New Approach Combining Numerical Technique and Simulation for Analysis of Large Discrete Event Systems Based on Petri Nets

Ming-Hung Lin and Li-Chen Fu

**Dept. of Computer Science and Information Engineering,
National Taiwan University,
Taipei, Taiwan, R.O.C
E-mail: lichen@ccms.ntu.edu.tw**

ABSTRACT

In this paper, we propose a new method that combines simulation and numerical techniques and directly integrated using interval arithmetic techniques for the analysis of large discrete event systems. A system is first divided into several layers. Identifying subsystems that can be modeled in isolation solves a system for each layer. In each subsystem, the Markovian assumption allows us to establish a set of linear equality constraints among the expectation of state variables in the Petri Nets, such as token numbers in the places. A pseudo random process is responsible for the timing of the model, event times are completely determined by simulation. Thus, linear equality constraints are computed according to the pre-simulated event times, and it is possible to know the probabilities of interactions between subsystems.

1 Introduction

In this paper, a new analysis method for a discrete event system is introduced. For the analysis of a discrete event system, there are two different approaches can be used, simulation and numerical analysis. Johannes [2] proposes mean value analysis for queueing network models with intervals as input parameters. Shikharesh [3] presents robust bounds and throughput guarantees for closed multiclass queueing networks. The generalized stochastic Petri Net modules are used as basic building blocks to model and analyze complex manufacturing system [6]. Matteo [8] describes approximate mean value analysis for stochastic marked graphs. Zhen Liu [9] proposed performance analysis of stochastic timed Petri Nets using linear programming approach. Numerical analysis techniques and simulation both have their advantages and limitation. The idea of combining analytical/numerical techniques and simulation has been proposed several times. Peter [1] presents a method for continuous time Markov chains. The basis of the method is the description of a continuous-time Markov chain as a set of communicating processes. It also serves as a good tutorial paper. However, in practice, the problem is usually the large size of state space and the model cannot be decomposed very cleanly.

In this paper, we propose a new method that combines simulation and numerical techniques and directly integrated using interval arithmetic techniques for the analysis of large discrete event systems. A system is first divided into several layers. Identifying subsystems that can be modeled in isolation solves a system for each layer. In each subsystem, the Markovian assumption allows us to establish a set of linear equality constraints among the expectation of state variables in the Petri Nets, such as token numbers in the places. A pseudo random process is responsible for the timing of the model, event times are completely determined by simulation. Thus, linear equality constraints are computed according to the pre-simulated event times, and it is possible to know the probabilities of interactions between subsystems. In this way, the state explosion problem of numerical analysis is avoided. It is still possible to obtain more accurate results than with pure simulation.

The organization of this paper is as follows. Section 2 describes several types of combining model for performance analysis. Section 3 introduces a JF queueing network that extends the framework of the original Jackson queueing network by adding event-driven in special queue. Section 4 describes a bounded Petri Net. Section 5 describes the development of integration using interval arithmetic technique. Section 6 is conclusion.

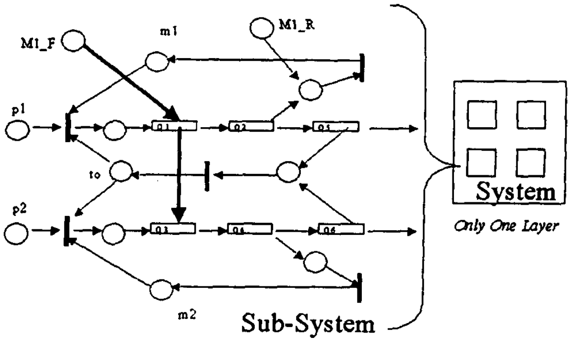

Figure 1: Hybrid model with only one layer

2 Combining Model

There are tow types of model for combining, namely

Hybrid model and *Hierarchical model.* The two model can bring together to solve the whole system. For example, in a flexible-manufacturing cell, there are two processes occurring in parallel. They both use a pool of common tools. Process 1 uses $m1$ machines and process 2 uses $m2$ machines. Fig 1 represents the hybrid model of this system, where

p1 number of process 1.

p2 number of process 2.

m1 available machines for process 1.

m2 available machines for process 2.

to available tools.

Q1 number of process 1 at input-buffer.

Q3 number of process 2 at input-buffer.

Q2 number of process 1 at working-area and being serviced.

Q4 number of process 2 at working-area and being serviced.

Q5 number of process 1 at output-buffer.

