Reliability and Availability Engineering: Modeling, Analysis, Applications

Chapter 12 - Petri Nets

Kishor Trivedi and Andrea Bobbio

Department of Electrical and Computer Engineering
Duke University
Dipartimento di Scienze e Innovazione Tecnologica
Università del Piemonte Orientale

Jan 2017

Outline

Petri Net: Introduction

The state explosion problem implies that the infinitesimal generator matrix of the CTMC is of very large dimensions.

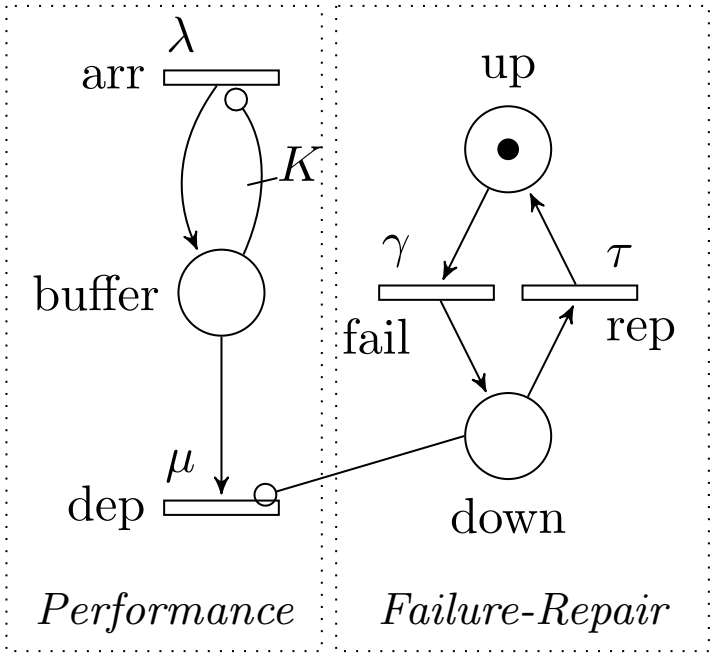
The manual specification/generation of a large matrix is a time consuming and error-prone process that can be alleviated by utilizing a higher level formalism that enables concise specification of complex system behavior.

Petri net (PN), and their variants, are the most popular high level formalisms utilized for this purpose since they allow the automated generation (and solution) of the underlying CTMC.

Queue with breakdown and repair

Returning to the $M/M/1/K$ queue with breakdown, we note that the state space of the CTMC expands with an increase in the value of K.

The model however mixes together behavior (performance and dependability) occurring at two time scales that are orders of magnitude apart. The clear separation of the performance and dependability behavior is unclear from the structure of the CTMC.



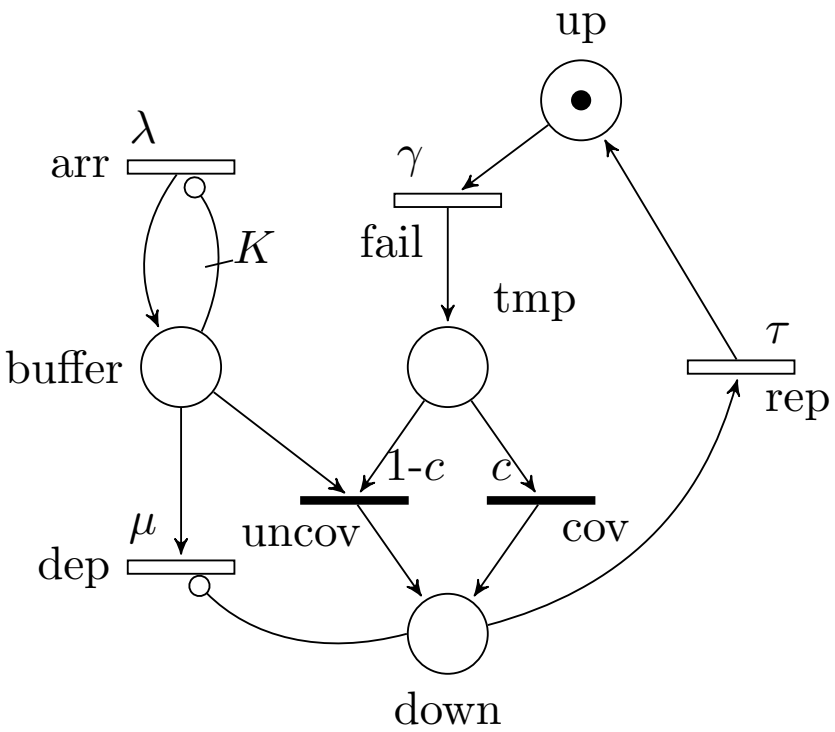
The same behavior expressed as a (stochastic) Petri net model is shown in Figure. In this model, we clearly marked out the part of the model that expresses the performance behavior and the failure-repair behavior.

Queue with breakdown, repair and imperfect coverage

Assume that with probability $1 - c$, the failure of the server will result in the loss of the job currently being served.

This can be represented by adding two more Transitions, *cov* and *uncov*.

These new *immediate* transitions (drawn as black bars) are useful in representing probabilistic branches, and events that take zero time.



From Petri Nets to Stochastic Reward Nets

Petri net

The Petri net (PN) formalism and its theory were first conceived in the doctoral thesis of C.A. Petri in 1962, hence the name.

Since then, the formal language of PN has been developed and used in many theoretical as well as application areas.

A forum related to any aspect of PN with an updated bibliographic database can be found at http://www.informatik.uni-hamburg.de/TGI/PetriNets/.

Petri net

A PN is a bipartite directed graph whose nodes are divided into two disjoint sets called *places* and *transitions*.

