states of the base automata: in other words, an EoS event triggers the firing of the PH event h . It is an “End of Service” event which generates situations similar to those just defined for an exponential event. Other events defined on $\mathcal{A}^{(h)}$ move it from phase to phase, but have no effect whatsoever on the base automata. To distinguish such events from an EoS event, we shall refer to them as a “CoS” events, for “Continuation of Service”. CoS events occasion transitions within the induced automaton only. The list of all possible transitions that can emanate from PH events may now be given. As before, we seek to identify the global states \tilde{y} that can be reached from a global source $\tilde{x} = (x^{(1)}, x^{(2)}, \dots, x^{(N)}, x^{(h_1)}, x^{(h_2)}, \dots, x^{(h_K)}) = (x, x^{(h_1)}, x^{(h_2)}, \dots, x^{(h_K)})$.

For all CoS PH events $e^{(h)} \in \mathcal{E}^{(h)}$, whether h is local or synchronizing, and independent of whether the sets of PH events disabled by the firing of h , namely $\perp_x(h)|B$ and $\perp_x(h)|C$, are empty or not, the only change in \tilde{x} is given by $(x^{(h)}/y^{(h)})$ for $y^{(h)} \neq 0$. This is the only possibility for CoS PH events. The rate at which this transition occurs is given by

$$Q_{CoS}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = rate(e^{(h)}) \times P_{e^{(h)}}(x^{(h)}, y^{(h)}).$$

For all CoS PH events $e^{(h)}$, which do not cause the transition (\tilde{x}, \tilde{y}) , the value of $Q_{CoS}^{h,e^{(h)}}(\tilde{x}, \tilde{y})$ is equal to zero ($Q_{CoS}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = 0$).

As concerns EoS PH events, the following possibilities arise and correspond to those defined previously for exponential events: the first three corresponding to the case when the event h is a local event and the second three when it is a synchronizing event. Bear in mind that it is the firing of the EoS event $e^{(h)}$ which triggers the event h , causing it to fire.

- $PH_l(a)$ When the set $\perp_x(h)$ is empty, no PH events are disabled when h fires and the only change in moving from state \tilde{x} to \tilde{y} occurs in the automaton on which the event h is defined and within the automaton induced by h . We write this change as $(x^{(i)}/y^{(i)}, x^{(h)}/0)$ where $\{i\} = \mathcal{O}(h)$.
- $PH_l(b)$ When only PH events having the preempt-resume preemption policy are disabled by the firing of a local PH event h , then $\perp_x(h)|B$ is empty but $\perp_x(h)|C$ is not. This gives rise to exactly the same change of state as before, namely $(x^{(i)}/y^{(i)}, x^{(h)}/0)$ where $\{i\} = \mathcal{O}(h)$, since the firing of h will leave the status of the censored PH events unaltered.
- $PH_l(c)$ The final possibility for a local PH event h is that the set $\perp_x(h)|B$ is not empty, and there exist preempt-restart PH events h_1, h_2, \dots, h_k ($\{h_1, h_2, \dots, h_k\} \subseteq \perp_x(h)|B$) enabled in x but disabled in the destination state y . It makes no difference whether $\perp_x(h)|C$ is empty or not. When h fires, it will force the induced automata corresponding to these phase-type events to move to state 0, i.e., to restart. This transition is written as $(x^{(i)}/y^{(i)}, x^{(h)}/0, x^{(h_1)}/0, \dots, x^{(h_k)}/0)$ where $\{i\} = \mathcal{O}(e)$.

In all three cases, the rate at which the transition occurs from state \tilde{x} to \tilde{y} is the same and is given by

$$Q_{EoS,l}^h(\tilde{x}, \tilde{y}) = rate(e^{(h)}) \times P_{e^{(h)}}(x^{(h)}, y^{(h)}) \times P_h(x^{(i)}, y^{(i)}).$$

For all EoS PH events $e^{(h)}$, which do not cause the transition (\tilde{x}, \tilde{y}) , the value of $Q_{EoS,l}^{h,e^{(h)}}(\tilde{x}, \tilde{y})$ is equal to zero ($Q_{CoS}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = 0$).

Lastly, when h is a synchronizing event with a phase-type firing distribution, the same three situations arise, and again additional automata, those concerned by the synchronization, must become involved. We follow the same procedure as before:

- PH_s.(a)* When the set $\perp_x(h)$ is empty, no PH events are disabled when h fires so the only automata that are affected are the base automata involved in the synchronization. We have the following transition $(x^{(i_1)}/y^{(i_1)}, x^{(i_2)}/y^{(i_2)}, \dots, x^{(i_n)}/y^{(i_n)}, x^{(h)}/0)$ where $n \geq 1$ and $\{i_1, i_2, \dots, i_n\} \subseteq \mathcal{O}(h)$.
- PH_s.(b)* When $\perp_x(h)|B$ is empty but $\perp_x(h)|C$ is not, we obtain exactly the same situation as if no PH are disabled by the firing of h , for reasons previously cited.
- PH_s.(c)* The final case to consider occurs when $\perp_x(h)|B$ is not empty, and there exist PH events h_1, h_2, \dots, h_k ($\{h_1, h_2, \dots, h_k\} \subseteq \perp_x(h)|B$) enabled in x but disabled in the destination state y . Following a similar analysis to that used in the local event case, we obtain $(x^{(i_1)}/y^{(i_1)}, \dots, x^{(i_n)}/y^{(i_n)}, x^{(h)}/0, x^{(h_1)}/0, \dots, x^{(h_k)}/0)$ where $\{i_1, i_2, \dots, i_n\} \subseteq \mathcal{O}(h)$ and $n \geq 1$.

Again, in all three cases, the rate at which the transition occurs from state \tilde{x} to \tilde{y} is the same and is given by

$$Q_{EoS,s}^h(\tilde{x}, \tilde{y}) = \text{rate}(e^{(h)}) \times P_{e^{(h)}}(x^{(h)}, y^{(h)}) \times \prod_{i \in \mathcal{O}(h)} P_h(x^{(i)}, y^{(i)})$$

Again, for EoS PH events $e^{(h)}$, which do not cause the transition (\tilde{x}, \tilde{y}) , the value of $Q_{EoS,l}^{h,e^{(h)}}(\tilde{x}, \tilde{y})$ is equal to zero ($Q_{EoS,s}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = 0$). Notice that in all these cases, if the “presence” of the PH event h does not cause the transition (\tilde{x}, \tilde{y}) , then

$$Q_l^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = Q_{EoS,l}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = Q_{EoS,s}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) = 0 \quad \forall e^{(h)}.$$

It follows then that the $Q(\tilde{x}, \tilde{y})$ element of the transition rate matrix of the Markov chain representing a PH-SAN is given by

$$\begin{aligned} Q(\tilde{x}, \tilde{y}) &= \sum_{e \in \mathcal{E}_{M-l}} Q_l^e(\tilde{x}, \tilde{y}) + \sum_{e \in \mathcal{E}_{M-s}} Q_s^e(\tilde{x}, \tilde{y}) \\ &+ \sum_{h \in \mathcal{E}_{PH}} \left(\sum_{e^{(h)} \in \mathcal{E}^{(h)}} Q_{CoS}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) \right) \\ &+ \sum_{h \in \mathcal{E}_{PH-l}} \left(\sum_{e^{(h)} \in \mathcal{E}^{(h)}} Q_{EoS,l}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) \right) \\ &+ \sum_{h \in \mathcal{E}_{PH-s}} \left(\sum_{e^{(h)} \in \mathcal{E}^{(h)}} Q_{EoS,s}^{h,e^{(h)}}(\tilde{x}, \tilde{y}) \right). \end{aligned}$$

It is worthwhile examining some global states of the matrix Q that arises from our earlier PH-SAN example and listing all the transitions that can occur from these states along with the

corresponding rates of transition. Consider first the state $\tilde{x} = (x, x^{(h_1)}, x^{(h_2)}) = [(1, 1, 1), 0, 0]$ in which the three base automata are each in local state 1, and the two induced automata are in state 0. Within this global state, 3 events are candidates for firing within $\mathcal{A}^{(1)}$, namely e_1 , e_2 and h_1 ; and one event is fireable in each of $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$, namely e_3 and e_4 respectively. We shall determine the state reached when each of these five events fires.

