# Quasi-Birth-Death Processes

AVACS S3
Phase 2

July 28, 2011

## 1 Description of the Model

This case study is analog to the one described in [2]. We considered a system consisting of a fixed number of $m$ processors and an infinite queue for storing job requests. The processing speed of a processor is described by the rate $\gamma$, while $\lambda$ describes the incoming rate of new jobs. If a new job arrives while at least one processor is idle, the job will be processed directly. Otherwise, it will be put into a waiting queue. If there are idle processors and the waiting queue is non-empty, a job will be taken from the queue and processed immediately. To model this spontaneous transition, a rate $\mu \gg \lambda$ is used. The stochastic Petri net (SPN) used in [2] is depicted in Figure 1 for $m = 3$. Tokens in

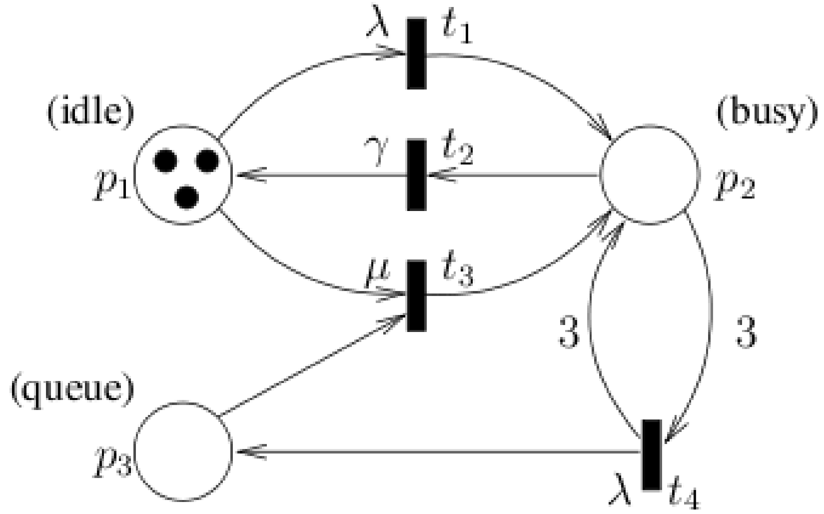


Figure 1: Stochastic Petri Net of the model for $m = 3$

place $p_1$ represent the number of idle processors, place $p_2$ describes the number of busy processors and place $p_3$ gives the number of jobs in the queue. Transition $t_1$ models the case of an incoming job given that at least one processor is idle, whereas $t_4$ describes the case in which all processors are busy, thus the job is put into the queue. Transition $t_2$ represents the successful termination of a job. Finally, $t_3$ is the spontaneous transition in

| $\lambda$ | Uniform | | | Layered | | | FSP exponential | | |
|---|---|---|---|---|---|---|---|---|---|
| | depth | time (s) | $n$ | depth | time (s) | $n$ | depth | time (s) | $n$ |
| 40 | 609 | 0.9/1.2 | 2434 | 533 | 0.9/1.1 | 2130 | 512 | 2.8/1.3 | 2046 |
| 60 | 846 | 0.9/1.6 | 3382 | 754 | 0.9/1.5 | 3014 | 1024 | 4.9/2.1 | 4094 |
| 80 | 1077 | 0.9/2.1 | 4306 | 971 | 0.9/1.9 | 3882 | 1024 | 4.9/2.0 | 4094 |
| 100 | 1305 | 0.9/2.5 | 5218 | 1187 | 0.9/2.4 | 4746 | 2048 | 9.3/4.0 | 8190 |

Table 1: Comparison of three different configurations

case there are idle processors and the queue is non-empty. We consider the probability that, given that all processors are busy and there are no jobs in the queue, within $t$ time units a state will be reached in which all processors are idle and the queue is empty. We can compute the probability by setting $p_1 = 0, p_2 = 3, p_3 = 0$ as the initial state and checking the formula $P_{=?}(F^{\leq t} p_1 = 3 \wedge p_3 = 0)$.

## 2 Results

We implemented this model in *INFAMY* [1] and tested it with three different configurations, various $\lambda$ but fixed $\mu = 1000, m = 3, t = 10$. The precision used in the truncation computation was $10^{-6}$. The probabilites of the property is given in Table 2 and Table 1 compares the three different configurations. All results were obtained on a Linux machine with an AMD Athlon XP 2600+ processor at 2 GHz equipped with 2 GB of RAM.

The uniformization rate of the underlying CTMC is $1000 + \lambda$. With the increase of $\lambda$, the depth grows approximately linearly. Thus, the performance of the FSP configuration suffers from the high cost of repeated transient analysis and, therefore, does not terminate within two hours. As observed by Munsky [3] this problem can be alleviated by adding more than one layer at each step. We consider a variant in which we double the number of layers we add per step, thus computing an error estimate every $1, 2, 4, 8, \ldots$ layers. We call this configuration *FSP exponential.* In contrary to FSP, FSP exponential configuration works reasonable. It is however not competitive with the Uniform or Layered configuration in time and memory.

| $\lambda$ | prob. |
|---|---|
| 40 | 4.22E-04 |
| 60 | 1.25E-04 |
| 80 | 5.27E-05 |
| 100 | 2.70E-05 |

Table 2: Computed probabilites for $P_{=?}(F^{\leq 10} p_1 = 3 \wedge p_3 = 0)$

## References

[1] H. Hermanns, E. M. Hahn, B. Wachter, and L. Zhang. Time-Bounded Model Checking of Infinite-State Continuous-Time Markov Chains. *Fundamenta Informaticae*, 95:129–155, 2009.

[2] Joost Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf. Three-valued abstraction for continuous-time markov chains. In *Proceedings of the 19th international conference on Computer aided verification*, CAV, pages 311–324, Berlin, Heidelberg, 2007. Springer.

[3] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124(4):044104+, 2006.