Directed arcs in the graph connect places to transitions (called *input* arcs) and transitions to places (called *output* arcs).

A *marked Petri net* is obtained by associating *tokens* with places. A *marking* of a PN is a vector listing the number of tokens in the places of the PN.

In a graphical representation of a PN, places are represented by circles, transitions are represented by bars or rectangles and tokens are represented by black dots or positive integers inside the places.

Petri net

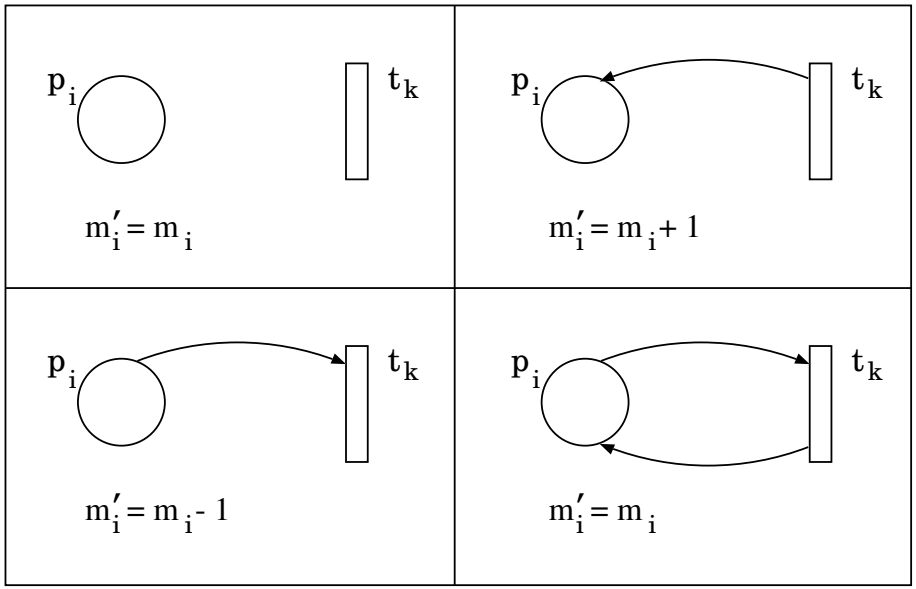
A Marked PN is a quintuple (P, T, I, O, M), where:

- $P = \{p_1, p_2, \ldots, p_{n_p}\}$ is the set of n_p places (drawn as circles in the graphical representation);
- $T = \{t_1, t_2, \ldots, t_{n_t}\}$ is the set of n_t transitions (drawn as bars or rectangles);
- I is the transition input relation and is represented by means of arcs directed from places to transitions;
- O is the transition output relation and is represented by means of arcs directed from transitions to places;
- $M = (m_1, m_2, \ldots, m_{n_p})$ is the initial marking. The generic entry m_i is the number of tokens (typically drawn as black dots) in Place p_i in the marking $M (M \in \mathbb{Z}_{\geq 0}^{|P|})$.

Petri net

A transition is considered *enabled* in the current marking if each input place contains at least one token.

The *firing* of a transition is an atomic action in which one token is removed from each input place of the transition and one token is added to each output place of the transition, possibly resulting in a new marking of the PN.

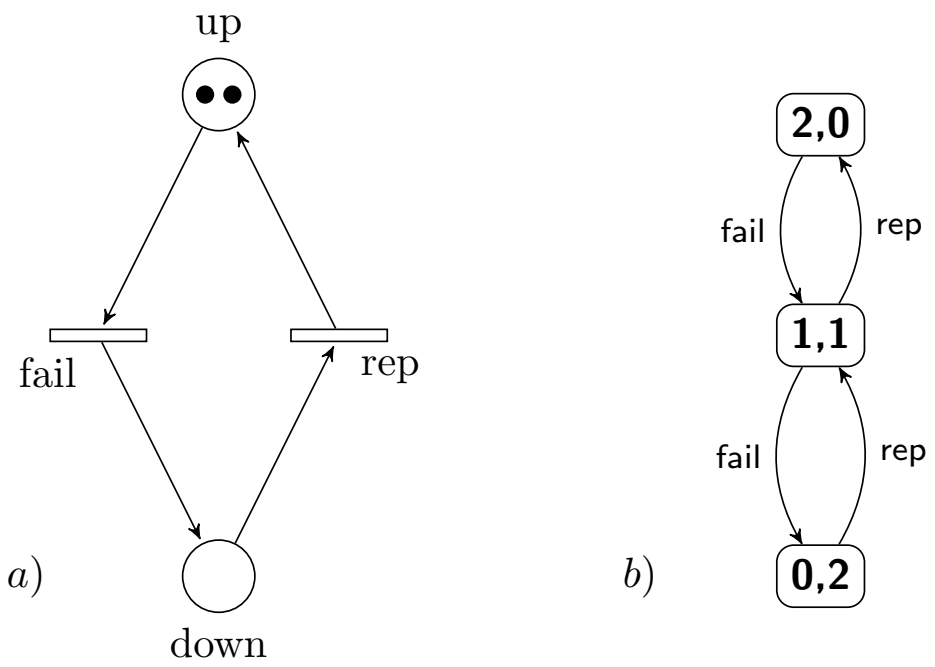


Marking change in Place p_i firing Transition t_k.

Two Repairable Components Availability Model

PN availability model of the two-server parallel redundant system.

In Figure a, the Places *up* and *down* represent the number of servers in the up and down states respectively.



The initial marking of the PN corresponds to both the servers being in the up state. The Transitions *fail* and *rep* respectively represent the failure and repair events in the system.