1. Event e_1 fires: $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(2, 1, 1), 0, 0] = \tilde{y}$ at rate $\tau_{e_1} \times (1 - p)$, or $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(4, 1, 1), 0, 0] = \tilde{y}$ at rate $\tau_{e_1} \times p$. Case $M_l.(c)$.
Since $\perp_x(e_1)|C = \emptyset$ and $\perp_x(e_1)|B = \{h_1\}$, only h_1 can be disabled by e_1 but since h_1 has the preempt-restart preemption policy and is already in local state 0, the only change is to move $\mathcal{A}^{(1)}$ from local state 1 to local state 2 at rate $\tau_{e_1} \times (1 - p)$, or to local state 4 at rate $\tau_{e_1} \times p$.
2. Event e_2 fires: $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(3, 1, 1), 0, 0] = \tilde{y}$ at rate τ_{e_2} . Case $M_l.(a)$.
No PH events are disabled by e_2 since h_1 continues to be enabled after e_2 fires.
3. Event h_1 initiation: $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(1, 1, 1), 1, 0] = \tilde{y}$ at rate μ . Case $PH_l.(a)$.
Event h_1 begins its firing procedure. Once h_1 becomes eligible to fire, the CoS event $e^{(h_1)}$ also becomes eligible to fire and when $e^{(h_1)}$ fires, the induced automaton $\mathcal{A}^{(h_1)}$ moves from local state 0 to local state 1. No PH events are disabled by h_1 .
4. Event e_3 fires: $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(1, 2, 1), 0, 0] = \tilde{y}$ at rate τ_{e_3} . Case $M_l.(a)$.
No PH events are disabled by e_3 in state \tilde{x} . In particular, h_2 is not functionally-disabled since $\mathcal{A}^{(3)}$ is in local state 1.
5. Event e_4 fires: $\tilde{x} = [(1, 1, 1), 0, 0] \rightarrow [(1, 1, 2), 0, 0] = \tilde{y}$ at rate τ_{e_4} . Case $M_l.(a)$.
No PH events are disabled by e_4 in state \tilde{x} since h_2 remains eligible to fire ($\mathcal{A}^{(2)}$ is in local state 1).

These are the only global states attainable from \tilde{x} and therefore, along with the diagonal element, they constitute a row of the transition rate matrix Q .

Consider now the transitions that are possible from the global state $\tilde{x} = [(1, 1, 1), 2, 0]$ in which the PH event h_1 is in its third service phase. The events that can fire in this state are e_1 , e_2 , $e^{(h_1)}$ (an EoS event whose firing triggers the firing of the PH event, h_1), e_3 and e_4 .

1. Event e_1 fires: $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(2, 1, 1), 0, 0] = \tilde{y}$ at rate $\tau_{e_1} \times (1 - p)$, or $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(4, 1, 1), 2, 0] = \tilde{y}$ at rate $\tau_{e_1} \times p$. Case $M_l.(c)$.
If, when e_1 fires in automata $\mathcal{A}^{(1)}$, the local destination state is state 2, then the PH event h_1 is disabled and since the preemption policy of h_1 is preempt-restart, the induced automaton $\mathcal{A}^{(h_1)}$ is forced to move from local state 2 to local state 0. The resulting global state is then $[(2, 1, 1), 0, 0]$ and the rate of transition is $\tau_{e_1} \times (1 - p)$. On the other hand, if the local destination state in automata $\mathcal{A}^{(1)}$ is state 4, h_1 remains fireable and the only change is to move $\mathcal{A}^{(1)}$ from local state 1 to local state 4 and the rate of transition this time is $\tau_{e_1} \times p$.
2. Event e_2 fires: $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(4, 1, 1), 2, 0] = \tilde{y}$ at rate τ_{e_2} . Case $M_l.(a)$.
The PH event h_1 continues to be eligible to fire after e_2 fires: h_1 is not disabled by the firing of e_2 .
3. Event $e^{(h_1)}$ fires: $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(3, 1, 1), 0, 0] = \tilde{y}$ at rate μ . Case $PH_l.(a)$.
The firing of $e^{(h_1)}$ when $\mathcal{A}^{(h_1)}$ is in local state 2 announces the end of the phase-type

distribution and causes the PH event h_1 to fire, taking $\mathcal{A}^{(1)}$ from local state 1 to local state 3. The induced automaton $\mathcal{A}^{(h_1)}$ returns to local state 0.

4. Event e_3 fires: $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(1, 2, 1), 2, 0] = \tilde{y}$ at rate τ_{e_3} . Case $M_l.(a)$.
No PH events are disabled by the firing of e_3 in state \tilde{x} . In particular, h_2 is not functionally-disabled by e_3 since $\mathcal{A}^{(3)}$ is in local state 1.
5. Event e_4 fires: $\tilde{x} = [(1, 1, 1), 2, 0] \rightarrow [(1, 1, 2), 2, 0] = \tilde{y}$ at rate τ_{e_4} . Case $M_l.(a)$.
No PH events are disabled by e_4 in state \tilde{x} . In particular, h_2 is still eligible to fire ($\mathcal{A}^{(2)}$ is in local state 1).

Finally, to show the effect of functionally-disabling events, consider transitions from the global state $\tilde{x} = [(3, 1, 2), 0, 1]$. When the PH-SAN is in this state, four events are eligible to fire, namely e_2, e_3, e_4 and $e^{(h_2)}$. The transitions are as follows

1. Event e_2 fires: $\tilde{x} = [(3, 1, 2), 0, 1] \rightarrow [(4, 1, 2), 0, 1] = \tilde{y}$ at rate τ_{e_2} . Case $M_l.(b)$.
As e_2 fires and $\mathcal{A}^{(1)}$ moves from local state 3 to local state 4, the PH event h_2 is no longer able to fire. It is structurally-disabled by e_2 . However, since the preemption policy of h_2 is preempt-resume, the state of the induced automaton $\mathcal{A}^{(h_2)}$ remains unchanged.
2. Event e_3 fires: $\tilde{x} = [(3, 1, 2), 0, 1] \rightarrow [(3, 2, 2), 0, 1] = \tilde{y}$ at rate τ_{e_3} . Case $M_l.(b)$.
In this case h_2 is no longer able to fire, but this time it is because it is functionally-disabled by e_3 : the value of the function is zero in global state \tilde{y} .
3. Event e_4 fires: $\tilde{x} = [(3, 1, 2), 0, 1] \rightarrow [(3, 1, 1), 0, 1] = \tilde{y}$ at rate τ_{e_4} . Case $M_l.(a)$.
The event h_2 remains enabled in global state \tilde{y} .
4. Event $e^{(h_2)}$ fires: Two transitions are possible:
 - $\tilde{x} = [(3, 1, 2), 0, 1] \rightarrow [(1, 1, 2), 0, 0] = \tilde{y}$ at rate $q\lambda_1$. Case $PH_l.(a)$.
 - $\tilde{x} = [(3, 1, 2), 0, 1] \rightarrow [(3, 1, 2), 0, 2] = \tilde{y}$ at rate $(1 - q)\lambda_1$. Case $PH_l.(a)$.

When $e^{(h_2)}$ fires, then with probability q , automaton $\mathcal{A}^{(h_2)}$ moves to local state 0, thereby signaling that $e^{(h_2)}$ is an EoS event which triggers the PH event h_2 and gives rise to the first transition. On the other hand, with probability $1 - q$, automaton $\mathcal{A}^{(h_2)}$ moves to local state 2, thereby signaling that $e^{(h_2)}$ is a CoS event which means that h_2 is not yet ready to fire: hence the second transition.

5 Transformation procedure for a PH-SAN Model

In the previous section we saw how to define a SAN which incorporates events whose firing time is distributed according to a phase-type probability law. To distinguish between a SAN having only exponentially distributed events and one with both exponential and PH events, we refer to the first as an ordinary (or regular, or standard) SAN and the second as a PH-SAN. However, the actual specification, or definition, of a PH-SAN only takes us so far. The real interest in developing stochastic automata networks lies, not so much in their specification, although this can be extremely useful in certain circumstances, but rather in the fact that the transition matrix of the underlying Markov chain of a SAN can be represented compactly, and hence manipulated, as a compact sum of tensor products. It remains for us to show that a PH-SAN has this same representational property. Of course, it is known that any matrix may be represented as a sum of tensor products: it simply suffices to choose each individual element

of the matrix, to represent each as a tensor product and to form the sum of all such products. Since we have already shown how to form the infinitesimal generator of the Markov chain that underlies a PH-SAN, its representation as a sum of tensor products is immediate. However, this is not what we need. We seek a *compact* tensor representation in which the number of terms in the sum is of the order of the number of automata, not of the order of the number of elements in the matrix. This is the objective of this section. We shall show that the transition matrix of the underlying Markov chain of a PH-SAN has a compact tensor representation. Our approach is to show that a PH-SAN can be transformed into a regular SAN that reproduces stochastically the behavior of the PH-SAN. This transformation is a two-stage process. In the first stage, we shall transform an arbitrary PH-SAN into a stochastically equivalent PH-SAN in which certain types of events have been eliminated. The second stage is to take this modified PH-SAN and to transform it into a stochastically-identical regular SAN. Since a regular SAN has the compact representation property that we seek, it follows that a PH-SAN also has this property.

