Analysis of Markov Reward Models with Stochastic Petri Nets

Dimitar Radev, Elena Rashkova, Vladimir Denchev

Abstract: *Analysis of Markov reward models with stochastic Petri nets is presented. Generation methods and analysis of continuous-time Markov chains and Markov reward models is provided for modeling of reliability of very large systems and working out measures for their performance. Examples and numerical results for M/D/1/2/2 system are shown.*

Key words: *Queuing Networks, Markov Chains, Markov Reward Model, Petri Nets, Stochastic Reward Nets*

INTRODUCTION

For description of the behavior of real systems can be use Continuous-Time Markov Chains (CTMC). The reliability behavior of real systems in general can be described by reward models, where importance of model generation is large. This importance increases for very large systems, which are closer to real systems. Therefore, it would be attractive to study such large models in a compact way. The compact description has to be based on the identification of regularities or repetitive structures in the system model.

The main results of the study are related to the analysis of stochastic processes including model analysis with stochastic Petri nets. Purposes of provided analysis is modeling of reliability of very large systems and working out measures for their performance.

MARKOV REWARDS MODELS

Performance measures for system behaviour can be introduced with Markov reward models. The problem here is that it is not clear what kind of measures should be generalized and calculated, because each system has different system requirements. The basic system requirements can be generalized into four classes: system availability, system reliability, system performance and task completion.

System availability is the probability of an adequate level of service, or, in other words, the long term fraction of time of actually delivered service. Usually, short outages can be accepted, but interruptions of longer duration exceeding a certain threshold may not be tolerable. System reliability is the probability of uninterrupted service exceeding certain duration of time.

System performance shows the capacity of different configurations into account.

Task completion is reflected in the probability that a user will receive service at the required quality, or in other words, in the proportion of users being satisfied by the received service and it's provided quality. Many different kinds of measures could be defined in the category.

Markov reward models (MRM) provide a unifying framework for an integrated specification of model structure and system requirements. Consider the explicit specification of system requirements as an essential part of the computational model. Once the model structure has been defined so that the infinitesimal generation matrix is known and the basic equations can be written depending on the given system requirements and the structure of the generation matrix. Markov reward models are used in Markov decision theory to assign cost and reward structures to states of Markov processes for an optimization [4, 5]. They provide a framework for an integrated approach for system performance and determining dependences between system characteristics. The term "performability" is referred to measures characterizing the ability of fault-tolerant systems.

Rewards in Markov reward models can be assigned to states or to transitions between states of a Markov chain. The reward rates are defined based on the system requirements, which can be availability, reliability, or task oriented. Let the reward rate r_i is assigned to state $i \in S$. Then, a reward $r_i \tau_i$ is accrued during the sojourn of time τ_i in state *i*. Let $\{X(t), t \geq 0\}$ denote a homogeneous finite-state of a continuous-time Markov chain with state space S. The random variable (1) refers to the instantaneous reward rate of the Markov reward model at time *t*.

$$Z(t) = r_{X(t)} \qquad (1)$$

Note that the difference between reward rates assigned to individual states *i* and the overall reward rate *Z*(*t*) of the Markov reward model is characterizing the stochastic process as a whole. With the instantaneous reward rate of the continuous-time Markov chain defined as (1), the accumulated reward *Y*(*t*) in the finite time interval [0,*t*) is (2).

$$Y(t) = \int_0^t Z(\tau) d\tau = \int_0^t r_{X(t)} d\tau \qquad (2)$$

For example, consider the sample paths of *X*(*t*), *Z*(*t*), and *Y*(*t*) processes, presented on Fig.1. A simple three-state Markov reward model is presented, consisting of a continuous-time Markov chain with infinitesimal generator matrix **Q**, and the reward rate vector **r**=(3,1,0) assigning reward rates to the states 0,1, and 2, respectively. While *X*(*t*) and *Z*(*t*) are discrete valued and non-monotonic functions, *Y*(*t*), in contrast, is continuous-valued, monotonically non-decreasing function.