Reachability Set and Reachability Graph

Each distinct marking of the PN constitutes a separate state of the PN.

A marking is reachable from another marking if there is a sequence of transition firings starting from the original marking which results in the new marking.

In any marking of the PN, a number of transitions may be simultaneously enabled.

Given an initial marking M_1, the *reachability set (graph)* $\mathcal{R}(M_1)$ of a PN is the set (graph) of markings that are reachable from marking M_1.

The initial marking M_1 is the root of the reachability set.

Reachability Set and Reachability Graph

Transition t_k is enabled in marking M if:

$$\text{for any} \qquad p_i \in I(t_k) \qquad , \qquad m_i \geq 1$$

Marking M', obtained from M by firing t_k, is said to be *immediately reachable* from M, and the firing operation is denoted by the symbol $(M - t_k \rightarrow M')$.

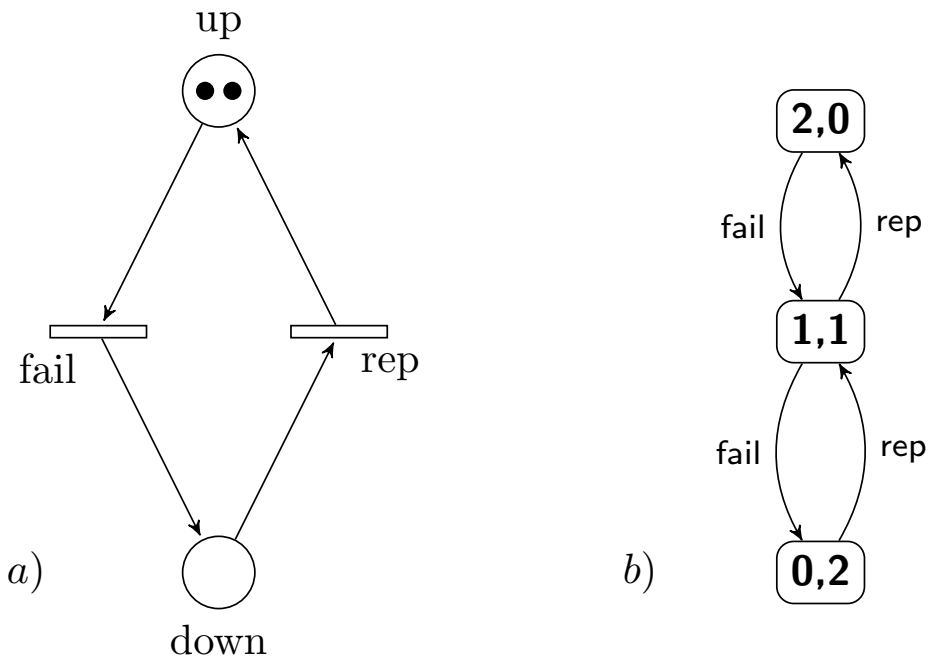
The token count in M' is given by the following relationship:

$$M'(p_i) = \begin{cases} M(p_i) + 1 & \text{if} \quad p_i \in O(t_k)\ , \quad p_i \notin I(t_k) \\ M(p_i) - 1 & \text{if} \quad p_i \notin O(t_k)\ , \quad p_i \in I(t_k) \\ M(p_i) & \text{otherwise} \end{cases}$$

The reachability set can be represented as a labelled directed graph whose vertices are the elements of $\mathcal{R}(M_1)$ and such that for each possible transition firing $M_i - t_k \rightarrow M_j$ there exists an arc (i, j) labelled t_k.

Two Repairable Components: Reachability graph

The reachability graph corresponding to the PN of the two repairable component system is illustrated in Figure b.



The label inside the states denotes the corresponding marking, where the first digit is the number of tokens in Place *up* and the second digit the number of tokens in Place *down*.

The arcs are labeled with the PN transition whose firing causes the change of marking (state).

Structural Extensions to Petri Nets

- Arc multiplicity
- Inhibitor Arc
- Priority Levels

Arc multiplicity

A *multiplicity* may be associated with each arc, and is graphically indicated on the PN by crossing the arc with a dash labelled with an integer indicating the arc multiplicity.

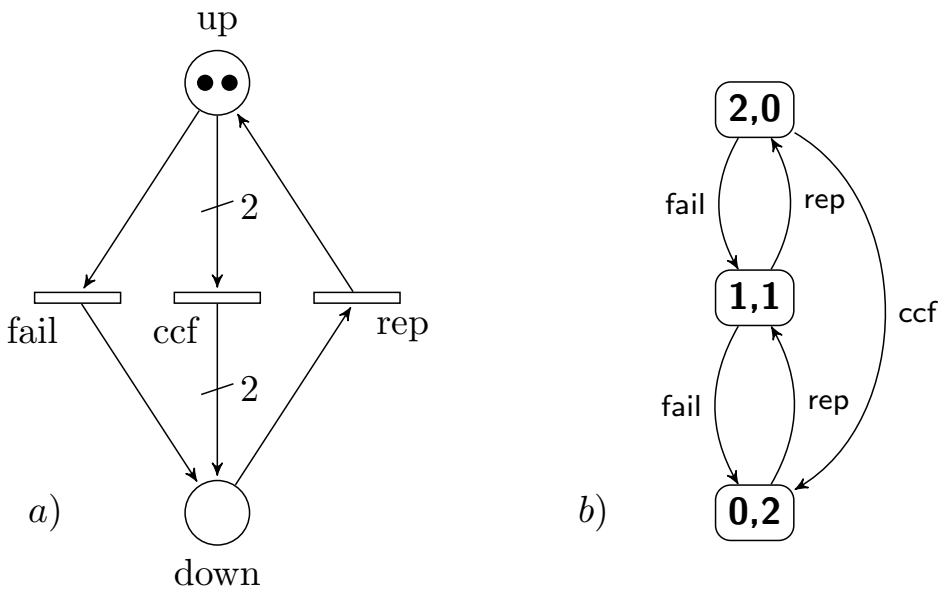
The transition on which an input arc with multiplicity k_i is incident will be considered enabled in the current marking if the number of tokens in the corresponding input place is at least equal to the multiplicity of the input arc k_i from that place.

