J. King Saud Univ., Vol. 15, *Comp. & Info. Sci.*, pp. 87-116 (A.H. 1423/2003)

A New Petri Net Modeling Technique for the Performance Analysis of Discrete Event Dynamic Systems

Samir M. Koriem, T.E. Dabbous* and W. S. El-Kilani**

Department of Systems and Computer Engineering, Faculty of Engineering, El-Azhar University, Cairo, Egypt

**Higher Technology Institute, 10th Ramadan City, Cairo, Egypt*

*** Department of Information Technology, Faculty of Computer and Information, Menoufia University, Shabeen El-Koum, Egypt*

(Received 14 October 2001; accepted for publication 30 April 2002)

Abstract. An interesting modeling problem is the need to model one or more of the system modules without exposition to the other system modules. This modeling problem arises due to our interest in these modules or incomplete knowledge, or inherent complexity, of the rest of the system modules. Whenever the performance measures (one or more) of the desired modules are available through previous performance studies, data sheets, or previous experimental works, the required performance measures for the whole system can be predicted from our proposed modeling technique. The incomplete knowledge problem of the dynamic behavior of some system modules has been studied by control theory. In the control area, such systems are known as partially observed discrete event dynamic systems, or POS systems. To the best of our knowledge, the performance evaluation of the POS system has not been addressed by the Petri net theory yet. Therefore, in this paper, we propose a new modeling technique for solving this kind of problem based on using the Petri net theory (i.e. Stochastic Reward Nets (SRNs)) in conjunction with the optimal control theory. In this technique, we develop an SRN Equivalent Model (EM) for the modeled system. The SRN EM-model consists of two main nets and their interface nets. One of the main nets represents the part(s) of interest or the known part(s) of the overall POS system that allows us to model its dynamic behavior and evaluate its performance measures. The other main net represents the remaining part(s) of the overall POS system that feeds the part(s) of interest. The well-known maximum principles have been used to develop an algorithm for determining the unknown transition rates of the proposed model. Numerical simulations are given to show that the proposed approach is more effective than the conventional modeling techniques, especially when dealing with systems having a large number of states.

Keywords: Discrete event dynamic systems; stochastic reward nets; largeness problem; parameter identification; partially observed systems.

1. Introduction

Modern technology has increasingly created man-made dynamic systems that cannot be easily described by ordinary or partial differential equations. Examples of such systems are manufacturing systems and computer/communication networks. The activity in these systems is governed by operational rules designed by humans. Therefore, the dynamics of these systems are characterized by asynchronous occurrences of discrete events such as the arrival or departure of a job, or the initiation and completion of a task. Such man-made systems are called Discrete Event Dynamic Systems (DEDS) [1-3].

Over the past few decades, two general directions have emerged for the analysis of DEDS stochastic models. The first direction is based on using the probability theory to develop analytical models for the desired system such as product form queuing network models [4], (semi-) Markov models [5] and Markov reward models [6, 7]. The second direction relies on the simulation technique [2, 8]. In fact, the simulation technique can be used throughout the design stage of the desired system. This application tends to be expensive due to the fact that a large amount of computation time may be needed in order to obtain statistically significant results. On the other hand, analytic modeling is an attractive technique compared to the simulation method, particularly if many different designs are to be evaluated.

Generalized Stochastic Petri Nets (GSPNs) [9, 10], Stochastic Reward Nets (SRNs) [6, 7], well formed nets [11], and Real-nets (R-nets) [12] are examples of stochastic modeling formalisms. Such analytical modeling techniques have been widely used for the performance evaluation and reliability analysis of parallel and distributed systems [12-14], complex systems [15], flexible manufacturing/industrial control systems [16], and fault-tolerant systems [17]. There are several factors that contribute to their successes: the graphical nature, the ability to model parallel and distributed processes in a natural manner, the simplicity of the model, and the firm mathematical foundation. When using stochastic modeling formalisms to describe real systems, several problems can arise such as largeness, parameter estimation, and stiffness. In what follows, we discuss some of these problems.