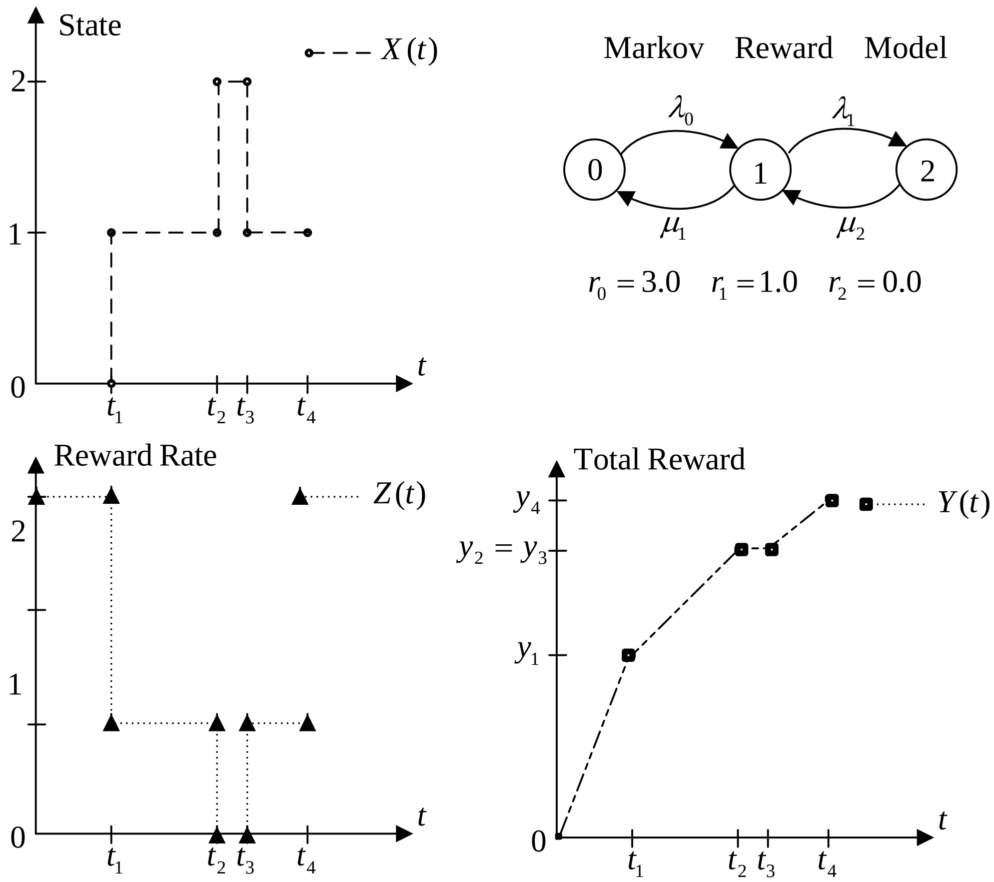

Figure 1. A three-state MRM with sample paths of the X(t),Z(t), and Y(t) processes.

Based on the definitions of *X*(*t*), *Z*(*t*), and *Y*(*t*), which are non-independent random variables, various measures can be defined. The most general measure is referred as the perform ability (3),

$$\Psi(y,t) = P[Y(t) \leq y] \qquad (3)$$

where $\Psi(y,t)$ is the distribution of the accumulated reward over a finite time interval [0,t). But it is difficult to compute for unrestricted models and reward structures. Smaller models can be analyzed via double Laplas transform [1, 2]. The same mathematical difficulties arise if the distribution $\Phi(y,t)$ of the time-average accumulated reward is computed as (4).

$$\Phi(y,t) = P[\frac{1}{t}Y(t) \leq y] \tag{4}$$

The problem is considerably simplified if the system requirements are limited to expectations and other moments of random variables rather than distribution functions of cumulative rewards. Alternatively, efficient solution algorithms are available if the reward ban be limited to a binary structure or if the model structure is acyclic.

GENERATION METHODS

Recall, that continuous-time Markov chains and similar other state-space based models tend to become very large when real-world problems are studied. Therefore, it is attractive to be able to specify such large models in a compact way and avoid the error-prone and tedious creation of models manually. The compact description is based on the identification of regularities or repetitive structures in the system model. Besides reducing the size of the description, such abstractions provide visual and conceptual clarity.

Effects of resource sharing and queuing are naturally represented by means of queuing systems, while synchronization can be easily specified with some Petri net variant. Petri net is a bipartite directed graph, consisting of two types of nodes, namely places *P* and transitions *T*. There exist two types of Petri nets for analysis of stochastic reward models: Global Stochastic Petri Nets (GSPN) and Stochastic Reward Nets (SRN).