Upon firing, k_i tokens are removed from the input place.

Similarly a multiplicity k_o associated with an output arc implies that upon firing the number of tokens deposited in the output place is equal to the multiplicity of the output arc k_o.

Two Repairable Components: Common cause failure

The two component system is also affected by a common cause failure that acts simultaneously on the two components.

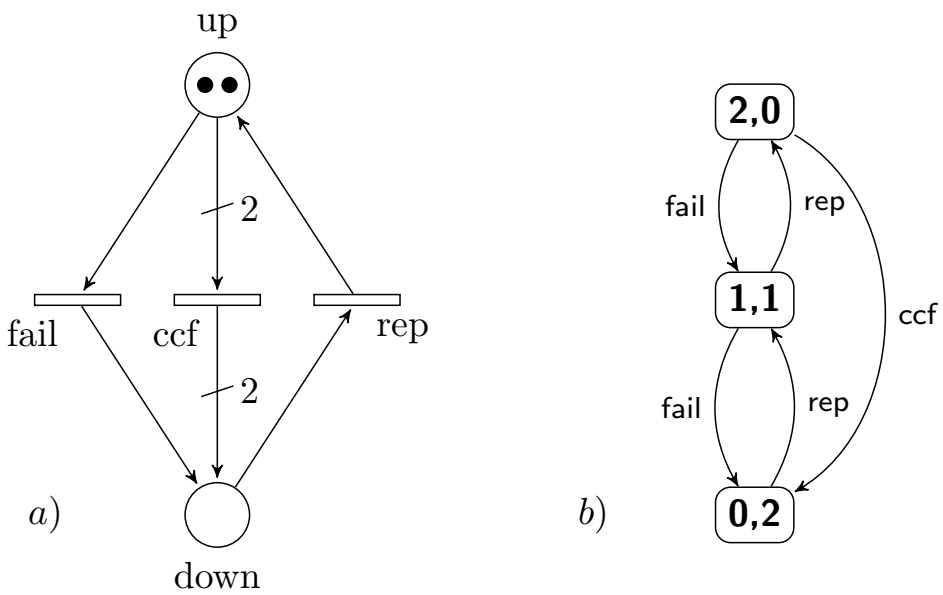


This new situation is represented in the PN of Figure a, by an additional Transition *ccf* with input Place *up* and output Place *down*.

The firing of Transition *ccf* removes two tokens from the *up* place and deposits two tokens in the *down* place.

Two Repairable Components: Common cause failure

The reachability graph in Figure b has an additional arc from Marking $(2, 0)$ to Marking $(0, 2)$ labeled *ccf*.

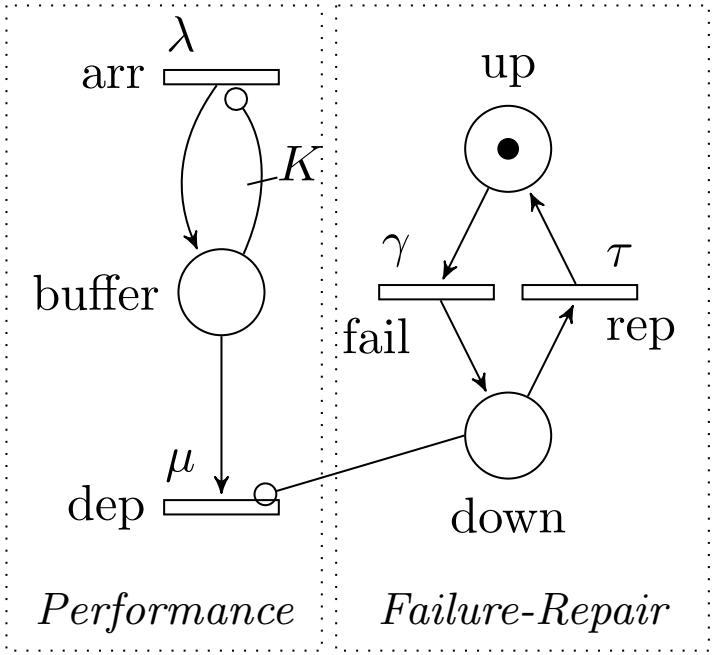


Inhibitor Arc

An *inhibitor* arc drawn from a place to a transition indicates that the transition is not enabled if the place contains at least as many tokens as the multiplicity of the inhibitor arc.

Inhibitor arcs are graphically represented as arcs ending with a small circular head (rather than an arrow), and their multiplicity is specified with the same notation as for the normal arcs.