Largeness tolerance and largeness avoidance are the two main techniques that have been used to overcome the impact of the largeness problem (enlarged size of the underlying Markov process). In the largeness avoidance technique, we do not require the Markov chain generation in its entirety. Examples of largeness avoidance approaches are state truncation [18], state aggregation [19, 20], decomposition [13, 21], state exploration [9, 10], and composition [11, 22]. These methods have proved to be difficult to generalize but are very effective when applicable. Moreover, most of the largeness avoidance techniques are used to obtain an approximate steady state solution. To the best of our knowledge, there is not much research work reported in the literature to obtain an approximate transient state solution for the large state space obtained from the SRN models [13]. In our research work, we concentrate our attention on the SRN modeling technique. SRN represents one of the main stochastic modeling formalisms that can be used for performance evaluation of real computer systems [6][23]. In fact, reward-based models could lead to more concise model specification and yield new measures through model-solving [17, 18].

When we use one of the stochastic modeling formalisms to model the dynamic behavior of the desired system, the transition rates of this model characterize the events occurring during the system evolution. Also, these transition rates represent the main parameter estimation of the stochastic models. The parameter estimation of the DEDS stochastic models is usually a cumbersome problem consuming a lot of time and cost [23, 24]. Parameter estimation is needed for an effective performance analysis of the stochastic models. Measurement or simulation methods have estimated these unknown parameters of the stochastic models through statistical inference approaches. The measurement method relies on obtaining field data by observing the behavior of the desired system, while the simulation method is designed to gather data by running a software model that describes the function and structure of the desired system. However, both techniques are costly and time-consuming [23].

In addition to the stochastic modeling problems that have been extensively discussed in the literature [23], there are still several problems that may face the stochastic net modelers due to the tremendous development in the DEDS systems. One of these problems is the need to model the inherent complexity of real-world systems (with the complex interactions between its modules), although system modelers are commonly concerned in analyzing one or a few of the system modules. In other words, system modelers have to go through the tedious work of modeling all the system modules, although they are interested only in a part of the system [25]. This work requires a great ingenuity of system modelers, which in turn elevates the cost of the modeling phase during system design. To the best of our knowledge, stochastic modeling techniques have not been proposed up till now to manage this problem [26].

Another problem is the difficulty of observing the functionality of some system modules. Most practical systems are built in a modular form. For such systems, the corresponding stochastic models cannot be easily constructed. This problem is known in the literature as partially observed DEDS systems (denoted by POS systems) [27]. To the best of our knowledge, the performance analysis of POS systems has been investigated only by the control theory [27-29] but not yet by the Petri net theory [23].

In this paper, we develop a novel approach for modeling the dynamic behavior of the DEDS systems using the SRN technique. The basic concept of this technique depends on developing an SRN model for the desired system. This model consists of two main nets. These nets are coupled through logical interface nets. The first main net is denoted by the MN-net. The MN-net describes the basic structure of the interesting part(s) of the DEDS that allows us to model its dynamic behavior and evaluate its performance measures. The second main net acts as an auxiliary net (denoted by AN-net). The AN-net is coupled with the MN-net in such a way that the overall equivalent model behaves as the original DEDS for a given performance measure. The AN-net contains an arbitrary number of places. This number defines the AN-net order. The choice of the AN-net order depends mainly on how accurate the proposed model behaves with respect to the original system. With this approach, the system designer can avoid several problems that have been encountered in the literature such as largeness, estimation of SRN parameters, and modeling difficulty of sophisticated systems. One of the major merits of this approach is that we do not need to build the whole SRN model for the desired system in contrary to all largeness avoidance techniques. Further, the proposed modeling approach has the capability of modeling the POS systems.

The rest of this paper is organized as follows. Section 2 illustrates the basic concept of the proposed modeling technique. Section 3 illustrates how to calculate the parameters of the developed system model. Section 4 illustrates the effectiveness of the proposed modeling approach throughout applying it to a real computer system. Section 5 concludes this paper.