Previously we defined a *disabling event*, e , as one whose firing disables PH events that were enabled prior to the firing of e . We also defined a *censored set* as the set of PH events disabled when such an event fires. It should be apparent that the censored set obtained when e fires in one global state x is not necessarily the same as that obtained when it fires in some other state y . Less apparent perhaps, is that different censored sets may be obtained when e fires on different occasions from the *same* global state x . This arises when transition probabilities are associated with different destination states. The firing of e may, with probability p , lead to a state in which a PH event h is disabled, or with probability $1 - p$, lead to a state in which h is still enabled. Thus, a PH event h may be disabled in one destination states but not in another and consequently the censored sets obtained will not be the same. In transforming a PH-SAN to a regular SAN, we need to be aware of instances in which the firing of a disabling event e is *guaranteed* to disable a PH event h . In this case, the PH event h is an element of all censored sets obtained by the firing of e from a state x . A disabling event which is guaranteed to disable a PH event is called a *surely-disabling* or *moniform* event. Otherwise it is called a *possibly-disabling* or *multiform* event.

Definition 5.1 *Let $\tilde{M} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$ be a PH-SAN. Let $e \in \mathcal{E}$ be an arbitrary event defined on \mathcal{A} , let $h \in \mathcal{E}$ be a PH event, and let $x \in S$ be a global state in which both e and h are enabled. The event e is said to be a surely-disabling or moniform event if and only if h is disabled in all successor states, $y \in \text{succ}_e(x)$. The event e is said to be a possibly-disabling or multiform event if there exist states $y_1, y_2 \in \text{succ}_e(x)$ where h is enabled in y_1 but disabled in y_2 .*

Consider the PH-SAN model previously presented in Figure 4. Events e_1 and h_1 are both enabled in state $x = (1, x^{(2)}, x^{(3)})$. When the event e_1 fires, then with probability p , the destination state is $(2, x^{(2)}, x^{(3)})$, a state in which h_1 is disabled and with probability $1 - p$ the destination state is $(4, x^{(2)}, x^{(3)})$, a state in which h_1 is enabled. Thus, e_1 is a possibly-disabling event. On the other hand, events e_2 and h_2 are both enabled in state $(3, x^{(2)}, x^{(3)})$. When e_2 fires, only one state can be reached, $(4, x^{(2)}, x^{(3)})$, a state in which h_2 is disabled. Thus e_2 is a surely-disabling event.

The procedure we adopt is to decompose multiform events into moniform events and events that are not disabling at all. This allows us to develop an intermediate PH-SAN model that contains only moniform and non-disabling events which we can then transform into a regular SAN.

5.1 The intermediate model: A monoform PH-SAN

As we have just seen, one difficulty in transforming a PH-SAN into a regular SAN arises from the fact that certain types of disabling events behave differently in different circumstances, in the sense that different sets of censored events can result from the firing of an event e in a given state x . Hence, the first step of the transformation procedure of a PH-SAN to a SAN is to remove this difficulty. This leads us to distinguish among the events in the set \mathcal{E} of a PH-SAN based on the censored sets obtained when the event fires, i.e., to distinguish between multiform (possibly-disabling) events and monoform (surely-disabling) events. At the moment we need only be concerned with multiform events whose censored sets contain PH events having the preempt-restart preemption policy, ($PP = B$). A PH-SAN model which contains only monoform (surely-disabling) and non-disabling events is called a monoform PH-SAN. A PH-SAN model containing multiform (possibly-disabling) events is called a multiform PH-SAN.

To transform a PH-SAN containing multiform events into a PH-SAN containing only surely-disabling and non-disabling events, all multiform events are replaced by a number of surely-disabling and non-disabling events. The number of events (monoform and non-disabling) that replace a multiform event is equal to the number of different censored sets generated by the multiform event. For example, the equivalent monoform model to that of Figure 4 is given in Figure 5. Notice that the single multiform event of the model, e_1 in state $x = (1, x^{(2)}, x^{(3)})$, has been replaced by two monoform events, e_1^1 and e_1^2 . The firing of e_1^1 moves the system from state x to state $y_1 = (2, x^{(2)}, x^{(3)})$ at rate $\tau_{e_1}(1-p)$, while the firing of e_1^2 moves the system from state x to state $y_2 = (4, x^{(2)}, x^{(3)})$ at rate $\tau_{e_1}(p)$.

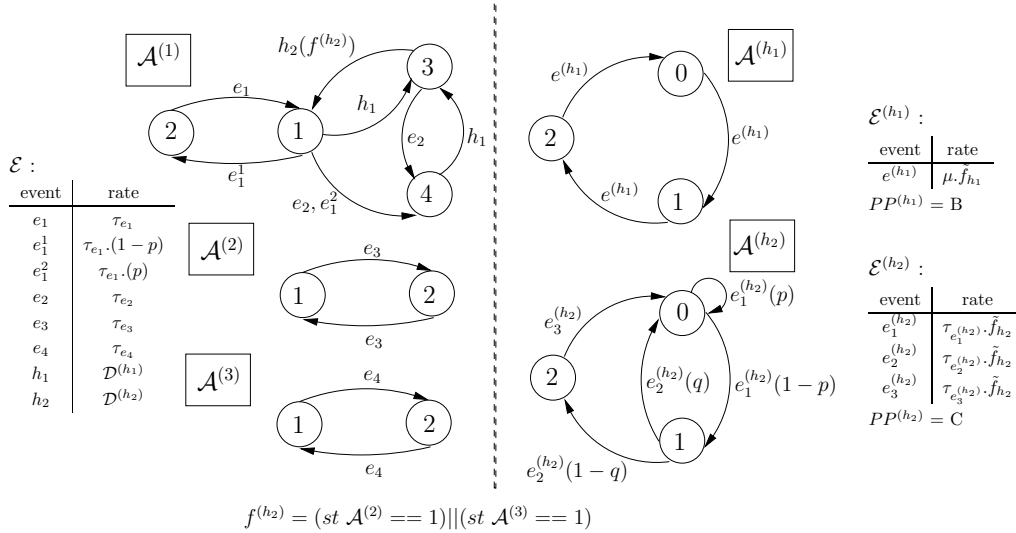


Figure 5: The equivalent monoform PH-SAN to that of Figure 4

If the multiform event e is exponential, then each newly generated event will be an exponential event; if e is a PH multiform event then each newly generated event will also be a PH event. To replace an exponential multiform event e , the only change that occurs to the PH-SAN is that the labels of the event are modified. In particular, the number of automata and the local states and arcs within automata, are unaffected. Even within the specification of the exponential multiform event, its type, master and rate are unchanged. Only the labels are affected for this is the means by which the behavior of the event is specified. The original labels concerning the

multiform event e are removed and replaced by new labels, one for each of the different censored sets that can arise by the firing of e . Each new label is imposed on the arc or arcs that represent the particular instance in which the firing of e gives rise to the associated censored set.

The situation is somewhat more complicated if e is a PH multiform event. A multiform event having r different censored sets gives rise to r new events, each of which is a PH event, and each must be associated with a descriptor to represent its firing distribution. However, all r events are associated with the same induced automaton, the one which represents the distribution of the original multiform PH event. This must be done to enforce the delay semantics of the distribution and moreover, it avoids useless state space explosion. This is illustrated in the following example.