Queues with breakdown



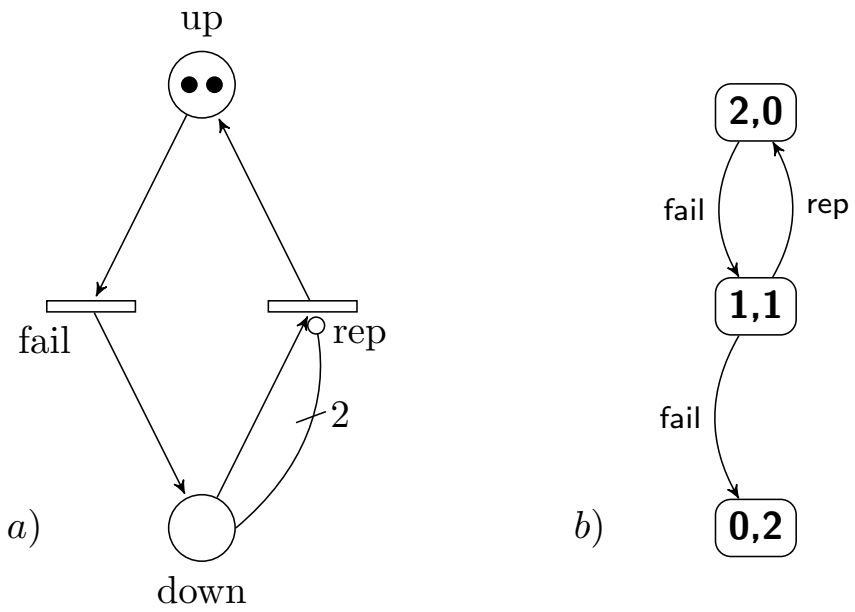
In the PN of Figure there are two inhibitor arcs.

The arc from Place *down* to Transition *dep*, prevents the service from occurring when the server is down.

The arc from Place *buffer* to Transition *arr*, with multiplicity K, blocks new job arrivals when the buffer has reached its maximum capacity.

Two Repairable Components

In the modified PN of the Figure, the assumption is that if both the components have failed, no repair action will be initiated thus giving rise to a PN reliability model.



This condition is modeled using an inhibitor arc with multiplicity 2.

The corresponding reachability graph has an absorbing marking as illustrated in Figure b.

Priority Levels

An alternative, but equivalent way to model the same behavior as represented by inhibitor arcs, can be achieved by attaching to each PN transition a priority level, typically specified using non-negative integers.

The transitions in the PN may be classified into different priority classes, where all the transitions with the same priority value belong to the same priority class.

Whenever a transition with a priority k is enabled, all transitions with priorities less than k are automatically inhibited from firing.

The standard execution rules are modified in the sense that, among all the transitions enabled in a given marking, only those at the highest priority level are allowed to fire.

Stochastic Petri net

Petri nets *per se* do not consider any notion of time as they were meant for logical analysis.

In order to use the PN formalism for the quantitative analysis of the performance and reliability of systems, extensions to PN have been considered by associating *firing times* with the transitions.

The delay between the time instant that a transition is enabled until the time it fires (in isolation) is the firing time of the transition.

We need to add the phrase "in isolation" since when several other concurrent transitions are enabled, firing of any of the concurrent transition may even disable the specific transition under discussion.

When the firing time random variables associated with PN transitions are exponentially distributed, the dynamic behavior of the PN can be mapped into a CTMC with state space isomorphic to the reachability graph of the PN.

Stochastic Petri net

We can formally define the SPN as a sextuple:

$$SPN = (P, T, I, O, M, L)$$

where P, T, I, O, M have the same meanings introduced in untimed PN, and

$$L = \{\lambda_1(M), \lambda_2(M), \ldots, \lambda_{n_t}(M)\}$$

is a set of n_t non-negative real numbers representing the (possibly marking dependent) firing rates of the exponentially distributed random variables associated with each PN-transition.

Note that in the formal definition of the SPN model the firing rates associated with each transition are considered *marking dependent*.

This possibility increases the flexibility of the model and is often used to make the models more compact in the case of the presence of multiple identical resources.

Stochastic Petri net

From the reachability graph it is thus possible to automatically generate the infinitesimal generator matrix $\boldsymbol{Q}$ of the associated homogeneous CTMC.

Here $\boldsymbol{Q}$ is a $n \times n$ matrix, where n is the cardinality of the reachability set $\mathcal{R}(M_1)$.

An updated list of tools that are available to help the user in editing, handling and solving PN is in https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html where the main features of each tool are briefly described.

In this chapter, we refer to the SPNP and SHARPE software packages.

SPN Model of $M/M/1/K$ Queue with Breakdown

Returning to the PN model of the $M/M/1/K$ queue with breakdown, we now associate the rates λ, μ, γ and τ with the transitions *arr, dep, fail* and *rep*, respectively.

With this, the PN now becomes an SPN.

Given a value of K, we can then generate the corresponding reachability graph of the SPN, which will be isomorphic with the CTMC of the $M/M/1/K$ queue with breakdown.

Two Repairable Components

The PN in Figure uses marking dependent rates.