2. The Proposed Modeling Technique

The basic concept of the proposed modeling technique depends on building an Equivalent Model (EM) for the desired system. The EM-model consists of two main nets. The first main net is denoted by the MN-net. The MN-net describes the basic structure of the interesting part(s) of the DEDS that allows us to model its dynamic behavior and evaluate its performance measures. The second main net acts as an auxiliary net (denoted by AN-net). The AN-net feeds the MN-net by the required tokens. To allow interaction between the MN-net and the AN-net, we have introduced three interface nets to the EM-model. The first interface net (denoted by MAI-net) works between the MN-net and the AN-net. The second interface net (denoted by AMI-net) works between the AN-net and the MN-net. While the MAI-net controls the flow of tokens from the MN-net to the AN-net, the AMI-net controls the flow of tokens in the other direction. This behavior is performed in a way that allows the preserving of the logical operation of both the MN-net and the AN-net. The third interface net is called the Compensation Net (CN). The CN-net protects the EM-model against deadlocks. Figure 1.a shows a block diagram for the EM-model that consists of the MN-net, the AN-net, the MAI-net, the AMI-net and the CN-net.

To describe the dynamic behavior of the EM-model, we assume that a part of the desired DEDS system is known. Thus, we can represent this part by the SRN technique. Subsequently, this developed SRN model will stand for our MN-net. We assume the MN-net has the following set of places PMN, transitions TMN, and rates (MN.

PMN ({pMN,i; 1(i ((PMN ()} (2.2)

TMN ({tMN,i; 1(i ((TMN ()} (2.2)

$$(MN (\{\lambda_{MN,i}; \quad 1 \leq i \leq |T_{MN}|\} \quad (2.3)$$

We also assume that the tokens can flow through the $|P_{MN}|$-places in n_{MN} combinations (states). In Fig. 1.b, a detailed block diagram of the MN-net is presented. In the following, we describe this figure.

$OP_{MN} \equiv$ is a subset of P_{MN} through which the MN-net exports tokens to the rest of system modules, where the number of its places k_{OP} satisfies $k_{OP} \leq | P_{MN} |$.

$IP_{MN} \equiv$ is a subset of P_{MN} through which the MN-net imports tokens from the rest of system modules, where the number of its places k_{IP} satisfies $k_{IP} \leq | P_{MN} |$.

$OT_{MN} \equiv$ is a subset of T_{MN} through which the MN-net exports tokens to the rest of system modules, where the number of its transitions k_{OT} satisfies $k_{OT} \leq | T_{MN} |$.

$IT_{MN} \equiv$ is a subset of T_{MN} through which the MN-net imports tokens from the rest of system modules, where the number of its transitions k_{IT} satisfies $k_{IT} \leq | T_{MN} |$.

We use the SRN technique to model the other part of the DEDS system that may suffer from any of the problems previously discussed in the introduction. Such a model is called an AN-net. Figure 1.c shows the description of this net. The AN-net is assumed to have the following set of places P_{AN}, transitions T_{AN}, and rates Λ_{AN}.

$$P_{AN} \equiv \{p_{AN,i}; \quad 1 \leq i \leq k_{AN}\} \quad (2.4)$$

$$T_{AN} \equiv \{t_{AN,i}; \quad 1 \leq i \leq |T_{AN}|\}$$
$$|T_{AN}| = k_{AN}(k_{AN} - 1) \quad (2.5)$$

$$\Lambda_{AN} \equiv \{\lambda_{AN,i}; \quad 1 \leq i \leq |T_{AN}|\} \quad (2.6)$$