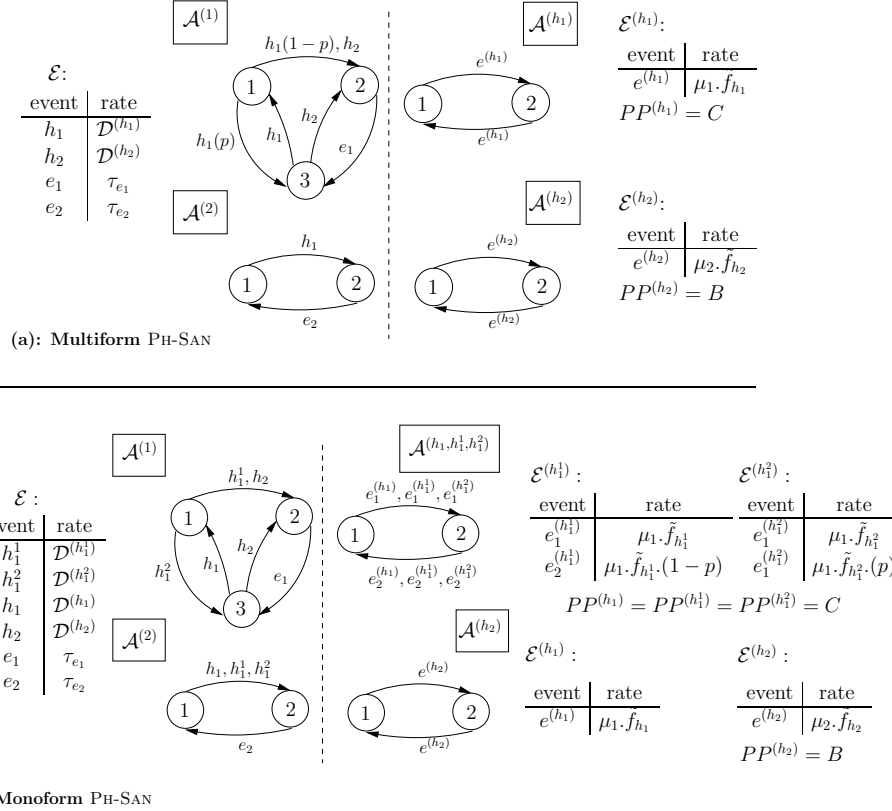


Figure 6: Model with a multiform PH event.

Figure 6 (a), represents a PH-SAN model having a multiform event, namely the PH event h_1 . The censored set obtained when h_1 fires in the state $x = (1, 1)$ is $\{h_2\}$, while the censored set obtained when h_1 fires in state $x = (3, 1)$ is the empty set. In Figure 6 (b), we present the corresponding monoform PH-SAN model. In the monoform model, the event h_1 is replaced by the two events h_1^1 and h_1^2 , one for each of the two different censored sets. Notice that in the situation where h_1 is not a disabling event, it is kept without replacement. The firing time distribution of each of all these events must be identical and equal to the firing time distribution of the original multiform event h_1 . Therefore each has the same description. We may write

$$\mathcal{D}^{(h_1)} = \mathcal{D}^{(h_1^1)} = \mathcal{D}^{(h_1^2)} = \mathcal{D}^{(h_1, h_1^1, h_1^2)} = \left(\mathcal{A}^{(h_1, h_1^1, h_1^2)}, \mathcal{E}^{(h_1^1)} \cup \mathcal{E}^{(h_1^2)} \cup \mathcal{E}^{(h_1)}, PP^{(h_1)} \right).$$

In particular, we do *not* introduce additional automata to represent additional PH events that are created during the transformation of a multiform PH event. The automaton that represents

the firing time distribution of a multiform PH event is unchanged in every way, except that the labels on its arcs are replaced with others that relate to the newly defined events. Thus this single automaton can function as the description of all the newly created events. For example, the arcs of automaton $\mathcal{A}^{(h_1, h_1^1, h_2^2)}$ are labeled with the events of $\mathcal{E}^{(h_1)}$, $\mathcal{E}^{(h_1^1)}$ and $\mathcal{E}^{(h_2^2)}$ respectively. For each of the events h_1 , h_1^1 and h_2^2 , we associate within the description $\mathcal{D}^{(h_1, h_1^1, h_2^2)}$ its own set of events, $\mathcal{E}^{(h_1)}$ for h_1 , $\mathcal{E}^{(h_1^1)}$ for h_1^1 and $\mathcal{E}^{(h_2^2)}$ for h_2^2 . The events $e_1^{(h_1^1)}, e_2^{(h_1^1)} \in \mathcal{E}^{(h_1^1)}$ and $e_1^{(h_2^2)}, e_2^{(h_2^2)} \in \mathcal{E}^{(h_2^2)}$ are the events that correspond to the event $e^{(h_1)} \in \mathcal{E}^{(h_1)}$ in the multiform PH-SAN. To ensure that h_1^1 and h_2^2 can fire only if h_1 is enabled in the base automata, two functions are used, namely $\tilde{f}_{h_1^1}$ and $\tilde{f}_{h_2^2}$ for h_1^1 and h_2^2 respectively. The probabilities of choice are multiplied with the rate of each EoS (End of Service) event of these two new PH events. Observe that when the two events h_1^1 and h_2^2 are enabled or disabled at the same time and if h_1^1 is enabled (respectively disabled) the event h_1 is not moniform. Thus, we can write that in the moniform model $\tilde{f}_{h_1^1} = \tilde{f}_{h_2^2} \neq \tilde{f}_{h_1}$. Automaton $\mathcal{A}^{(h_1, h_1^1, h_2^2)}$ behaves as follows: when h_1^1 and h_2^2 are enabled, only the events $e_1^{(h_1^1)}$ and $e_1^{(h_2^2)}$ (respectively $e_2^{(h_1^1)}$ and $e_2^{(h_2^2)}$) can fire in automaton $\mathcal{A}^{(h_1, h_1^1, h_2^2)}$. The event $e^{(h_1)}$ is not enabled when h_1^1 (and h_2^2) is enabled but only if h_1 is enabled.

The association of the same set of states and arcs with the events h_1 , h_1^1 and h_2^2 is a necessary part of the transformation process. Consider the situation that arises if h_1^1 (or h_2^2) is interrupted while in state 1 (2^{nd} service stage of its firing distribution) by the event h_2 . Since $PP^{(h_1^1)} = C$, this state must be preserved so that the distribution can continue from this point. Suppose now that h_1 is enabled when $\mathcal{A}^{(1)}$ is in state 3. Since h_1 , h_1^1 and h_2^2 essentially represent the same multiform event, the distribution must continue from the state in which it was interrupted, which is state 1, even though it is the event h_1 rather than h_1^1 that is now enabled.

Consider a global state of the moniform model $[(x^{(1)}, x^{(2)}), x^{(h_1, h_1^1, h_2^2)}, x^{(h_2)}]$ corresponding to state $[(x^{(1)}, x^{(2)}), x^{(h_1)}, x^{(h_2)}]$ of the multiform model and let us, by way of example, explore some possibilities. From state $[(1, 1), 0, 0]$ of the moniform PH-SAN model of Figure 6 (b), the events h_1^1 and h_2^2 are both enabled. Suppose that, after some time, the first phase in the activation of $e_1^{(h_1^1)}$ finishes. The automaton $\mathcal{A}^{(h_1, h_1^1, h_2^2)}$ therefore moves to state 1, and the global state is given by $[(1, 1), 1, 0]$. Suppose now that h_2 , which is enabled in state $x^{(1)} = 1$, actually fires and that its PH duration finishes before the second stage of h_1^1 finishes. The firing of h_2 interrupts h_1^1 whose preemption policy is preempt-resume, which implies that the current phase of h_1^1 must be saved. This takes us to state $[(2, 1), 1, 0]$. After the firing of event e_1 , the event h_1 is now enabled in state $[(3, 1), 1, 0]$. The work of the PH distribution of h_1 (which is the same as that of h_1^1 and h_2^2) is therefore restarted at the second phase (state 1 of automaton $\mathcal{A}^{(h_1)} = \mathcal{A}^{(h_1, h_1^1, h_2^2)}$), which leads, at the end of its service, to state $[(1, 2), 0, 0]$, the result of the firing of event $e^{(h_1)}$. It is easy to see that this scenario is the same as that of the multiform PH-SAN model.

The general procedure to be adopted for the transformation of any multiform PH event is similar to that just described. As was the case for a multiform exponential event, a multiform PH event h with r different censored sets of disabled PH events, gives rise to r PH events, all r having the same firing time distribution, the same preemption policy and all described on the same induced automaton, the automaton induced by the multiform PH event, h . Within this automaton, the set of states and the set of arcs are not altered in any way. The arcs on this automaton bear the labels of all r newly created events and it is by the use of functions that we distinguish among the r different possibilities. The firing indicator function associated with the multiform PH event is replaced with r firing indicator functions, one for each of the newly created events. A newly created event is enabled only if the value of its indicator function is 1

and only one of these r indicator functions can have the value 1 at any given time.

Theorem: Equivalence between Multiform and Monoform Models

Let $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$ be a multiform PH-SAN and let $\tilde{\mathcal{M}}' = (N, \mathcal{A}', \mathcal{E}', \hat{f}, K', \mathcal{D}')$ be the monoform PH-SAN derived from $\tilde{\mathcal{M}}$ by replacing multiform events having r censored sets with r monoform events as described above. Then $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$ define the same Markov chain.

