MODELING AND SIMULATION OF QUEUE WAITING THROUGH THE CONCEPT OF PETRI NETS

Emilija KAMCEVA [*]
Filip TANCEV [**]

Abstract

Petri Nets-PN are a graphical formalism which is gaining popularity in recent years as a tool in Matlab for the representation of complex logical interactions among physical components or activities in a system. This notes are devoted to introduce the formalism of Petri nets with particular emphasis on the application of the methodology in the area of the performance and reliability modelling and analysis of systems. A technique is presented whereby queueing network models and generalized stochastic Petri nets are combined in such a way as to exploit the best features of both modeling techniques. The resulting hierarchical modeling approach is useful in the solution of complex models of system behavior.

Keywords: *Petri nets, modeling and simulation, toolbox PNTool 2.3, queue.*

Petri nets

Petri Nets offer profound mathematical background originating namely from linear algebra and graph theory. Various Petri Net tools offer convenient graphical environment and sometimes they provide complex simulation and analysis of various high level Petri Net classes.

Petri Net (PN) is mathematical and graphical modeling tool well suited for describing and analyzing discrete events systems (DES). PNs allow to model and visualize systems, which contain concurrence, resource sharing or synchronization. These possibilities allow them to be used for various applications in areas including computer systems, communication protocols, flexible manufacturing systems and software verification.

Within the mentioned context, the initiative of developing instruments for simulation, analysis and design of PNs under MATLAB brought remarkable benefits for training and research because Control Engineering people are familiar with the exploitation of *Graphical User Interfaces* (GUIs)[1] based on this popular software. Although a recent list of the programs developed for PNs includes many resources (Mortensen, 2003) running under different operating systems, our initiative was successful due to the large preference shown for MATLAB.

It is worth separately mentioning that the overall design and implementation philosophy that sustains the ***PN Toolbox***, as well as the integration with MATLAB, allow further

[*] Faculty of Information and Communication Technology, FON University, Skopje, Macedonia (e-mail: emilija.kamceva@fon.edu.mk)
[**] Faculty of Information and Communication Technology, FON University, Skopje, Macedonia (e-mail: tancev_filip@yahoo.com)

[1] The GUI gives the possibility to draw PNs in a natural fashion and allows a straightforward access to various commands starting adequate procedures for exploiting the PN models.

developments in the modern direction of studying hybrid dynamics involving both DES and ODE models.

After ending a simulation experiment, several *Performance Indices* are available to globally characterize the simulated dynamics. Some of the indices recorded for the transitions of the net refer to: the total number of firings during the simulation

(*Service Sum*), the mean frequency of firings (*Service Rate*), the mean time between two successive firings (*Service Distance*), the fraction of time when server is busy (*Utilization*). For the places of the net, the recorded indices refer to: the total number of arrived (*Arrival Sum*) and departed (*Throughput Sum*) tokens, the mean time between two successive instants when tokens arrive in (*Arrival Distance*) and depart from (*Throughput Distance*) the place, the mean time a token spends in a place (*Waiting Time*), the average number of tokens weighted by time (*Queue Length*).

Only for timed or (generalized) stochastic PNs, the time evolution for both current and global values of a *Performance Index* may be displayed dynamically while in the *Step* and *Run Slow* simulation modes by means of the *Scope* command. Another facility available only for timed or (generalized) stochastic PNs is *Design*, which can be used for the synthesis of the models. One or two *Design Parameters* varying within intervals defined by the user can be included in the model. For each test-point belonging to this (these) interval(s) a simulation experiment is performed in the *Run Fast* mode. The dependence of a *Design Index* on the *Design Parameter*(*s*) can be visualized as a graphical plot (2-D or 3-D, respectively).

.

Queueing theory

Queueing theory is the mathematical study of waiting lines, or queues. The theory enables mathematical analysis of several related processes, including arriving at the (back of the) queue, waiting in the queue (essentially a storage process), and being served at the front of the queue. The theory permits the derivation and calculation of several performance measures including the average waiting time in the queue or the system, the expected number waiting or receiving service, and the probability of encountering the system in certain states, such as empty, full, having an available server or having to wait a certain time to be served. Queueing models are generally constructed to represent the steady state of a queueing system, that is, the typical, long run or average state of the system. As a consequence, these are stochastic models that represent the probability that a queueing system will be found in a particular configuration or state.