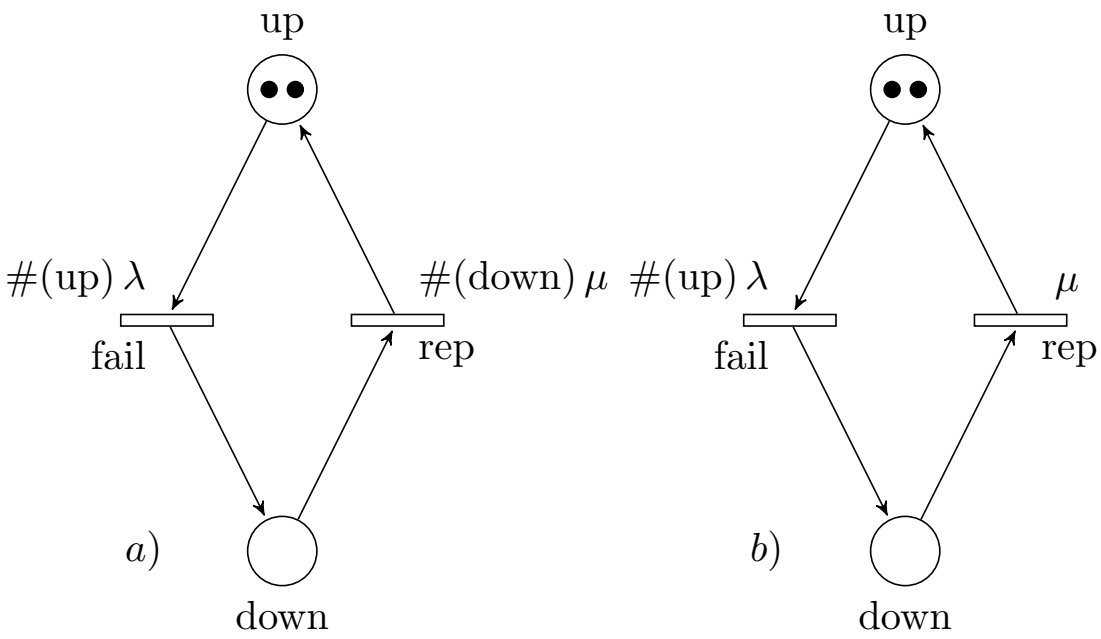
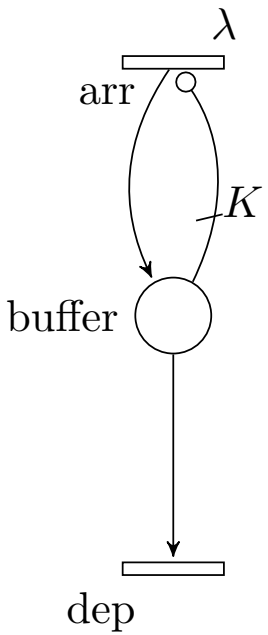


Figure a models non-shared repair;

Figure b models shared repair.

SPN Model of $M/M/m/K$ Queue

The model of the $M/M/m/K$ queue illustrates the modeling power of SPN using the marking dependent firing rate.



$$\mu_{dep} = \begin{cases} i\mu, & 0 \leq (\#(\text{buffer}) = i) \leq m \\ m\mu, & otherwise \end{cases}$$

The specification of the firing rate μ_{dep} in the Figure, indicates that the firing rate of the Transition *dep* is proportional to the number of tokens in the buffer below m, and is constant at the rate $m\,\mu$ thereafter.

Generalized Stochastic Petri Nets

The generalization of SPNs introduced in [*] allows transitions to have either zero firing times (*immediate* transitions) or exponentially distributed firing times (*timed* transitions) giving rise to a model called Generalized Stochastic Petri Nets (GSPN).

Graphically timed transitions are typically represented by empty rectangles while immediate transitions are represented by thin black bars.

Markings (states) enabling immediate transitions are passed through in zero time and are called *vanishing* states.

Markings enabling only timed transitions are called *tangible*.

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

Generalized Stochastic Petri Nets

Since the process spends zero time in the vanishing states, they do not contribute to the time behaviour of the system so that a procedure can be envisaged to eliminate them from the final Markov chain.

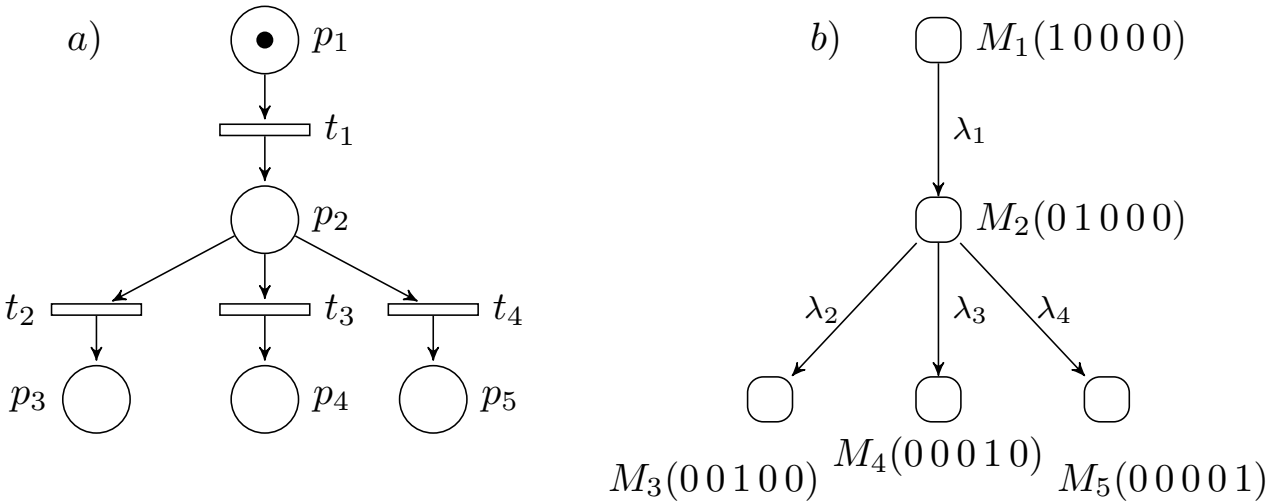
With the partition of PN-transitions into a timed and an immediate class, we introduce a greater flexibility in the modelling power without increasing the dimensions of the final state space of the underlying CTMC.

Analysis of a GSPN involves construction of the underlying reachability graph, referred to as the extended reachability graph (ERG) that contains both tangible and vanishing states.

An arc of the ERG that is associated with the firing of an immediate transition will have a probability of firing while an arc that is associated with a timed transition will have a firing rate.