Proof:

The only difference in the two models concerns the events which are defined on them. These events give rise to transitions that constitute the Markov chain. From the manner in which the multiform PH-SAN is transformed into a monoform PH-SAN, each transition of $\tilde{\mathcal{M}}$ generated by a multiform event in state x is also generated by a monoform or non-disabling event from the same state x in $\tilde{\mathcal{M}}'$ and vice versa. In other words,

- \Rightarrow For each transition that occurs from state \tilde{x} in $\tilde{\mathcal{M}}$, the same transition occurs from the same state in $\tilde{\mathcal{M}}'$.
- \Leftarrow Similarly, for each transition that occurs from state \tilde{x} in $\tilde{\mathcal{M}}'$, the same transition occurs in state \tilde{x} of $\tilde{\mathcal{M}}$.

A straightforward, but somewhat long, formal proof can be obtained by a simple rewriting of events.

5.2 Conversion of a monoform PH-SAN to a SAN

Let $\tilde{\mathcal{M}} = (N, \mathcal{A}, \mathcal{E}, \hat{f}, K, \mathcal{D})$ be a monoform PH-SAN with K PH events. Our goal in this section is to show how $\tilde{\mathcal{M}}$ may be transformed into an equivalent regular SAN, $\check{\mathcal{M}} = (\check{N}, \check{\mathcal{A}}, \check{\mathcal{E}}, \check{f})$. Let us consider each of these components separately, so that we may observe the effects of this transformation process.

- $\check{N} = N + K$ is the number of automata in $\check{\mathcal{M}}$. It is equal to the number of automata in $\tilde{\mathcal{M}}$ (N base automata and K induced automata).
- $\check{\mathcal{A}} = \{\check{\mathcal{A}}^{(1)}, \dots, \check{\mathcal{A}}^{(N)}, \check{\mathcal{A}}^{(h_1)}, \dots, \check{\mathcal{A}}^{(h_K)}\}$ is the set of automata that constitute $\check{\mathcal{M}}$.
 The first N of these correspond to the N base automata of $\tilde{\mathcal{M}}$ and they differ from those of $\tilde{\mathcal{M}}$ in one respect only: labels corresponding to monoform events in $\tilde{\mathcal{M}}$ are replaced by others in $\check{\mathcal{A}}$. The set of local states and the set of arcs are identical in both cases.
 The K automata corresponding to induced automata have the same set of states as their counterparts in $\tilde{\mathcal{M}}$, but the set of arcs in $\check{\mathcal{A}}$ contain those of \mathcal{A} in $\tilde{\mathcal{M}}$ and possibly additional arcs introduced to handle interruptions of PH events with the preempt-restart policy. Also, as is the case for the N base automata, the labels in the K induced automata may need to be altered.
- $\check{\mathcal{E}}$ is the set of events in $\check{\mathcal{M}}$ and is not the same as the set of events in $\tilde{\mathcal{M}}$. New events are introduced in $\check{\mathcal{M}}$ to replace PH events, to replace events that disable PH events and to replace events in the descriptions of PH events.
- \check{f} is the reachability function of $\check{\mathcal{M}}$.

It should be apparent that we need to consider the construction of: (a) the set of events, $\check{\mathcal{E}}$; (b) the set of labels in the base automata, $Label^{(i)}$; (c) the set of labels in the induced automata,

$\check{Label}^{(h)}$; and (d) the additional arcs that need to be added into induced automata, a set that we shall denote $\check{Int}^{(h)}$. The construction of these four sets may be accomplished by considering the different types of events that occur in the PH-SAN and the preemption policies of PH events.

Events defined on induced automata.

Let us consider changes to the automata that result from the transformation of events defined on an induced automaton, i.e., on an automaton that represents the firing time distribution of a PH event. Previously, in an induced automaton $\mathcal{A}^{(h)}$ that describes a phase-type distribution in a PH-SAN, we distinguished two types of events, namely CoS (Continuation of Service) events and EoS (End of Service) events. We first consider CoS events and the changes that arise from the transformation of CoS events in $\tilde{\mathcal{M}}$ to corresponding events in $\check{\mathcal{M}}$. The changes that occur in this case can affect only the labels of the induced automaton. On the other hand, the transformation of EoS events in $\tilde{\mathcal{M}}$ may affect the labels in those base automata associated with the event as well as the labels in the induced automaton itself.

(a) CoS (Continuation of Service) events:

The label of a CoS event $e^{(h)}$ on an arc from any node $x^{(h)}$ to any other node $y^{(h)}$ (other than the node zero) gives rise to a local event $\check{e}^{(h)}$ in the SAN model with rate equal to that of $e^{(h)}$ multiplied by the probability $P_{e^{(h)}}(x^{(h)}, y^{(h)})$. More precisely,

For all $h \in \mathcal{E}_{PH}$, $x^{(h)} \in S^{\check{(h)}} = S^{(h)}$ and $y^{(h)} \in S^{\check{(h)}} - \{0\}$:

For all PH-SAN CoS events $e^{(h)}$ ($loc, \mathcal{A}^{(h)}, rate(e^{(h)}) \in \mathcal{E}^{(h)}$) such that $(e^{(h)}, P_{e^{(h)}}(x^{(h)}, y^{(h)})) \in \check{Label}^{(h)}(x^{(h)}, y^{(h)})$, construct the SAN event:

$$\check{e}^{(h)} = (loc, \check{\mathcal{A}}^{(h)}, rate(e^{(h)}) \times P_{e^{(h)}}(x^{(h)}, y^{(h)})) \in \check{\mathcal{E}}$$

with the label $(\check{e}^{(h)}, 1) \in \check{Label}^{(h)}(x^{(h)}, y^{(h)})$.

Observe that the set of all automata affected by the event $\check{e}^{(h)}$ is simply the singleton set consisting of the automaton $\check{\mathcal{A}}^{(h)}$ itself, i.e., we have $\mathcal{O}(\check{e}^{(h)}) = \check{\mathcal{A}}^{(h)}$. It is important to understand why the firing rate of the event $e^{(h)}$ in the PH-SAN $\tilde{\mathcal{M}}$ is multiplied by the probability $P_{e^{(h)}}(x^{(h)}, y^{(h)})$ to form the firing rate of $\check{e}^{(h)}$ in the SAN $\check{\mathcal{M}}$. Consider, for example, the case of a PH-SAN where there exists a state $x^{(h)}$ such that $(e^{(h)}, p_1) \in \check{Label}^{(h)}(x^{(h)}, y^{(h)})$, $y^{(h)} \neq 0$, and $(e^{(h)}, 1 - p_1) \in \check{Label}^{(h)}(x^{(h)}, 0)$, meaning that the two edges exiting from the same state $x^{(h)}$ are both labelled by $e^{(h)}$. In this case $e^{(h)}$ is simultaneously a CoS event and an EoS event and in the SAN model, this event will be replaced by two different events, namely $\check{e}_1^{(h)}$ for the edge $(x^{(h)}, y^{(h)})$ and $\check{e}_2^{(h)}$ for the edge $(x^{(h)}, 0)$. Now, if we associate the label $(\check{e}_1^{(h)}, p_1)$ with the edge $(x^{(h)}, y^{(h)})$, the model no longer conforms to the conditions (in particular, condition 3) of a coherent SAN presented in Section 2.2. It is therefore necessary in the SAN model to multiply the probability by the firing rate. Multiplying the rate by the probability $P_{e^{(h)}}(x^{(h)}, y^{(h)})$ guarantees the correct probabilistic behavior since an event in the PH-SAN may be replaced by several events according to the number of labels containing that event. Finally, notice that the transitions of the PH-SAN which occur when the event $e^{(h)}$ fires, are equivalent to the transitions of the SAN model that occur upon the firing of the event $\check{e}^{(h)}$.

(b) EoS (End of Service) events:

Let us now move onto the transformation of EoS events in $\tilde{\mathcal{M}}$. The transformation is specified precisely as

For all $h \in \mathcal{E}_{PH}$, $x^{(h)} \in S^{\check{(h)}} = S^{(h)}$ and $y^{(h)} = 0$:

For all PH-SAN EoS events $e^{(h)}$ ($loc, \mathcal{A}^{(h)}, rate(e^{(h)}) \in \mathcal{E}^{(h)}$) such that $(e^{(h)}, P_{e^{(h)}}(x^{(h)}, y^{(h)})) \in \tilde{Label}^{(h)}(x^{(h)}, y^{(h)})$, construct the SAN event:

$$\check{e}^{(h)} = (syn, \check{\mathcal{A}}^{(h)}, rate(e^{(h)}) \times P_{e^{(h)}}(x^{(h)}, y^{(h)}) \in \check{\mathcal{E}}$$

with the label $(\check{e}^{(h)}, 1) \in \check{Label}^{(h)}(x^{(h)}, y^{(h)})$.

