Perfect simulation of index based routing queueing networks

Jean-Marc Vincent and Jérôme Vienne[1]

Introduction

Markovian queueing networks models are widely used for performance evaluation of computer systems, production lines, communication networks and so on. Routing strategies allocate clients to queues after the end of service. In many situations such as deterministic, probabilistic, or state dependent like *Join the shortest queue* routing, the routing function could be written in terms of index scheduling functions introduced in [3, 6].

When the capacities of queues are finite [7], product form property of the steady-state usually vanishes. Thus approximations [1] could be applied. Finally when analytical methods fail, simulation provides estimates on performances of the network. Drawbacks of simulation are the control of the warm-up period before sampling and the influence of the initial state on the stochastic behavior of the system.

Perfect simulation [8] directly builds steady state samples avoiding the warm-up period and the initial state bias. This method is extremely efficient when the state space is large and the dynamic monotone. The aim of this work is to adapt perfect simulation to queueing networks with index based routing policies. Monotonicity of indexes implies the monotonicity of the global dynamic which improves deeply the simulation time. Previous works on perfect simulation of queueing networks [5, 10, 9] present results on state independent routing.

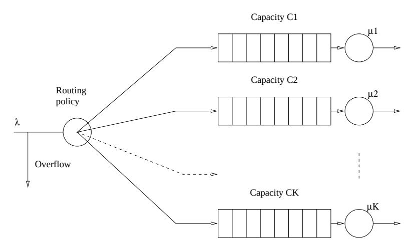
Queuing networks

Consider a queueing network with K finite capacity queues. The state space of each queue Q_i is the set of integers $\mathcal{X}_i = \{0, \cdots, C_i\}$, where C_i is the capacity of queue Q_i. The state space $\mathcal{X}$ of the system is the Cartesian product of all $\mathcal{X}_i$; $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$. The natural order on integer is extended to a partial order on $\mathcal{X}$ using component-wise ordering.

Following the Poisson calculus methodology [2], routing events are driven by homogeneous independent Poisson processes and the dynamic of the system is defined by a Poisson process (uniformization of all the Poisson processes) and a transition function $\Phi(x, e)$ defined for each state x and each event e occuring on a Poisson process.

One should note that the transition function is defined on $\mathcal{E} \times \mathcal{X}$. It is convenient to include the fact that some events could not be applied to a state (not allowed transitions) inside the transition function. For example, the event *end of service* could be executed only if the number of customers in the queue is greater than one.

In a queueing network, a customer arrival, the end of a service and the following routing, a customer departure, are typical events in networks. The transition corresponding to an arrival in queue Q_i is an increment of x_i provided that $x_i < C_i$. In that case one should specify the routing policy (rejection, overflow on another queue,...). Attached with the end of service event e of a customer on Q_i, a routing strategy is defined by a function $\mathcal{R}^e(x)$ on the state space that indicates the destination of the customer. Particular routing functions are defined by index on queues. An index is a function $I_j^e(x_j)$ for each queue Q_j, and the routing function $\mathcal{R}^e(x) = \operatorname{argmin}_i I_i^e(x_i)$, routes the client to the queue with the minimal index value. This formalism supposes that $argmin$ is well defined as if all index values are different.



Bernoulli routing In that case, we have K independent Poisson input processes, with rates $\lambda_i = r_i \lambda$ for Q_i, r_i is the routing probability to Q_i. The corresponding indices are for an arrival a_i on Q_i : for Q_i, $I_i^{a_i}(x_i) = 1\!\!1_{x_i < C_i} + 3 1\!\!1_{x_i = C_i}$ for Q_j, $j \neq i$, $I_j^{a_i}(x_j) = +\infty$ and $I_{K+1}^{a_i} = 2$ with Q_{K+1} is a dummy queue, and corresponds to client rejection.

Priority server policy A client is routed to the first available queue $Q_1, \cdots, Q_K$. Let $(x_1, \cdots, x_K)$ be the state of the system. Indices are defined for the arrival event a :

$$I_i^a(x_i) = \begin{cases} i & \text{if } x_i < C_i \\ +\infty & \text{elsewhere,} \end{cases} \quad \text{and } I_{K+1}^a = K + 1.$$

and allocate clients prioritarly to server 1, 2, . . . ,K, for example in the decreasing order of service rate, and reject the client if all queues are full.

State dependent routing For the *Join the shortest queue* policy, indices are defined by $I_i(x_i) = x_i + 1$ and for the *Join*

[1]Laboratoire ID-IMAG, Mescal-INRIA Projet , 51, avenue Jean Kuntzmann, F-38330 Montbonnot, France {Jean-Marc.Vincent,Jérome.Vienne}@imag.fr This work was partially supported by ACI SurePath project

the shortest response time we have $I_i(x_i) = (x_i + 1)/\mu_i$. Moreover, this work have also been applied to general indices scheduling policies [6].