Global stochastic Petri nets generalize Petri nets in such a way that each transition has a firing time assigned to it, which may be exponentially distributed or constant zero. Transitions with exponentially distributed firing times are called timed transitions, while the other are referred to immediate transitions. The markings in the reachability set of Petri nets in global stochastic Petri nets, which are partitioned into two sets, are connected with vanishing and tangible markings. Vanishing markings comprise those in which at least one immediate transition is enabled. Immediate transitions always have priority over timed transitions to fire. From a given global stochastic Petri net, an Extended Reachability Graph (ERG) is generated containing the markings of the reachability set as nodes and some stochastic information attached to the arcs.

Stochastic reward nets can be concerned as an extension of global stochastic Petri nets. Some of the most interesting features of stochastic reward nets are arc multiplicity, inhibitor arcs, priorities, guards, marking- dependent arc multiplicity, marking- dependent firing rates, and reward rates, defined at the net level.

The question is which type of Petri nets is more convenient to be used. When more than one token need to be removed from a place or deposited into a place, this can be easily represents with arc multiplicities. It is a syntactical extension, which makes the use of global stochastic Petri nets more convenient. Arcs with multiplicity are presented by a small number attached to the arc or by a small line cutting through the arc.

An inhibitor arcs from place to a transition disables the transition in any marking where the number of tokens in the place is equal to or greater than the multiplicity of an inhibitor arc. Graphically, an inhibitor arc is indicated by a small circle instead of an arrowhead. The use of inhibitor arc is demonstrated in Fig. 2. The figure entails a queue with finite capacity, which customers enter with mean inter-arrival time of $1/\lambda$ as long as the queue is not full, that is, when the fewer than *k* customers are already in the system. Arrivals to a full system may not enter and are considered as lost. The effective arrival level must be less than λ.

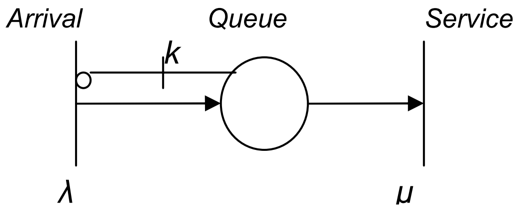

Figure 2. Global stochastic Petri net with inhibitor arc.

Although inhibitor arcs can be used to specify priority relations, it is easier if priority assignments are explicitly introduced in the paradigm. Priorities are specified by integer numbers assigned to transitions. Guards are general predicates that determine when transitions are to be enabled. This feature provides a powerful means to simplify the graphical representation and makes global stochastic Petri nets easier to understand, by example, with identifying modular substructures.

ANALYSIS OF REWARD MODELS WITH GLOBAL STOCHASTIC PETRI NETS

In this section are introduced techniques for automated generation of stochastic processes underlying as GSPN/SRN. In general, the analysis of a GSPN/SRN can be decomposed into four subtasks:

- Generation of the extended reachability graph for defining the underlying stochastic process. A semi-Markov process (SMP) results, with zero or exponentially distributed sojourn times.
- Transformation of SMP into CTMC by elimination of the vanishing markings with zero sojourn times and the corresponding state transitions.
- Steady-state, transient, or cumulative transient analyses of the CTMC.
- Computation of measures. In the case of a global stochastic Petri net, standard measures such as the average number of tokens in each place and the throughput of each timed transition are computed.

In the extended reachability graph which reflects the priorities of the underlying stochastic process, arc representing the firing of timed transitions are labeled with the corresponding rates, and arc representing the firing of immediate transitions are marked with probabilities. An extended reachability graph is a directed graph with nodes corresponding to the markings (states) of a GSPN/SRN and weighted arcs representing the probabilities or rates with which marking changes occur. Starting with an initial marking m_0, subsequent markings are generated by systematically following all possible firing patterns. If a new marking is generated, the arcs labelled with the corresponding rates and probabilities are added to the extended reachability graph. The algorithm terminates if all possible markings have been generated. Three main data structures are used in this algorithm: a stack, the extended reachability graph and a search structure.