Furthermore, for all $i \in \mathcal{O}(h)$ and for all $(x^{(i)}, y^{(i)}) \in T^{(i)}(h)$ such that

$(h, P_h(x^{(i)}, y^{(i)})) \in \tilde{Label}^{(i)}(x^{(i)}, y^{(i)})$, construct the label:

$$(\check{e}^{(h)}, P_h(x^{(i)}, y^{(i)})) \in \check{Label}^{(i)}(x^{(i)}, y^{(i)}).$$

In this case, the set of automata affected by the event $\check{e}^{(h)}$ includes not only the induced automaton, $\check{\mathcal{A}}^{(h)}$, but also, all base automata affected by the event h , i.e., $\mathcal{O}(\check{e}^{(h)}) = \mathcal{O}(h) \cup \check{\mathcal{A}}^{(h)}$. Notice that the event $\check{e}^{(h)}$ in the SAN which corresponds to an EoS event $e^{(h)}$ in a PH-SAN model, is a synchronizing event, whereas the original event $e^{(h)}$ is a local event. This is due to the fact that an event representing the end of service of a phase-type distribution must be synchronized with the occurrence of the transition generated by the actual firing of h . Once again, it is easy to show that the transitions generated by the firing of $e^{(h)}$ in the PH-SAN are equivalent to those generated by the firing of $\check{e}^{(h)}$ in the SAN.

Events defined on the base automata.

Let us now consider events that are defined on the base automata. To examine this more closely, consider the effect of disabling a PH event h whose preemption policy is preempt-resume. A function \check{f}_h is associated with CoS events in the induced automaton of the PH-SAN and has the value 1 when the PH-SAN is in any state in which h may fire. Suppose that h is in its k^{th} service stage where the firing of an event $\check{e}^{(h)}$ will signal the end of this service phase. If an event that disables h should happen to fire at this moment, then the service is disrupted because the function \check{f}_h , which in the SAN model controls the firing of the event $\check{e}^{(h)}$, is now equal to 0. The work done to this point is therefore saved at the k^{th} service phase. The next time that h is enabled the value of the function \check{f}_h will be equal to 1, and the work continued from the k^{th} phase by restarting the exponential delay at that phase. For this reason, exponential events in the PH-SAN, $\tilde{\mathcal{M}}$, having censored sets that are either empty or containing only PH events with the preempt-resume policy, are replicated in the SAN, $\check{\mathcal{M}}$. Exponential events that disable PH events with the preempt-resume policy simply postpone the evolution of the phase-type distribution and do not reset the associated automata to zero. As we have just seen, this is modeled by a function and does not affect the exponential event.

In the case of the interruption of a PH event h having a preempt-restart policy, the interruption of the work of h may be expressed in terms of a synchronization between the firing of the disabling event and a return to phase 0 in the automaton that models the phase-type distribution. At this point, new edges may have to be added to this automaton, to represent the interruption of service and the return to phase 0.

Exponential Events which disable PH-R events

We now consider the case when the event which disables a PH event with a preempt-restart policy is an exponential event. For all events $e \in \mathcal{E}_M$ such that $\perp(e)|R \neq \emptyset$, we construct the event \check{e} such that:

$$\check{e} : (syn, master(e), rate(e)) \in \check{\mathcal{E}}, \text{ and}$$

for all $i \in \mathcal{O}(e)$, $\forall (x^{(i)}, y^{(i)}) \in T^{(i)}(e)$ such that $(e, P_e(x^{(i)}, y^{(i)})) \in \tilde{Label}^{(i)}(x^{(i)}, y^{(i)})$, construct the label $(\check{e}, P_e(x^{(i)}, y^{(i)})) \in \check{Label}^{(i)}(x^{(i)}, y^{(i)})$.

Additional Edges and Labels:

For all $h \in \perp(e)|R$ and for all $x^{(h)} \in \check{S}^{(h)} - \{0\}$, Construct:

- Edge: $(x^{(h)}, 0)$ if $(x^{(h)}, 0) \notin \check{Edge}^{(h)}$. Label: $(\check{e}, f_{(e,h)}) \in \check{Label}^{(h)}(x^{(h)}, 0)$.
- Edge: $(x^{(h)}, x^{(h)})$ if $(x^{(h)}, x^{(h)}) \notin \check{Edge}^{(h)}$. Label: $(\check{e}, 1 - f_{(e,h)}) \in \check{Label}^{(h)}(x^{(h)}, x^{(h)})$.
- Edge: $(0, 0)$ if $(0, 0) \notin \check{Edge}^{(h)}$. Label: $(\check{e}, 1) \in \check{Label}^{(h)}(0, 0)$.

We have $\mathcal{O}(\check{e}) = \mathcal{O}(e) \cup_{h \in \perp(e)|R} \check{\mathcal{A}}^{(h)}$.

Corresponding to each exponential event e of the PH-SAN, which disables a PH-R event h , there corresponds an event \check{e} which synchronizes the end of the exponential delay represented by e and the interruption of the progress of the phase-type distribution, represented by a return to phase 0. This interruption depends on the boolean function $f_{(e,h)}$ whose value is equal to 1 for states in which e can disable h . Let $S(e)$ be the set of states of the model in which the event e can fire. Among these states, we distinguish those in which the event e can disable the PH event h . We denote this set by $S(e, h)$. The value of the function $f_{(e,h)}$ is equal to 1 for these states. More precisely we write $f_{(e,h)}(x) = 1$ for all $x \in S(e, h)$, and $f_{(e,h)}(x) = 0$ for $x \in S(e) - S(e, h)$. The construction of the function $f_{(e,h)}$ consists in exploring all the states of the PH-SAN where e and h are enabled, and in which the firing of e can disable h (which allows us to construct the set $S(e, h)$). This search may be carried out in parallel with the construction of the censored sets $(\perp(e))$ of an event e . Now suppose that in the SAN model, the system is in a state x of $S(e, h)$ and that the automaton induced by h is in local state $x^{(h)}$. In this case, the firing of the event \check{e} implies the interruption of the event h because $f_{(e,h)}(x) = 1$ and the automaton induced by h must return to phase 0. Hence the need for an edge between phase $x^{(h)}$ and 0. On the other hand, if the system is in a state x where e cannot disable h whose induced automaton is in local state $x^{(h)}$, $f_{(e,h)}(x) = 0$, a self-loop on state $(x^{(h)}, x^{(h)})$ is needed to synchronize with the firing of \check{e} . In this case \check{e} does not interrupt h .

It is important at this stage of the translation to underline the interest in passing first by a monoform PH-SAN. If the event e is multiform in a state, then the interruption of the progress of a PH event must depend on the successor state after the firing of e , which is impossible to effect without decomposing the event. Furthermore, the use of a function to model this situation is impossible, because this would involve the evaluation of a future state. Since the event e is monoform, then, if it can disable an event h in some state x , it will interrupt h no matter what the successor state. Again, transitions generated by the firing of the exponential event e in the PH-SAN are equivalent to those generated by the firing of the event \check{e} in the SAN model. This observation may be verified by examining the set of effects caused by these events.

Phase-type Events which disable PH-R events

The final part of the construction concerns the case of a PH event v , of a PH-SAN, which may disable a PH-R event h . Indeed, in this case, it is the EoS events of the induced automaton for v that interrupt the progress of the phase-type distribution of h . The analysis is similar to that performed for the exponential disabling event just described. More precisely,

For all events $v \in \mathcal{E}_{PH}$ such that $\perp(v)|R \neq \emptyset$, construct the event \check{v} such that:

Additional Edges and Labels:

For all $h \in \perp(v)|R$, for all events $\check{e}^{(v)}$ that correspond to EoS events, and for all $x^{(h)} \in \check{S}^{(h)} - \{0\}$, Construct:

- Edge: $(x^{(h)}, 0)$ if $(x^{(h)}, 0) \notin \check{Edge}^{(h)}$. Label: $(\check{e}^{(v)}, f_{(v,h)}) \in \check{Label}^{(h)}(x^{(h)}, 0)$.
- Edge: $(x^{(h)}, x^{(h)})$ if $(x^{(h)}, x^{(h)}) \notin \check{Edge}^{(h)}$. Label: $(\check{e}^{(v)}, 1 - f_{(e,h)}) \in \check{Label}^{(h)}(x^{(h)}, x^{(h)})$.
- Edge: $(0, 0)$ if $(0, 0) \notin \check{Edge}^{(h)}$. Label: $(\check{e}^{(v)}, 1) \in \check{Label}^{(h)}(0, 0)$.