2.1 M/M/1

The basic queueing model is shown in figure1. It can be used to model, e.g., machines or operators processing orders or communication equipment processing information.

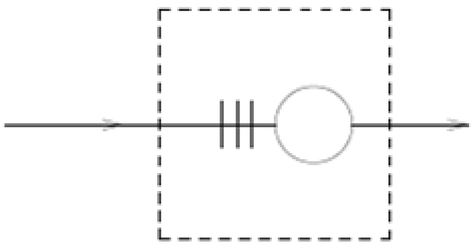

figure 1: Basic queueing model

Among others, a queueing model is characterized by:
- The arrival process of customers.
- The behaviour of customers.
- The service times.
- The service discipline.
- The service capacity.
- The waiting room.

We analyze the model with exponential interarrival with mean $1/\lambda$, exponential service times with mean $1/\mu$ and have a single server. Customers are served in order of arrival. Service rate is $\rho = \frac{\lambda}{\mu} < 1$.

The required characteristics are of great importance can be obtained theoretically for arbitrary values for λ and μ.

The following table shows basic formulas for calculating the important characteristics of the queues.

Performance Measures	$M/M/1$	$M/M/c$	$M/M/1/K$
Traffic Intensity ρ	$\frac{\lambda}{\mu}$	$\frac{\lambda}{c \times \mu}$	$\frac{\lambda}{\mu}$
Utilisation U (per server)	ρ	ρ	$\rho(1 - \frac{(1-\rho)\rho^K}{1-\rho^{K+1}})$
Prob. system is idle π_0	$1-\rho$	$\left(1 + \frac{(c\rho)^c}{c!(1-\rho)} + \sum_{n=1}^{c-1} \frac{(c\rho)^n}{n!}\right)^{-1}$	$\frac{1-\rho}{1-\rho^{K+1}}$
Prob. buffer non-empty B	ρ^2	$\frac{(c\rho)^c}{c!(1-\rho)}\pi_0$	
Mean no. in system N	$\frac{\rho}{1-\rho}$	$c\rho + \rho B/(1-\rho)$	$\frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}$
Mean no. in buffer N_b	$\frac{\rho^2}{1-\rho}$	$\rho B/(1-\rho)$	$\frac{\rho}{1-\rho} - \rho\frac{1+K\rho^K}{1-\rho^{K+1}}$
Mean response time R	$\frac{1}{\mu(1-\rho)}$	$\frac{1}{\mu}\left(1 + \frac{B}{c(1-\rho)}\right)$	$\frac{N}{\lambda(1 - \frac{(1-\rho)\rho^K}{1-\rho^{K+1}})}$

table 1 Basic formulas

The following examples are examined theoretical and simulation values of the queue M/M/1 and M/M/3. Made a comparison of theoretical and simulaciskite values and received a percentage of error.

We consider several cases for M/M/1 queue with Petri nets. Modeling is done and shown in the following figure.

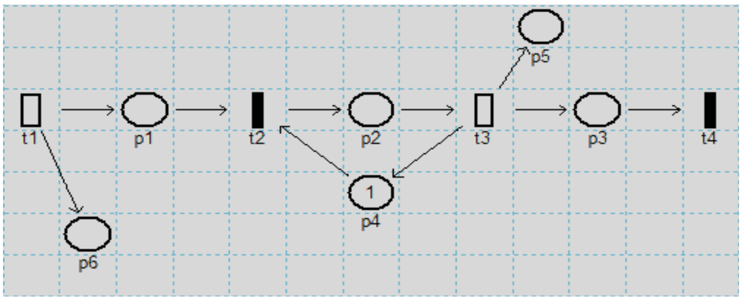

figure 2 Petri nets model on M/M/1

M/M/1	λ	μ	ρ	po	p1	p2	p3	E(X)	E(S)	E(W)
theoretical	0,100	0,400	0,250	0,750	0,188	0,047	0,012	0,333	3,333	0,833
simulation	0,101	0,375	0,269	0,731	0,197	0,053	0,014	0,368	3,645	0,980
percentage of error	0,820	6,213	7,498							