Where k_{AN} denotes the number of P_{AN} places in the AN-net. The construction of the AN-net implies that the tokens can flow through the k_{AN} places of the AN-net in n_{AN} combinations (states), where

$$n_{AN} = 2^{k_{AN}} \quad (2.7)$$

The number k_{AN} is used to define the AN-net order. This number can be arbitrarily chosen. Therefore, the measures obtained from the proposed EM-model are sufficiently close to those of the original DEDS measures. In fact, one may think of an optimal number of $k_{AN,opt}$ that achieves this requirement. The main function of the AN-net is to supply the MN-net with tokens through the logical interface AMI-net. The AN-net receives (sinks) its tokens through the other logical interface MAI-net (see Figure 1.a). The interfaces AMI-net and MAI-net have been introduced to the proposed EM-model to guarantee proper interaction between the MN-net and the AN-net. In other words, the interfaces AMI-net and MAI-net prevent the formation of deadlocks in both the MN-net and the AN-net. In what follows, we describe both of the interfaces MAI-net and AMI-net.

The interface MAI-net behaves as an outlet for the tokens of the MN-net. The MAI-net has the responsibility of preventing the deadlocks to allow the proper logical evolution for the AN-net. The behavior of the MAI-net is accomplished by controlling the flow of tokens from OP_{MN} and OT_{MN} to the places of the AN-net. Fig. 1.d shows the following two forms of the MAI-net. One connects each place of OP_{MN} to the places of the AN-net, while the other connects each transition of OT_{MN} to the places of the AN-net. To perform this connection, the MAI-net utilizes the following set of transitions T_{MAI}, rates Λ_{MAI}, and places P_{MAI}.

$$T_{MAI} \equiv \{t_{MAI,i};\ 1 \le i \le |T_{MAI}|\}$$
$$|T_{MAI}| = k_{AN} \times (k_{OP} + k_{OT}) \tag{2.8}$$

$$\Lambda_{MAI} \equiv \{\lambda_{MAI,i};\ 1 \le i \le |T_{MAI}|\} \tag{2.9}$$

$$P_{MAI} \equiv \{p_{MAI,i};\ 1 \le i \le |P_{MAI}|\}$$
$$|P_{MAI}| = k_{AN} \times k_{OT} \tag{2.10}$$

The interface AMI-net behaves as an inlet for the MN-net tokens. The AMI-net guarantees a proper logical evolution for the MN-net according to the original system physical operation. The behavior of the AMI-net is accomplished by controlling the flow of tokens from the k_{AN} places of the AN-net to IP_{MN} and IT_{MN}. Figure 1.e shows the following two forms of the AMI-net. One connects the k_{AN} places of the AN-net to a place of IP_{MN}, while the other connects the k_{AN} places of the AN-net to a transition of IT_{MN}. To perform this connection, the AMI-net utilizes the following set of transitions T_{AMI}, rates Λ_{AMI}, and places P_{AMI}.

$$T_{AMI} \equiv \{t_{AMI,i};\ 1 \le i \le |T_{AMI}|\}$$
$$|T_{AMI}| = k_{AN} \times (k_{IP} + k_{IT}) \tag{2.11}$$

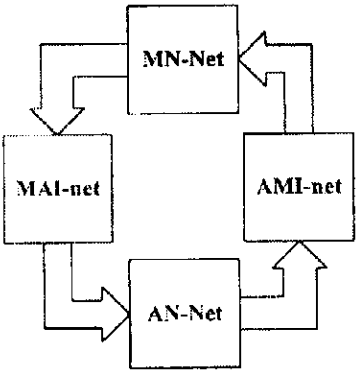

Fig. 1.a. Block diagram of the equivalent model (EM).

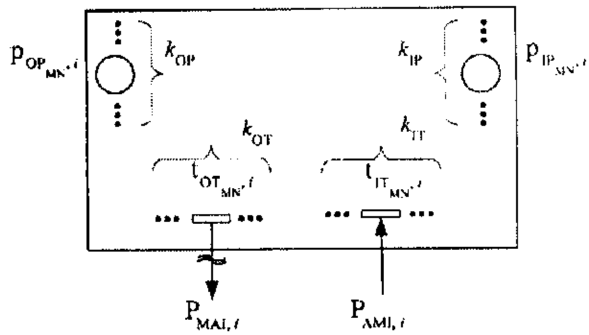

Fig. 1.b. The module SRN (EM-net).