The elimination of vanishing markings is an important step to be accomplished for generation of a CTMC from a given GSPN/GRN. Two different techniques can be distinguished "elimination on the fly" and "post-elimination" [6]. The elimination of vanishing markings can either be integrated into the generation of the extended reachability graph (elimination on the fly) or performed afterwards (post-elimination).

Let discuss the merits of the two cases. The elimination on the fly is efficient with respect to memory requirements, because markings need not be stored. But these savings in the space have to be traded with additional cost in time. The same vanishing markings may be hit several times through the ERG - generation process. The elimination step would be repeated although it had been perfectly executed before. The repetition could amount to a significant waste of execution time.

In post - elimination, the complete extended reachability graph is stored during generation. With this technique, it is easier to recognize and resolve (transient) loops consisting of vanishing markings only [3]. Further more, no work is duplicated during extended reachability graph generation, and, as an additional advantage, the complete information included in the extended reachability graph is kept for further use.

NUMERICAL EXAMPLE

Let introduce a Petri net model for non-preemptive M/D/1/2/2 queue, where the steady state solution was derived. We examine two different mechanisms of preemptive service with reliability interpretation of two machines and one repairman. Here the cases are two: when the machines are identical or different.

Let consider the first case: the M/D/1/2/2 queue has preemptive service with the same kind of machines. The repair in execution is preempted as soon as a new demand for repair eventually arrives to the repairman. The preempted repair is restarted as soon as the repairman becomes free again. Two different recovery policies can be considered depending on whether the repairman is able to remember the work already performed on the machine before preemption or not. Figure 3 shows a Petri net, which describes an M/D/1/2/2 system containing two machines and in which any new failure preempts the repair eventually in progress. Place p_1 contains the machines working without failure, while place p_2 contains the number of failed machines (including one under repair). Starting from the initial marking M_1=(2 0 0 1), tr_1 is the only one enabled transition. Firing of tr_1 represents the failure of the first machine and leads to state M_2=(1 1 1 0). In M_2 transitions tr_2 and tr_3 are competing. The repair of failed machine is represented by tr_2 and its firing returns the system to the initial state M_1. Determining of tr_3 is connected with representation the machine and its firing disables tr_2 by removing one token from p_3 (the first repair becomes dormant). In state M_3 = (0 2 0 1) one machine is under repair and one repair is dormant, and the only enabled activity is the repair of the actual machine. Firing of tr_4 leads the system again in state M_2, where the dormant repair is recovered. Assuming the failure time of both machines to be exponentially distributed with parameter *λ*, and tr_1 is associated an exponential firing rate equal to (*2λ*). The value of tr_3 is a firing rate equal to *λ*. Deterministic repair time of duration ϖ is assigned to transitions tr_2 and tr_4.