Theorem 1 *Provided that the indices are non-decreasing functions of state coordinates the routing strategies are monotone.*

In all previous examples, indices are monotone functions of states. One should note that these properties could also be deduced from [4].

Perfect simulation

When the Markov chain is built by a stochastic recurrence equation $X_{n+1} = \Phi(X_n, E_n)$, with $\{E_n\}_{n\in\mathbb{N}}$ an innovation process, we apply the backward simulation method [11] to generate steady-sate samples. When the function $\Phi(x, e)$ is monotone in x for each e, the perfect simulation process is defined by the following algorithm, adaptation of [8].

Algorithm 1 Monotone backward coupling simulation

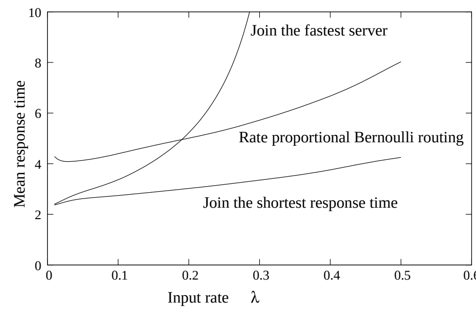
```
n=1;E[1]=Generate-event()
repeat
  n ← 2n;
  y(m) ← minimal state, {System empty}
  y(M) ← Maximal state, {System full}
  for i=n downto n/2+1 do
    E[i]=Generate-event() {generate event −i}
    y(m) ← Φ(y(m), E[i]), y(M) ← Φ(y(M), E[i])
  end for
  for i=n/2 downto 1 do
    y(m) ← Φ(y(m), E[i]), y(M) ← Φ(y(M), E[i])
  end for
until y(m) = y(M)
return y(m)
```

When the algorithms stops the returned value is distributed according to the steady-state distribution.

Experiments have been done on several routing policies. In the following example, we consider 4 queues with capacity 50; the size of the state space is $6.7\,10^6$ and the parameters are $\mu_1 = 0.4$, $\mu_2 = 0.3$, $\mu_3 = 0.2$, and $\mu_4 = 0.1$. The size of samples is 10000 and the performance under study is the response time as a function of the input rate λ. The comparison between these policies is shown in the following figure [1].

The mean simulation time to generate one state, steady-state distributed, is less than 1ms on a Standard PC (speed : 1GHz, memory : 1Go). This is sufficiently small for generating a huge number of independent samples and so increase the accuracy of the results [9].

[1] This work has been done in colaboration with V. Berten, and B. Gaujal



Future works

This method has been fruitfuly applied to several types of monotone routing techniques for finite capacity queueing network. Moreover near optimal allocation policies [6] are currently validated on large systems with multiple servers. This work could be extended to batch arrival or batch services and more generally to monotone events in networks.

References

[1] S. Balsamo, V. De Nitto Person, and P. Inverardi. A review of queueing network models with finite capacity queues for software architectures performance prediction. *Performance Evaluation*, 51(2-4):269–288, 2003.

[2] P. Brémaud. *Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues*. Springer-Verlag, 1999.

[3] J. Gittins. Bandit processes and dynamic allocation indices. *J. Roy. Statist. Soc. (B)*, 41:148–177, 1979.

[4] P. Glasserman and D. Yao. *Monotone Structure in Discrete-Event Systems*. John Wiley & Sons, 1994.

[5] D. Mattson. *On Perfect Simulation of Markovian Queueing Networks with Blocking*. PhD thesis, Chalmers Göteborg University, 2002.

[6] J. Palmer and I. Mitrani. Optimal and heuristic policies for dynamic server allocation. *Journal of Parallel and Distributed Computing*, 65(10):1204–1211, 2005.

[7] H. Perros. *Queuing Networks with Blocking Exact and Approximate Solutions*. Oxford University Press, 1994.

[8] D. Propp, J.and Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9(1&2):223–252, 1996.

[9] J.-M. Vincent. Perfect simulation of monotone systems for rare event probability estimation. In *Winter Simulation Conference*, Orlando, dec 2005. to appear.

[10] J.-M. Vincent. Perfect simulation of queueing networks with blocking and rejection. In *Saint, IEEE Symposium on Applications on the Internet*, pages 268–271, Trento, Jan 2005.

[11] J.-M. Vincent and C. Marchand. On the exact simulation of functionals of stationary markov chains. *Linear Algebra and its Applications*, 386:285–310, 2004.