We have $\mathcal{O}(\check{e}^{(v)}) = \mathcal{O}(\check{e}^{(v)}) \cup_{h \in \perp(v)|R} \check{\mathcal{A}}^{(h)}$

Again, it is not difficult to see the equivalence between transitions in the PH-SAN and SAN models after this transformation. Figure 7 represents the SAN model that corresponds to the monoform PH-SAN model of Figure 5.

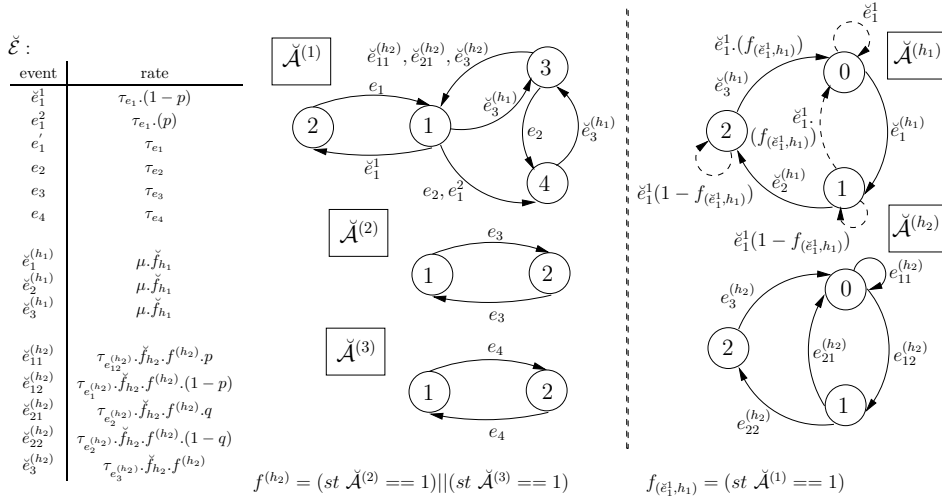


Figure 7: The SAN model that corresponds to the PH-SAN model of figure 5

Observe in this model that

- Exponential events that do not disable PH-B events are conserved (Events: e_1^2 , e_1 , e_2 , e_3 and e_4).
- Events defined on induced automata representing phase-type distributions are replaced with new events whose rates are equal to the rates of the corresponding events in the PH-SAN, multiplied by the associated probability. The EoS events of h_1 (respectively h_2) are replaced by synchronizing events $\check{e}_3^{(h_1)}$ (respectively $\check{e}_{11}^{(h_2)}$, $\check{e}_{21}^{(h_2)}$ and $\check{e}_3^{(h_2)}$) which synchronize the EoS edges of $\check{\mathcal{A}}^{(h_1)}$ (respectively $\check{\mathcal{A}}^{(h_2)}$) and the edges of $\check{\mathcal{A}}^{(1)}$ which correspond to the edges labelled by h_1 (respectively h_2) in $\mathcal{A}^{(1)}$.
- Finally, the event e_1^1 , being an event which disables h_1 in the PH-SAN, is replaced in the SAN model by the synchronizing event \check{e}_1^1 which also appears in the automaton $\check{\mathcal{A}}^{(h_1)}$ for the purpose of interrupting the phase-type distribution modeled by $\check{\mathcal{A}}^{(h_1)}$ (Observe the additional dashed edges that have been added to the automaton). Notice that the states

in which \check{e}_1^1 can interrupt the progress of $\check{\mathcal{A}}^{(h_1)}$ are in which automaton $\check{\mathcal{A}}^{(1)}$ is in its local state 1. This phenomenon is modeled by the function $f_{(\check{e}_1^1, h_1)}$ whose construction is evident for this model.

Let us examine some transitions of the Markov chain of the SAN model of Figure 7. Recall that in Section 4, we previously displayed all the transitions possible from states $\tilde{x} = [(1, 1, 1), 0, 0]$, $\tilde{x} = [(1, 1, 1), 2, 0]$ and $\tilde{x} = [(3, 1, 2), 0, 1]$ of the PH-SAN. In the context of the SAN model of Figure 7, we shall only consider transitions from state $\check{x} = (1, 1, 1, 0, 0)$. The event that are enabled from state \check{x} are \check{e}_1^1 , e_1^2 and e_2 in automaton $\check{\mathcal{A}}^{(1)}$, the event e_3 in $\check{\mathcal{A}}^{(2)}$, the event e_4 in $\check{\mathcal{A}}^{(3)}$ and the event $\check{e}_1^{(h_1)}$ in $\check{\mathcal{A}}^{(h_1)}$ since the function $f_{h_1}(\check{x})$ is equal to 1.

- The firing of event \check{e}_1^1 leads to state $(2, 1, 1, 0, 0)$ at rate $\tau_{e_1} \times (1 - p)$.
- The firing of event e_1^2 leads to state $(4, 1, 1, 0, 0)$ at rate $\tau_{e_1} \times (p)$.
- The firing of event e_2 leads to state $(4, 1, 1, 0, 0)$ at rate τ_{e_2} .
- The firing of event e_3 leads to state $(1, 2, 1, 0, 0)$ at rate τ_{e_3} .
- The firing of event e_4 leads to state $(1, 1, 2, 0, 0)$ at rate τ_{e_4} .
- The firing of event $\check{e}_1^{(h_1)}$ leads to state $(1, 1, 1, 1, 0)$ at rate μ .

Observe that the transitions exiting state $(1, 1, 1, 0, 0)$ of the SAN model and those exiting state $[(1, 1, 1), 0, 0]$ of the PH-SAN model are the same: same successor state and same firing rate.

6 Complexity

The transformation of a PH-SAN model into a SAN model is based on identifying situations in which certain events are disabled. Consider an arbitrary event $e \in \mathcal{E}$ (either exponential or PH) and a PH event $h \in \mathcal{E}$. As previously stated, if the preemption policy of h is preempt-resume, then no change need be made to the representation of the event e nor to the automaton induced by the event h . It follows that we really do not need to know if e disables a preempt-resume PH event or not. On the other hand, if the preemption policy of h is preempt-restart, the representation of the event e must be replaced by an event which synchronizes with the automaton induced by h . Thus, only events which disable PH-B events need be identified. We show below that *structurally-disabling* events can be detected rather easily, but that *functionally disabling* events are more costly to detect and the complexity of the algorithm in this case is related to the function associated with the PH-B event.

We begin with structurally-disabling events. Our goal is to identify the states in which a given event e structurally disables a PH-B event h . This must be done for every $e \in \mathcal{E}$ and $h \in \mathcal{E}_{PH-B}$. From among all states x in which both these events are enabled, we can identify the successor states obtained from the firing of e in which h is no longer enabled. Since our concern at this point is with structurally-disabling events, such a state has, in each automaton $\mathcal{A}^{(i)}$, $i \in \mathcal{O}(e) \cap \mathcal{O}(h)$,

- an edge $(x^{(i)}, y^{(i)}) \in Edge^{(i)}$ labeled with e , and an edge $(x^{(i)}, z^{(i)}) \in Edge^{(i)}$ labeled with h .
- after the firing of e , a successor state $y^{(i)}$ of $x^{(i)}$, ($y^{(i)} \in succ_e(x^{(i)})$), in which h is not enabled, which implies that in automaton $\mathcal{A}^{(i)}$ there is no edge out of $y^{(i)}$ labeled with h .