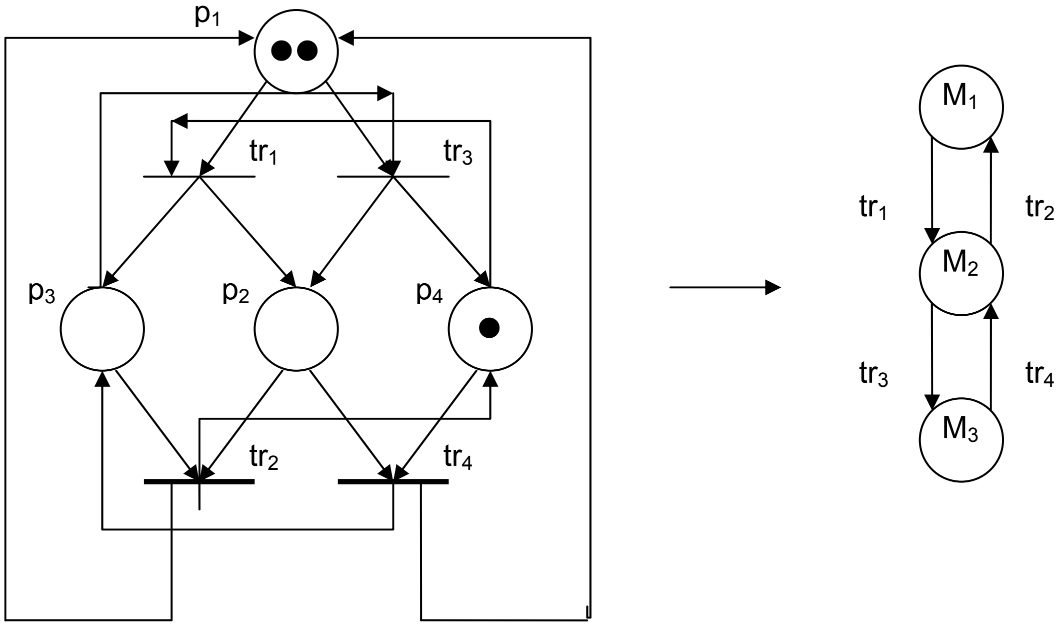

Figure 3. Preemptive M/D/1/2/2 queue with identical machines.

Let introduce the enabling memory policy assigned to tr_2 and tr_3; and age memory policy assigned to tr_2 and tr_3. In first case, each time tr_2 is disabled by the failure of the second machine (tr_3 fires before tr_2), the corresponding enabling age variable a_2 is reset. As soon as tr_2 becomes enabled again (the second repair completes and tr_4 fires) no memory is kept of the prior service, and the execution restarts from scratch. The age memory policy is assigned to tr_2 and tr_3. Each time tr_2 is disabled without firing (tr_3 fires before tr_2) the age variable a_2 is not reset. Hence, as the second repair completes (tr_4 fires), the system returns to M_2 keeping the value of a_2, so that the time to complete the interrupted repair can be evaluated as the residual service time given a_2.a_2 counts the total time during which tr_2 is enabled before firing, and it is equal to the cumulative sojourn time in M_2. The regeneration time points in the marking process $Z_x(t)$ corresponds to the epochs of entrance to markings given in which the age variables associated to all the transitions are equal to zero. In the present example, M_3 can never be a regeneration marking, since a_2 is not reset at entrance to M_3.

CONCLUSIONS

Analysis of Markov reward models with stochastic Petri nets is provided. Purposes of provided analysis is modeling of reliability of very large systems and working out measures for their performance. Effects of resource sharing and queuing are represented by means of queuing systems, while synchronization is specified with some type of Petri nets -GSPN and SRN. A three-state Markov reward model is concerned. Numerical results are shown.

REFERENCES

[1] Ciardo, C., D. Nicol, K. Trivedi. Discrete-event Simulation of Fluid Stochastic Petri Nets. Proc. of the 7th Int. Workshop on Petri Nets and Performance Models, St. Malo, France, Los Alamitos, CA, USA, pp. 217-225, 1997

[2] German, R. Performance analysis of communication systems: Modeling with Non-Markovian Stochastic Petri Nets. Wiley & Sons, Chichester, 2000.

[3] Gribaudo, M., M. Sereno, A. Horvath, A. Bobbio. Fluid Stochastic Petri Nets Augmented with Flush-out Arcs: Modelling and Analysis, Discrete Event Dynamic Systems: Theory and Applications, 11, Kluwer Academic Publishers, pp. 97–117, 2001.

[4] Rácz, S., A. Tari, M. Telek, A distribution estimation method for bounding the reward measures of large MRMs, International Conference on Numerical Solution of Markov Chains - NSMC 2003, Urbana, IL, USA, pp. 341-342, 2003.

[5] Radev, D., S. Radeva, M. Iliev. 2D Steady-state Models of Homogeneous Continuous Time Markov Chains with Vector Quantization. Proceedings of the 26th International Conference Information Technology Interfaces ITI 2004, Cavtat, Croatia, pp. 537-542, 2004.

[6] Wolter, K. Second Order Fluid Stochastic Petri Nets: an Extension of GSPNS for Approximate and Continuous Modelling. Proc. of World Congress on System Simulation, Singapore pp. 328–332, 1997.

ABOUT THE AUTHORS

Assoc. Prof. Dimitar Radev, PhD, Eng. Department of Communication Technique and Technologies, University of Rousse, Phone: +359 82 888 683, E-mail: dradev@abv.bg

Elena Rashkova, PhD student, Department of Comunication Technique and Technologies, University of Rousse; E-mail: elena_pl@gbg.bg

Vladimir Denchev, PhD student, Department of Comunication Technique and Technologies, University of Rousse, E-mail: vdenchev@yahoo.com