table 2

M/M/1	λ	μ	ρ	po	p1	p2	p3	E(X)	E(S)	E(W)
theoretical	0,100	0,220	0,455	0,545	0,248	0,113	0,051	0,833	8,333	3,788
simulation	0,100	0,219	0,456	0,544	0,248	0,113	0,052	0,837	8,400	3,828
percentage of error	0,312	0,573	0,262							

table 3

M/M/1	λ	μ	ρ	po	p1	p2	p3	E(X)	E(S)	E(W)
theoretical	0,100	0,120	0,833	0,167	0,139	0,116	0,096	5,000	50,000	41,667
simulation	0,100	0,119	0,837	0,163	0,137	0,114	0,096	5,127	51,366	42,983
percentage of error	0,178	0,592	0,416							

table 4

We can be concluded that the percentage of error is less than 10%. Therefore I believe that the simulation values obtained are accurate.

We consider several cases for M/M/3 queue.

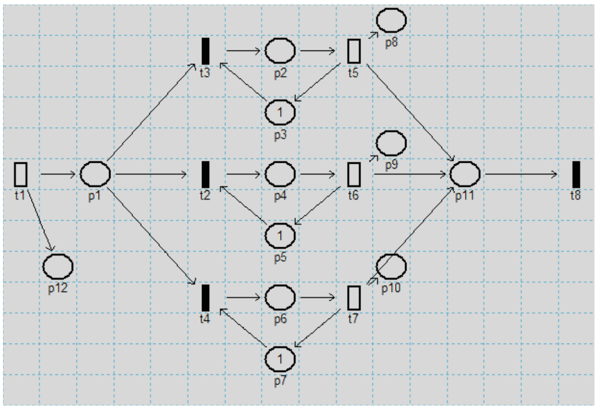

figure 3

M/M/3	λ	μ	ρ	po	p1	p2	p3	E(U)	E(X)	E(S)	E(W)
theoretical	0,100	0,667	0,050	0,868	0,130	0,010	0,000	0,150	0,150	1,500	0,000
simulation	0,100	0,675	0,049	0,870	0,129	0,010	0,000	0,148	0,148	1,481	0,000
percentage of error	0,003	1,250	1,231								

table 4

M/M/3	λ	μ	ρ	po	p1	p2	p3	E(U)	E(X)	E(S)	E(W)
theoretical	1,000	0,351	0,950	0,004	0,012	0,017	0,016	0,328	0,328	3,178	3,178
simulation	1,020	0,327	1,041	0,003	0,009	0,013	0,014	0,342	0,335	3,397	3,465
percentage of error	1,990	6,928	9,581								

table 5

M/M/3	λ	μ	ρ	Po	p1	p2	p3	E(U)	E(X)	E(S)	E(W)
theoretical	1,000	0,337	0,990	0,001	0,002	0,003	0,003	0,332	0,332	3,302	3,302
simulation	0,990	0,330	1,000	0,000	0,000	0,000	0,000	0,333	0,337	3,367	3,3338
percentage of error	0,987	1,990	1,023								

table 6

We can be concluded that the percentage of error is less than 10%. Therefore I believe that the simulation values obtained are accurate. The simple queue of waiting has formulas that can be obtained theoretical values, but for complex queue there are no such formulas. Because the simulation values have a small percentage of error for simple queue are true, then will believe that is correct simulation values wide following a complex queue.