Elimination of vanishing markings

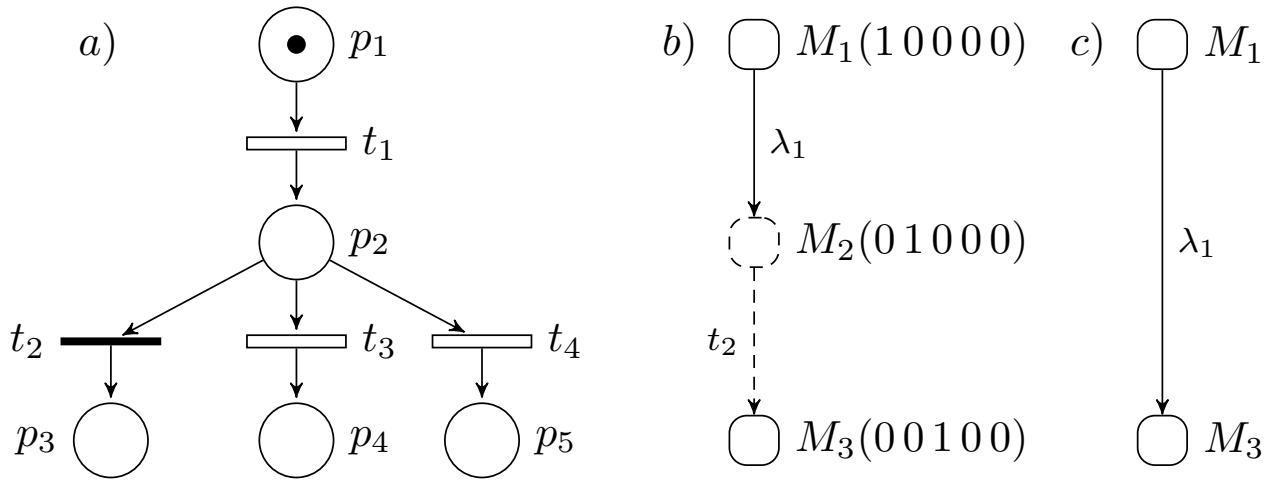
Situation 1 - Only timed transitions are enabled so that only tangible markings are generated.



a) The PN - b) The reachability graph

Elimination of vanishing markings

Situation 2 - One or more timed transitions are enabled simultaneously with one immediate Transition t_2. Only the immediate transition is allowed to fire

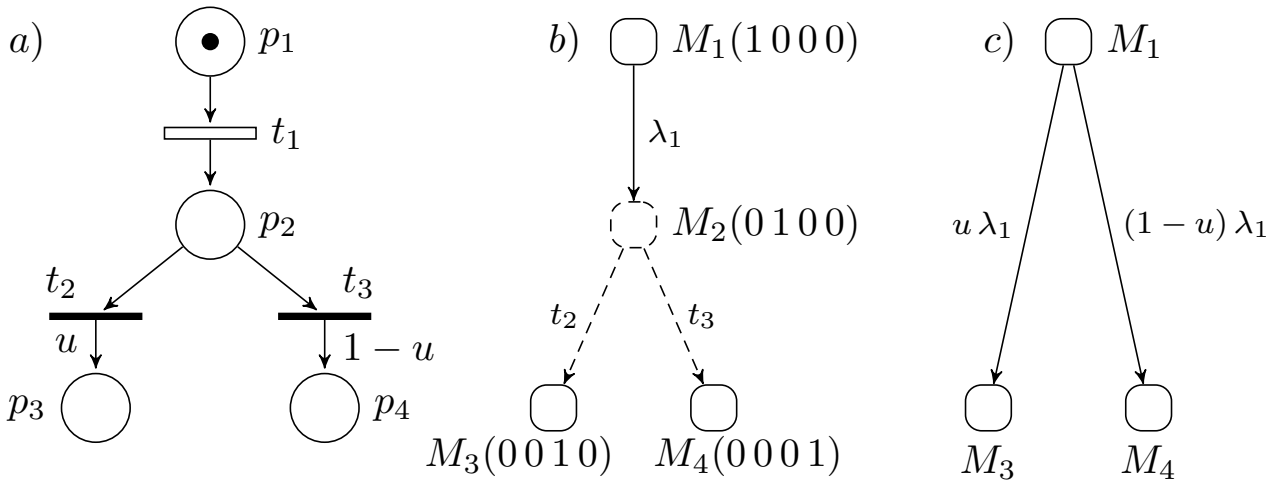


a) The PN - b) The ERG - c) The reduced tangible reachability graph

Elimination of vanishing markings

Situation 3 - Several immediate transitions are simultaneously enabled in a marking.

to determine which immediate transition fires first, a probability mass function needs to be specified: in the language of GSPN this construct is called a *random switch*.