Q6 number of process 2 at output-buffer.

M1_F number of failure machine 1.

M1_R number of repaired machine 1.

. Notice that, transferring of a process (after service) from queue Q2,Q4,Q5 and Q6, a token will be generated. In this system, the machine 2 is the back-up machine for machine 1. When machine 1 becomes failure, a new token triggers the instantaneous passage of a process from queue Q1 to Q3.

On the other hand, in hierarchical model with multiple layers, the system performance is calculated layer by layer. The Fig 2 represents this hierarchical model. The upper layer gives a coarse analysis and the lower layer gives a fine one.

A system is first divided into several layers. Identifying subsystems that can be modeled in isolation solves a system for each layer. In each subsystem, we establish a set of linear equality constraints among the expectation of state variables, such as token numbers in the places. A pseudo random process is responsible for the timing of the model, event times are completely determined by simulation. Thus, linear equality constraints are computed according to the pre-simulated event times.

The next key goal is to know the interactions between subsystems. In next section, we proposed a *JF queueing network* that extends the framework of the original *Jackson Queueing Network* to add ability of event-driven.

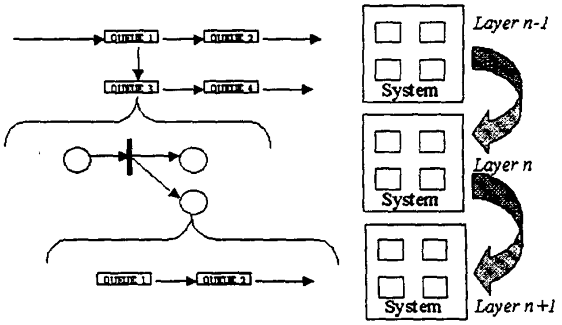

Figure 2: Hierarchical model with multiple layers

3 JF Queueing Network

The Model

There are two types of queues, namely, joining queue and forking queue . The service time of joining queue is infinite. The service time of forking queue follow some probability distribution. The service time of forking queue is zero. When a customer enter a joining queue, the customer will receive no service from the server and will stay in queue for blocking. In our queueing network model, we allow work both to be moved from a queue to another without service, and destroyed, under the effect of control customer which are either exogenous, or generated by the ordinary customers after service. Before presenting our queueing model the necessary terminology is introduced. Let,

C = number of customer classes.

K = number of queues (resources modeled as queueing centers or infinite server).

N_{kc}= number of class $c \in (1, \ldots, C)$customers at queue k.

V_{kc}= mean number of visits made by a class c ordinary customer at queue k.

S_{kc}= mean number of external visits made by a class ordinary customer at queue k.

Z_{kc}= mean number of external visits made by a class control customer at queue k.

R_{kc}= mean service demand rate per visit for a class c ordinary customer at queue k

A customer(control customer or ordinary customer) of class which leaves forking queue i (after finish service) goes to queue j as a customer of class l with probability $p(i,j)(k,l)$, it may depart from the network with probability $P(i)(k) = 1 - \sum_j \sum_l p(i,j)(k,l)$. A customer (control customer or ordinary customer) of class k which leaves joining queue i (after finish service)

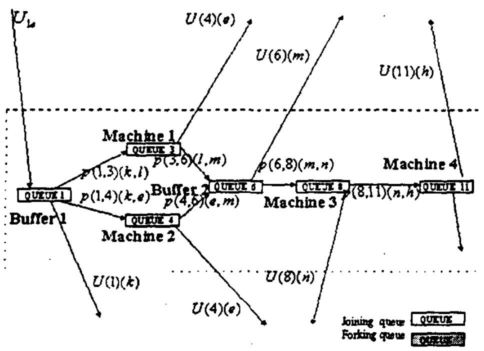

Figure 3: A JF queueing model of a ordinary Jackson queueing network

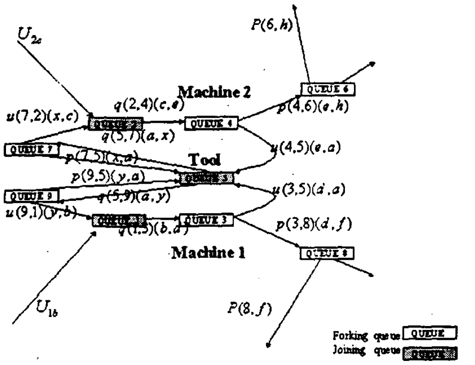

Figure 4: A JF queueing model of share resource

goes to queue j as a customer of class l with probability zero. The arriving control customer triggers the instantaneous passage of a customer(control customer or ordinary customer) of class k from joining queue i to class l of some other queue j with probability $q(i,j)(k,l)$. With probability $Q(i)(k) = 1 - \sum_j \sum_l q(i,j)(k,l)$, it forces a customer to leave the network. After triggering the instantaneous passage of a customer, the arriving control customer is destroyed. The arriving control customer triggers the instantaneous passage of a customer(control customer or ordinary customer) of class k from forking queue i to class l of some other queue j with probability zero.