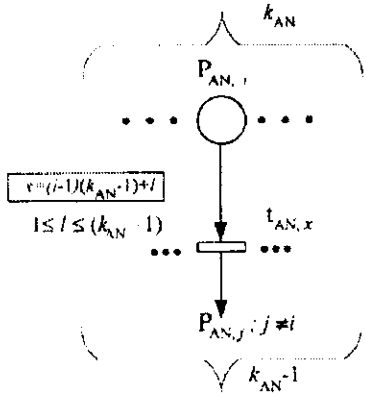

Fig. 1.c. The auxiliary SRN (AN-net).

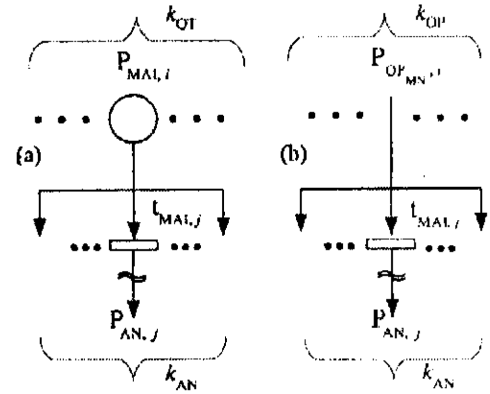

Fig. 1.d. The two forms (a), (b) of the module auxiliary interface SRN (MAI-net).

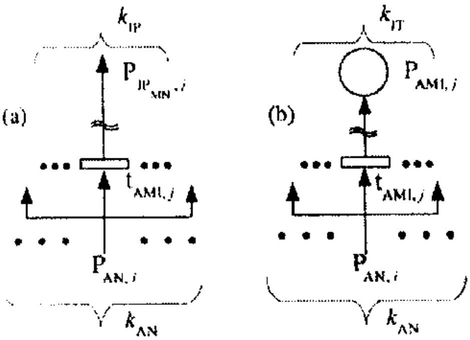

Fig. 1.e. The two forms (a), (b) of the auxiliary module interface SRN (AMI-net).

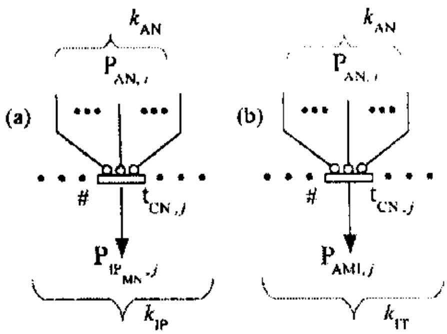

Fig. 1.f. The two forms (a), (b) of the compensation SRN (CN-net).

Fig. 1. The equivalent model (EM) SRN's.

$$\Lambda_{AMI} \equiv \{\lambda_{AMI,i};\ \ 1 \le i \le |\ T_{AMI}|\} \tag{2.12}$$

$$\begin{aligned} P_{AMI} &\equiv \{p_{AMI,i};\ \ 1 \le i \le |P_{AMI}|\} \\ |P_{AMI}| &= k_{AN} \times k_{IT} \end{aligned} \tag{2.13}$$

The control action of the AMI-net and the MAI-net is achieved by governing the arcs connecting the MAI-net and the AMI-net to the AN-net and the MN-net respectively, using arc weight rules concluded from the desired logical operation of the MN-net and the AN-net. Moreover, the flow of tokens from the MN-net to the MAI-net is similarly governed by the arc weight rules. The governing of arcs is shown by the presence of "≈" on the arcs (see Figs. 1.d and 1.e).