For each automaton $\mathcal{A}^{(i)}$, $i \in \mathcal{O}(e) \cap \mathcal{O}(h)$, let $S^{(i)}(e, h)$ be the set of local states $x^{(i)}$ that satisfy these two conditions. The construction of the sets $S^{(i)}(e, h)$ is relatively simple. For each $i \in \mathcal{O}(e) \cap \mathcal{O}(h)$, it suffices to check the set of local states $S^{(i)}$ of automaton $\mathcal{A}^{(i)}$ to identify states $x^{(i)}$ which satisfy the first condition, and for each $x^{(i)}$, it is sufficient to check the successor states of $x^{(i)}$ to see if they satisfy the second condition. The complexity of the first part is of order $n^{(i)}$, the number of local states of $\mathcal{A}^{(i)}$. The complexity of the second part is, at most, also of $n^{(i)}$ (if all the states of $\mathcal{A}^{(i)}$ are reachable from any other local state $x^{(i)}$). Thus, the complexity of the construction of all the sets $S^{(i)}(e, h)$ is $O\left(\sum_{i \in \mathcal{O}(e) \cap \mathcal{O}(h)} (n^{(i)})^2\right)$. Thus the complexity for all possible pairs (e, h) , is given by

$$O\left(\sum_{e \in \mathcal{E}} \sum_{h \in \mathcal{E}_{PH-B}} \sum_{i \in \mathcal{O}(e) \cap \mathcal{O}(h)} (n^{(i)})^2\right).$$

In the case of *functionally-disabling* events, the analysis is more complex. Consider a functional PH-B event h and let $\omega(f^{(h)}) = \{i_1, i_2, \dots, i_\omega\} \subseteq \{1, 2, \dots, N\}$ be the set of automata whose current state is an argument of the function $f^{(h)}$. Let $S^{(\omega(f^{(h)}))}$ be the subspace of states defined by the cartesian product of all local spaces $S^{(i)}$ such that $i \in \omega(f^{(h)})$. The size of the state space $S^{(\omega(f^{(h)}))}$ is then equal to $\prod_{i \in \omega(f^{(h)})} n^{(i)}$. For each global state $x = (x^{(1)}, x^{(2)}, \dots, x^{(N)}) \in S$, let $x^{(\omega(f^{(h)}))} = (x^{(i_1)}, x^{(i_2)}, \dots, x^{(i_\omega)}) \in S^{(\omega(f^{(h)}))}$ be the vector of states needed to evaluate the function $f^{(h)}$. It is evident that, for an event e to functionally-disable an event h , it is necessary that it be defined on at least one automaton of $\omega(f^{(h)})$, which means that $\mathcal{O}(e) \cap \omega(f^{(h)})$ cannot be empty. In order to specify this more formally, we need the following definition. If v is a vector defined on a space A and w is a vector defined on space B for which $B \subset A$, then w is the *image* of v on B if and only if, for all $i \in B$, $v^{(i)} = w^{(i)}$. Now, for each such event e , and for each state x whose image on $\omega(f^{(h)})$ is $x^{(\omega(f^{(h)}))}$ with $f^{(h)}(x) = 1$ (i.e., h is enabled), two conditions must be satisfied if e is to functionally disable h , namely:

- e must be enabled in state $x^{(\omega(f^{(h)}))}$
- after the firing of e , there must be a successor state of $x^{(\omega(f^{(h)}))}$, i.e., a $y^{(\omega(f^{(h)}))} \in \text{succ}_e(x^{(\omega(f^{(h)}))})$, such that $f^{(h)}(y^{(\omega(f^{(h)}))}) = 0$

If these two conditions hold, then e functionally disables h in all states x where e and h are enabled and for which the image of x on $\omega(f^{(h)})$ is $x^{(\omega(f^{(h)}))}$. However, for each event e , it is necessary to traverse the set of states $S^{(\omega(f^{(h)}))}$ to identify those states $x^{(\omega(f^{(h)}))}$ in which e is enabled (first condition). Then, for each of these $x^{(\omega(f^{(h)}))}$, another traversal is needed to test for the second condition. Therefore, for each pair (e, h) with non-empty $\mathcal{O}(e) \cap \omega(f^{(h)})$ and PH-B event h , the complexity of this identification operation is given by

$$O\left(\prod_{i \in \omega(f^{(h)})} n^{(i)}\right)^2.$$

While in the process of identifying disabling situations in a model, it is possible at the same time to mark those states in which a disabling event is a multiform event. In this manner, the process of replacing multiform events with other monoform events is relatively easy to perform and its complexity is of the order of the number of automata edges on which a multiform event is defined.

To summarize, once we know which events are disabling events and the states in which these disabling actions occur in the PH-SAN model, then to transform a PH-SAN into a SAN, the complexity of the four transformation steps previously presented is a function of the number of events, edges and states of each of the automata in the model, and is at most

$$O \left(|\mathcal{E}| \cdot \sum_{i=1}^N (n^{(i)})^2 + \sum_{h \in \mathcal{E}_{PH}} (k^{(h)})^2 \right),$$

where $k^{(h)}$ is the number of phases in the phase-type distribution associated with event h .

7 Conclusion

In this paper we have shown how phase-type probability distributions may be incorporated into stochastic automata networks, (SANS). The key step is in realizing that it is the events defined on the various automata that are responsible for generating the transitions of the Markov chain that underlying a SAN. The distribution of the time between the enabling of an event and its firing is either exponentially distributed or else it has a phase-type distribution. In this later case, it has been necessary to model the distribution as the passage of time through a sequence of exponential phases, which implies that each phase-type event must be modeled as an automaton, i.e., each phase-type event gives rise to an induced automaton.

A major emphasis of this paper has been to show that the Markov chain of a SAN with phase-type events (which we called a PH-SAN) has a compact tensor representation. This was accomplished by showing that it is possible to derive a regular, yet *stochastically-equivalent*, SAN from a PH-SAN. In a first step, a single event that can disable different multiple sets of phase-type events is replaced with multiple events that do not have this property. This is followed by a second step which completes the reduction of the original PH-SAN to a regular SAN.

The techniques which we propose in this paper lay the groundwork for the development of hierarchical SANS. In this paper we have shown how the base automata must interact with the induced automata: induced automata are essentially subservient to the events that happen at a higher level, namely those defined on the base automata. Similarly, events that occur at one level of a hierarchical SAN need conform to the effects of the firing of events at a higher level in the hierarchy. This opens the way for a formal definition of hierarchical SANS and its representation as compact sum of tensor products.

References

- [1] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In G. Rozenborg, editor, *Advances in Petri Nets 1987*, number 266 in LNCS, pages 132-145, Springer-Verlag, 1987.
- [2] A. Benoit. Méthodes et algorithmes pour l'évaluation des performances des systèmes informatiques à grand espace d'états. PhD thesis, INPG, Grenoble, France, Juin 2003.
- [3] A. Benoit, L. Brenner, P. Fernandes, B. Plateau and W.J. Stewart. The PEPS Software Tool. In *Performance TOOLS 2003*, pages 98-115, Urbana, Illinois, USA, 2003. Springer-Verlag.
- [4] P. Chen, S.C. Bruell and G. Balbo. Alternative methods for incorporating non-exponential distributions into stochastic timed Petri nets. In *Proc. of the third International Workshop on Petri Nets and Performance Models*, pages 187-197, Kyoto, Japan, December 11-13 1989. IEEE Computer Society Press.

- [5] G. Ciardo, R. German and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. In Proc. of the 5th International Workshop on Petri Nets and Performance Models, pages 170-179, Toulouse, France, October 19-22 1993. IEE Computer Society Press.
- [6] A. Cumani. ESP - a package for the evaluation of stochastic Petri nets with Phase-type distributed transition times. In Proc. of the International Workshop on Timed Patri Nets, pages 144-151, Torino, Italy, July 1-3 1985. IEEE Computer Society Press.
- [7] S. Donatelli, S. Haddad and P. Moreaux. Structured characterization of the Markov chains of phase-type SPN. In Proc. of the 10th International Conference on Computer Performance Evaluation. Modelling Techniques and Tools (TOOLS'98), number 1469 in LNCS, pages 243-254, Palma de Mallorca, Spain, September14-18 1998. Springer-Verlag.
- [8] P. Fernandes. Méthodes numériques pour la solution de systèmes Markoviens à grand espace d'états. PhD thesis, INPG, Grenoble, France, 1998.
- [9] S. Haddad, P. Moreaux and G. Chiola. Efficient handling of phase-type distributions in generalized stochastic Petri nets. In Proc. of the 18th International Conference on Application and Theory of Petri Nets, number 1248 in LNCS, pages 175-194, Toulouse, France, June23-27 1997. Springer-Verlag.
- [10] M. K. Molloy. On the integration of delay and throughput in distributed processing models. PhD dissertation, University of California, Los Angeles, CA, USA, September 1981.
- [11] M.F. Neuts and B. M. Rao. On the Design of a Finite Capacity Queue with Phase Type Service Times and Hysteretic Control, European Journal of Operations Research, Volume 62, 221-240, 1992.
- [12] M.F. Neuts and K. S. Meier. On the Use of Phase Type Distributions in Reliability Modeling of Systems with a Small Number of Components, OR Spectrum, 2, 1981, pp. 227-234.
- [13] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms, Proc. of the SIGMETRICS Conference, Texas, 1985, 147-154.