When a customer(control customer or ordinary customer) of class k which leaves forking queue i (after finish service) , it generates a new control customer from forking queue i to class l of some other queue j with probability $u(i,j)(k,l)$. it may not generate a new control customer into the network with probability $U(i)(k) = 1 - \sum_j \sum_l u(i,j)(k,l)$ When a customer(control customer or ordinary customer) of class k which leaves joining queue i (after finish service) , it generates a new control customer from joining queue i to class l of some other queue j with probability zero.

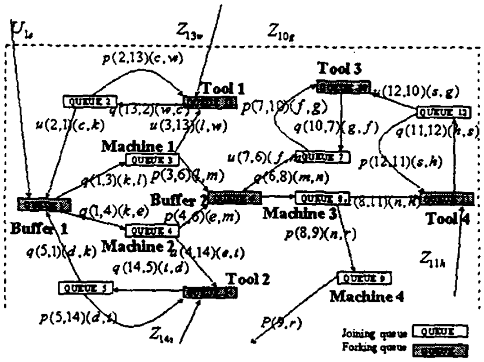

Figure 5: A JF queueing model of a production network

Bounds on throughput

As usual the customers, in whom each queue is, define the operation of the system in terms of the execution of response cycles visited a certain number of times. Let,

W_{kc} = mean waiting time(pure delay) for a class c at queue k. (seconds)

X_c = mean throughput for class c (cycle/second).

Y_c = mean cycle time for a single customer in class c (seconds)

$H_c = \frac{1}{Y_c}$= mean cycle rate for a single customer in class c

The upper and lower bounds on the throughput for different customer classes in a open multiclass queueing network with a FIFO queueing discipline used at each queue (queueing center) are presented in this section.

The first type of upper bound ignores contention at the queue and is termed the no contention upper bound. By ignoring the queueing delay at every queue k we obtain

$$Y_c = \sum_{k=K'}^{K} V_{kc} R_{kc} + \sum_{k=1}^{K'} V_{kc} W_{kc}$$

for each node of forking queue, we have $\forall k \in \{K', \ldots, K\}$

$$V_{kc} = S_{kc} + Z_{kc} + \sum_{k'=1}^{K'} \sum_{c'=1}^{C} V_{k'c'} q(k',k)(c',c)$$

$$+\sum_{k'=K'}^{K}\sum_{c'=1}^{C}\frac{1}{2}V_{k'c'}p(k',k)(c',c)$$

On the other hand, for each node of joining queue, we have $\forall k \in \{1,\ldots,K'\}$

$$V_{kc} = S_{kc} + Z_{kc} + \sum_{k'=K'}^{K}\sum_{c'=1}^{C}\frac{1}{2}V_{k'c'}p(k',k)(c',c)$$

$$+\sum_{k'=K'}^{K}\sum_{c'=1}^{C}\frac{1}{2}V_{k'c'}u(k',k)(c',c)$$

, and form the *little law* we have

$$\sum_{k=1}^{K} N_{kc} = X_c Y_c$$

. Thus,

$$X_c = \frac{\sum_{k=1}^{K} N_{kc}}{Y_c} \leq \frac{\sum_{k=1}^{K} N_{kc}}{\sum_{k=K'}^{K} V_{kc}R_{kc} + \sum_{k=1}^{K'} V_{kc}W_{kc}}$$

. Then,

$$X_c \leq \min\{\frac{\sum_{k=1}^{K} N_{kc}}{\sum_{k=K'}^{K} V_{kc}R_{kc} + \sum_{k=1}^{K'} V_{kc}W_{kc}}, \frac{1}{V_{bc}W_{bc}}, \frac{1}{V_{bc}R_{kc}}\} \quad (1)$$