Example: Show Petri net model of queue 3*M/M/1

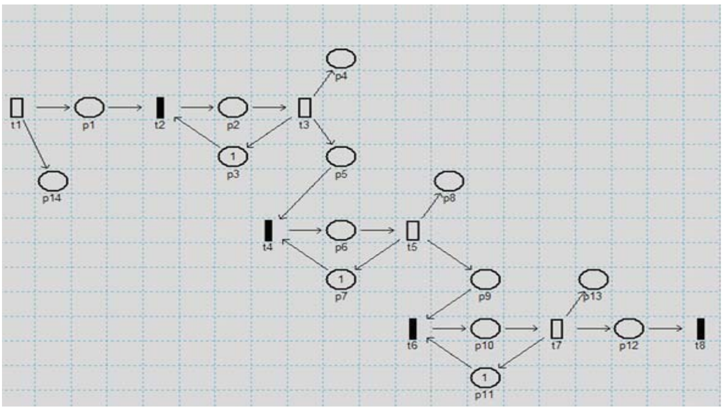

figure 4 **Basic model on 3*M/M/1**

Characteristic simulation values are shown in the following tables. Value of arrival rate λ and service rate μ taken randomly.

Events:10000

Time:6496.6981

Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
p1	1254	0.19302	5.1808	1252	0.19271	5.1891	18.7507	3.6135
p2	1252	0.19271	5.1891	1251	0.19256	5.1932	4.1994	0.80862
p3	1251	0.19256	5.1932	1252	0.19271	5.1891	0.99306	0.19138
p4	1251	0.19256	5.1932	0	0	Inf	Inf	626.9505
p5	1251	0.19256	5.1932	1249	0.19225	5.2015	28.9124	5.5584
p6	1249	0.19225	5.2015	1249	0.19225	5.2015	4.4764	0.8606
p7	1249	0.19225	5.2015	1249	0.19225	5.2015	0.72511	0.1394
p8	1249	0.19225	5.2015	0	0	Inf	Inf	620.5315
p9	1249	0.19225	5.2015	1249	0.19225	5.2015	3.9695	0.76313
p10	1249	0.19225	5.2015	1248	0.1921	5.2057	2.9545	0.56755
p11	1248	0.1921	5.2057	1249	0.19225	5.2015	2.2494	0.43245
p12	1248	0.1921	5.2057	1248	0.1921	5.2057	0	0
p13	1248	0.1921	5.2057	0	0	Inf	Inf	619.2008
p14	1254	0.19302	5.1808	0	0	Inf	Inf	631.3727

table 7

Events:10000

Time:6496.6981

Transition Name	Service Sum	Service Rate	Service Dist.	Service Time	Utilization
t1	1254	0.19302	5.1808	1.3842	0.26718
t2	1252	0.19271	5.1891	0	0
t3	1251	0.19256	5.1932	1.31	0.25226
t4	1249	0.19225	5.2015	0	0
t5	1249	0.19225	5.2015	1.3723	0.26382
t6	1249	0.19225	5.2015	0	0
t7	1248	0.1921	5.2057	1.1283	0.21674
t8	1248	0.1921	5.2057	0	0

table 8

References

- G.A. Agha, F. De Cindio, and G. Rozenberg, editors. Concurrent Object-Oriented Programming and Petri Nets, Advances in Petri Nets 2001, volume 2001. Lecture Notes in Computer Science, Springer-Verlag, 2001.
- M. Abramowitz, I.A. Stegun, Handbook of mathematical functions, Dover, 1965..
- M. Ajmone-Marsan, editor. Proceedings of the 14th International Conference on Application and Theory of Petri Nets, Chicago (USA), volume 691. Lecture Notes in Computer Science, Springer-Verlag, June 1993.
- S. Balsamo and V. de Nitto-Persone. A survey of product-form queueing networks with blocking and their equivalences. Annals of Operation Research, 1994..
- S.L. Brumelle. A generalization of L /W to moments of queue length and waiting times. Operations Research, 1972.
- B. Baumgarten. Petri-Netze. Grundlagen und Anwendungen. BI Wissenschaftsverlag, Mannheim, 1990.
- J. Desel, editor. Proceedings of the 19th International Conference on Application and Theory of Petri Nets, Lisboa (Portugal). Lecture Notes in Computer Science, Springer-Verlag, June 1998.
- J. Desel and J. Esparza. Free Choice Petri Nets. Cambridge University Press, 1995.