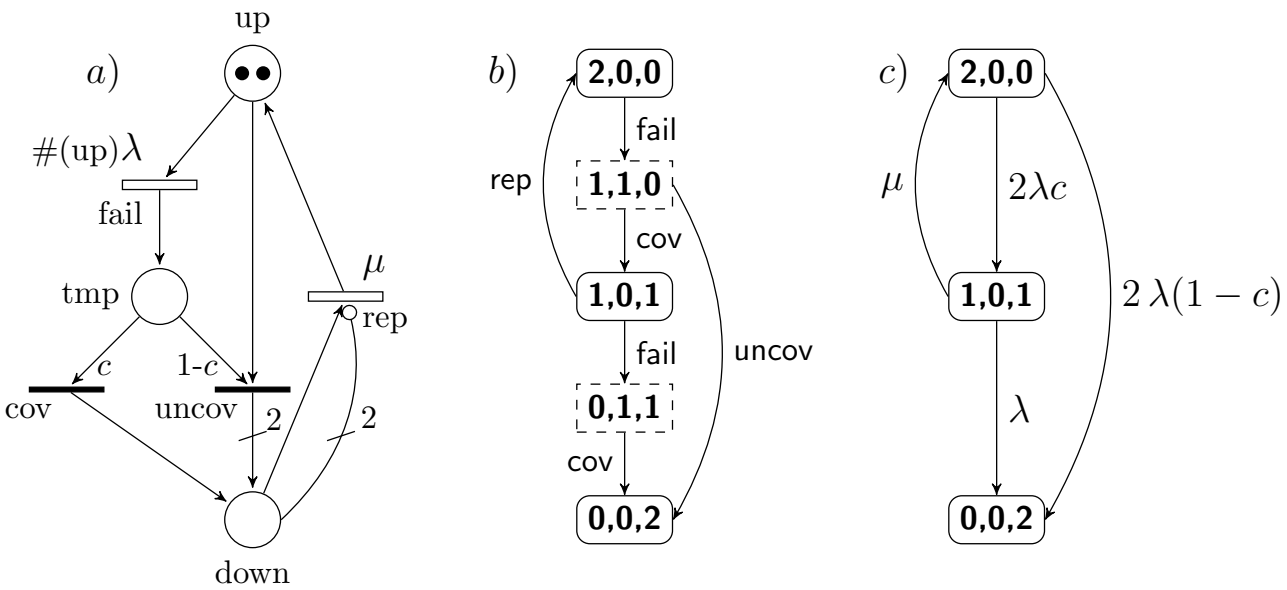


a) The PN - b) The ERG - c) The reduced tangible reachability graph

Two Component Reliability: imperfect coverage

When the server failure occurs, with probability c the failure is covered, and the system is able to recover from this failure.

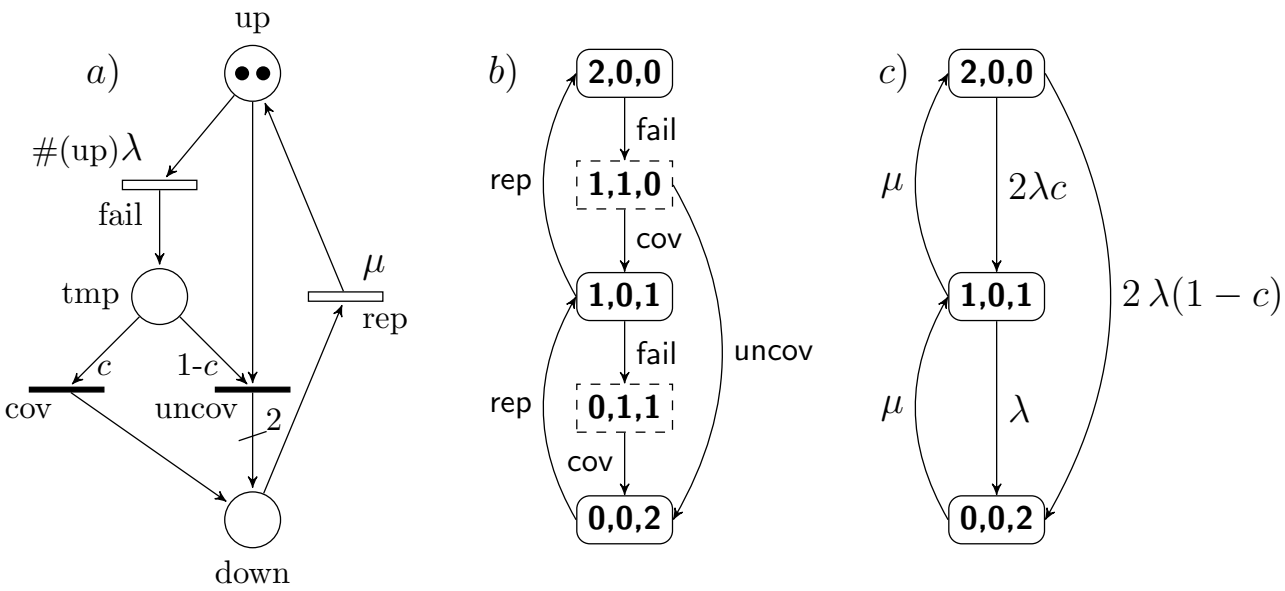
With probability $1 - c$, the server failure causes an irrecoverable failure of the system, and failure of both the servers is catastrophic and the system can no longer be repaired.



a) The PN - b) The ERG - c) The reduced tangible reachability graph

Two Component Reliability: imperfect coverage

We now assume that the system can be repaired (with shared repair) even when both the servers have failed.

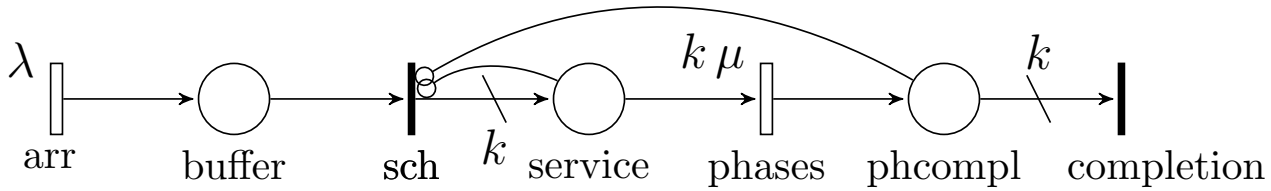


a) The PN - b) The ERG - c) The reduced tangible reachability graph

Two Component Reliability: imperfect coverage

We study a variation of the $M/M/1$ queue where in we introduce k-stage Erlang service time distribution.

A GSPN model of this queue is shown in Figure.



Stochastic Reward Nets