The lack of tokens in the AN-net may cause deadlocks in the behavior of the overall system. Therefore, we have introduced the CN-net to supply the MN-net with tokens in the absence of tokens in the AN-net. Figure 1.f shows two forms of the CN-net. The first form connects the k_{AN} places of the AN-net with each place of IP_{MN}. Similarly, the second form connects the k_{AN} places of the AN-net with each transition of IT_{MN} through the places P_{AMI}. To perform this connection, the CN-net utilizes the following set of transitions T_{CN}, rates Λ_{CN}, and places P_{CN}.

$$\begin{aligned} T_{CN} &\equiv \{t_{CN,i};\ 1 \le i \le |T_{CN}|\} \\ |T_{CN}| &= k_{IP} + k_{IT} \end{aligned} \tag{2.14}$$

$$\Lambda_{CN} \equiv \{\lambda_{CN,i};\ \ 1 \le i \le |T_{CN}|\} \tag{2.15}$$

$$\begin{aligned} P_{CN} &\equiv \{p_{CN,i};\ \ 1 \le i \le |P_{CN}|\} \\ |P_{CN}| &= k_{IT} \end{aligned} \tag{2.16}$$

Unfortunately, the MN-net may not need any tokens while there is a lack of tokens in the AN-net. In this case, we must guarantee that the CN-net does not provide the MN-net with tokens. To accomplish this objective, each transition of T_{CN} is controlled by an enabling function. The enabling function is denoted by "#" as shown in Fig. 1.f.

3. Estimation of the Parameters of the EM-model

To use the SRN EM-model, the transition rates of AN-net, AMI-net, MAI-net and CN-net need to be determined. In this section, we use the well-known optimal control theory along with the underlying Markov Reward Process (MRP) resulting from firing the transitions of this SRN model, to develop an algorithm for determining these unknown parameters. For this purpose, we assume that one of the transitions of the EM-model is t_k. Further, suppose that the firing of t_k transfers the SRN EM-model from a state S_i to a state S_j, where

$$S_i, S_j \in \text{RS} \equiv \{ S_l;\ 0 \leq l \leq N\text{-}1\}, \quad i \neq j \tag{3.1}$$

With N being the total number of states that the underlying MRP may have and RS denotes the Reachability Set [11][24]. Let $\pi_i(\theta);\ \theta \geq 0$ be the probability that the SRN model is in the state S_i at time θ. It is well-known that the process $\pi_i(\theta)$ is governed by the following Kolmogorov forward (differential) equation [18, 30, 31].

$$\frac{d\pi(\theta)}{d\theta} = \pi(\theta)\,\text{Q};\ \theta \geq 0 \qquad 3.2)$$

Where π is a vector defined as $\pi \equiv \{ \pi_i;\ 0 \leq i \leq N\text{-}1\}$, with

$$\sum_{i=0}^{N-1} \pi_i(\theta) = 1 \qquad \text{for all} \theta \geq 0 \tag{3.3}$$

In equation 3.2, the matrix Q represents the infinitesimal generator of the MPR. Each element of the matrix Q (denoted by q_{ij}; $0 \leq\ i,j\ \leq N\text{-}1,\ \ i \neq j$) is related to the transition rate λ_t that corresponds to the transition $t \in T$, where T is the set of all transitions of the SRN EM-model. The relation between q_{ij} and λ_t can be defined as follows.

$$q_{ij} = \sum_{t \in T: S_i \xrightarrow{t} S_j} \lambda_t \tag{3.4}$$

With the constraints that for each $0 \leq i \leq N\text{-}1$,

$$q_{ii} = -\sum_{\substack{j=0 \\ i \neq j}}^{N-1} q_{ij} \tag{3.5}$$

Let $X(\theta)$; $\theta \geq 0$ be one of the given performance measures. Let $\hat{X}(\theta)$; $\theta \geq 0$ be the estimated measure of $X(\theta)$ obtained from the EM-model. The measure $\hat{X}(\theta)$ may have the following known form [6, 7].