, where b is the bottleneck queue for class c customers.
Lemma 1 *Consider a closed system in which two customers, i in class a and j in class b, respectively, be at a FIFO queue k. The probability $P(a,b)$ that customer i upon arrival finds customer j in queue or service at k is upper bounded by $P(a,b) \leq \min\{1, \frac{H_b V_{kb}}{H_a V_{ka}}\}$*
Proof : we omit the proof, which follows similar technique [3].
Theorem 1 *Consider a closed multiclass queueing network composed of K queues and C customer classes. If the customers arriving at a queue are served in FIFO order, then,*

$$X_c \geq \frac{\sum_{k=1}^{K} N_{kc}}{\sum_{k=1}^{K} V_{kc}\sum_{n=1}^{C} N_{kn}R_{kn}\min\{1, \frac{H_n V_{kn}}{H_c V_{kc}}\}} \quad (2)$$

Proof : we omit the proof, which follows similar technique [3].

4 Bounded Petri Net

In order to cast the models of such systems in a linear system-theoretic framework, minmax algebra is applied to the Petri net models, and analogous concepts to transfer functions, input-output models, feedback, etc., are developed. This section will introduce some of these basic concepts.

Fig 6 represents the basic Minimax that may be found in any TPPN model, where the time delays d_i are associated with the places. Denote by $t_i(k)$ the time of the kth firing of transition t_i. Then, applying this notation leads to

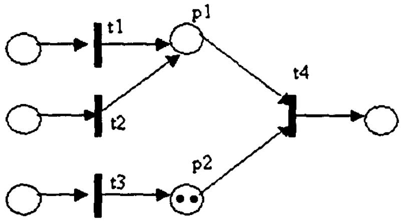

Figure 6: The basic bounded TPPN model

$$t_4(k) = \max\{\min\{p1+t_1(k), p1+t_2(k)\}, p2+t_3(k-2)\} \quad (3)$$

There are recursive equations representing the firing time of the transitions. The real potential for this approach is the ability to draw on analogies with traditional control theory. It is continue to be proven that such analogies are not only with mathematical interest, and that in this approach can be extended to a large model.

However, uncertainties may be associated with Petri Net parameters such as number of tokens in a place. The uncertainties are based on the various reasons. For example, exact values of firing time or exact numbers of arrival process are often unknown in some period of time. An external token visits made by a queueing network can be occur. On the other hand, a Petri net of another subsystem can also make external token visits. If the uncertainty is evaluated with the number of tokens in a place, then Eq 3 can be represented as Eq 4. Where $k^- \leq k \leq k^+$.

$$\begin{aligned}
&[t_4(k^-), t_4(k^+)] = \max\{\min\{[p1 + t_1(k)^-,\\
&p1 + t_1(k)^+], [p1 + t_2(k)^-, p1 + t_2(k)^+]\},\\
&[p2 + t_3(k-2)^-, p2 + t_3(k-2)^+]\}
\end{aligned} \quad (4)$$

5 Integration Technique

A data type called interval, an object represents a real number locating between an upper and a lower bound which may vary during the lifetime of a computation. An interval x written as $[x^-, x^+]$ which means that x lies between a lower bound x^- and upper bound x^+. The main principle is that any operation whose arguments are intervals produces an interval that is guaranteed to contain all possible results of the operations in interval arithmetic;

Addition: $[x^-, x^+] + [y^-, y^+] = [x^- + y^-, x^+ + y^+]$,

Subtraction: $[x^-, x^+] - [y^-, y^+] = [x^- - y^+, x^+ - y^-]$,

Multiplication:

$$[x^-, x^+] * [y^-, y^+] =$$

$$[min(x^-y^-, x^-y^+, x^+y^-, x^+y^+)$$
$$, max(x^-y^-, x^-y^+, x^+y^-, x^+y^+)]$$

,

Division: $[x^-, x^+]/[y^-, y^+] = [x^-.x^+] * [1/y^+, 1/y^-]$,

In queueing network, the input parameters S_{kc} (mean number of visits made by a class c ordinary customer at queue k) and Z_{kc} (mean number of external visits made by a class ordinary customer at queue k) can be evaluated with intervals. They can be replaced by $[S_{kc}^-, S_{kc}^+]$ and $[S_{kc}^-, S_{kc}^+]$ respectively. For the queueing network bounds and Petri Net bounds, the interval arithmetic can be used to calculate separate upper and lower bounds on each class throughput, giving a rectangular box circumscribed around the feasible throughput region. Then, the next essential objective is to integrate those two different models. It is indicates that we must solve the interval parameters.

In the queueing network, the external arrival tokens can be viewed as *samples* over a period Δt. The period may be short or long. In a number of instances the sample mean, or the sample mean and sample variance, are used to estimate the parameters of a hypothesized distribution [14]. If the observations in a sample of size n are $X_1, \ldots, X_n$, the sample mean $\bar{X}$ is defined by

$$\bar{X} = \frac{\sum i = 1^n X_i}{n} \quad (5)$$

and the sample variance, S^2, is defined by

$$S^2 = \frac{\sum i = 1^n X_i^2 - n\bar{X}^2}{n-1} \quad (6)$$

On the other hand, we consider the marking $m_k(p)$ in Petri Net, or m_k for simplicity. Then, assuming that transition t_j fires, $j = 1, \ldots, m$

$$m_{k+1}(p_i) = m_k(p_i) + O(t_j, p_i) - I(p_i, t_j) \quad (7)$$

for all places p_i ,$i = 1, \ldots, n$. Thus, marking $m_k(p)$ be evaluated with intervals. The Eq 7 can be transfer into Eq 8 as following.

$$[m_{k+1}^-(p_i), m_{k+1}^+(p_i)] = [m_k^-(p_i), m_k^+(p_i)] + O(t_j, p_i) - I(p_i, t_j) \quad (8)$$

In Petri Net, the arrival external tokens can be redistributed from the interval input of queueing network. Using Quantile-Quantile Plots technique[14], the redistributed problem can be solved.

6 Conclusion

The present work introduces a new method that combines simulation and numerical techniques and directly integrated using interval arithmetic techniques for the analysis of large discrete event systems. In this way, the state explosion problem of numerical analysis is avoided. It is still possible to obtain more accurate results than with pure simulation. Our future work is development of throughput approximation technique based on interpolation or MVA [2].The interval arithmetic, however, has so-called dependency problem . It is also as our future research.

REFERENCES

[1] Peter Buchholz, "A New Approach Combining Simulation and Randomization for the Analysis of Large Continuous Time Markov Chains.", *ACM Tran. on Modeling and Computer Simulation*, Vol. 8, No. 2, April 1998, pp. 194-222.

[2] Johannes Luthi and Gunter Haring, "Mean Value Analysis for Queueing Network models with intervals as input parameters", *Performance Evaluation*, Vol. 32, pp. 185-215, 1998.

[3] Shikharesh Majumdar and C. Murray Woodside, "Robust Bounds and Throughput Guarantees for Closed Multiclass Queueing Networks", *Performance Evaluation*, Vol. 32, pp. 101-136, 1998.

[4] Charles Knessl and Charles Tier, "Asymptotic Approximations and Bottleneck Analysis in Product Form Queueing Networks with Large Populations", *Performance Evaluation*, Vol. 33, pp. 219-248, 1998.

[5] K. Laevens and H. Bruneel, "Discrete-time Multiserver Queues with Priorities", *Performance Evaluation*, Vol. 33, pp. 249-275, 1998.

[6] Robert Y. Al-Jaar and Alan A. Desrochers, "Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets.", *IEEE Tran. on Robotics and Automation*, Vol. 6, No. 6, Dec. 1990, pp. 621-638.

[7] Rossano Gaeta, "Efficient Discrete-Event Simulation of Colored Petri Nets.", *IEEE Tran. on Software Eng.*, Vol. 22, No. 9, Sep. 1996, pp. 629-639.

[8] Matteo Sereno, " Approximate Mean Value Analysis for Stochastic Marked Graphs.", *IEEE Tran. on Software Eng.*, Vol. 22, No. 9, Sep. 1996, pp. 654-664.

[9] Zhen Liu, "Performance Analysis of Stochastic Timed Petri Nets Using Linear Programming Approach.", *IEEE Tran. on Software Eng.*, Vol. 24, No. 11, Nov. 1998, pp. 1014-1030.

[10] Alan A. Desrochers and Robert Y. Al-Jaar, "Application of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis.", *IEEE Press.*

[11] Ng Chee Hock, "Queueing Modeling Fundamentals.", *John Wiley & Sons Press.*

[12] Leonard Kleinrock, "Queueing Systems Volume 1 : Theory .", *John Wiley & Sons Press.*

[13] Leonard Kleinrock, "Queueing Systems Volume 2 : Computer Applications .", *John Wiley & Sons Press.*

[14] Jerry Banks, John S. Carson, and Barry L. Nelson, "Discrete-Event System Simulation.", *Prentice Hall International, Inc. Press.*