$$\hat{X}(\theta) \equiv K_1 \sum_{S_i \in RS} \rho_i(\lambda,\theta)\, \pi_i(\theta) + K_2 \sum_{S_i \in RS} \rho_i(\lambda,\theta)\sigma_i(\theta);$$

$$\sigma_i(\theta) = \int_0^{\theta} \pi_i(u)du \qquad (3.6)$$

Where K_1 and K_2 are given constants chosen according to the measure required, and $\rho_i(\lambda, \theta)$ represents the reward value for the state S_i ($0 \leq i \leq N-1$) at time θ, where $\rho_i > 0$ and $\lambda \in \Lambda \equiv \{\lambda_t;\ t \in T\}$. Also, $\sigma(\theta)$ represents the expected time spent in a state S_i during the interval $[0,\theta]$. As can be observed from equation 3.6, the measure $\hat{X}(\theta)$ depends on the transition rate λ and the process $\pi_i(\theta)$. Also, $\pi_i(\theta)$ depends on the transition rate λ as shown in equations 3.3 and 3.4. Hence, the transition rates can be chosen so that the estimated measure $\hat{X}(\theta)$ obtained from the SRN model is fairly close to the given measure $X(\theta)$ of the original system, for all $\theta > 0$.

Since the EM-model consists of the MN-net and the AN-net, the matrix Q of the MRP that represents the EM-model will be of dimension $n_{MN} \times n_{AN}$ (see equation 2.7). Further, $\hat{X}(\theta)$ (as given by equation 3.6) depends on the transition rate λ (see equations 3.2, 3.3, and 3.4). Hence, we use the symbol $\hat{X}(\theta,\lambda)$ instead of $\hat{X}(\theta)$ to indicate such dependence.

According to the above preparation, we can now state the identification problem of the EM-model as follows. Given the performance measure $X(\theta)$ of the original DEDS model, the transition rate $\lambda^{\circ} \in \Lambda$ can be defined as follows.

$$J(\lambda^{\circ}) \leq J(\lambda) \quad \text{for all } \lambda \in \Lambda$$

$$J(\lambda) = \frac{1}{2} \int_0^{T_f} w(X(\theta) - \hat{X}(\theta,\lambda))^2 \, d\theta \qquad (3.7)$$

Where T_f is the estimation time of $X(\theta)$ and w is some suitable weighting factor. For determining the optimal rate $\lambda^\circ \in \Lambda$, we use the *Hamiltonian* H definition [29].

$$H(\pi(\theta), \Psi(\theta), \lambda) = \frac{1}{2} w\,(X(\theta) - \hat{X}(\theta,\lambda))^2 + (\pi(\theta)\,Q)'.\Psi(\theta) \qquad (3.8)$$

Where, the variable ψ satisfies the following (backward) differential equation.

$$\frac{d\Psi}{dt} = \frac{\partial H}{\partial \pi} = Q'.\Psi(\theta) - w.(X(\theta) - \hat{X}(\theta,\lambda)) . \frac{\partial \hat{X}(\theta,\lambda)}{\partial \pi}$$

$$\Psi(T_f) = 0 \qquad (3.9)$$

Where " ' " denotes the transpose.

Based on the above definitions, we introduce the following algorithm with the help of which the optimal parameter λ° can be determined·

Algorithm: Determine the optimal parameter λ°·

Step 1. Set l =1, guess $\lambda^{(l)} \in \Lambda$, and obtain the elements $q_{ij}^{(l)}$ of the matrix Q using equation (3.3).

Step 2. Solve the differential equation 3.2, and get $\pi^{(l)}(\theta) = \pi^{(l)}(\theta, \lambda^{(l)})$; $\theta \in [0,T_f]$.

Step 3. Use $\pi^{(l)}(\theta)$ to solve equation 3.9, and obtain $\Psi^{(l)}(\theta) = \Psi^{(l)}(\theta, \lambda^{(l)})$.

Step 4. Update $\lambda^{(l)}$ for all members of the rate vector Λ using the following relation.

$$\lambda^{(l+1)} = \lambda^{(l)} + \varepsilon_l\, g^{(l)} \qquad (3.10)$$

Where ε_l >0 is chosen sufficiently small, and

$$g \equiv \frac{\partial H(\pi, \Psi, \lambda)}{\partial \lambda} \tag{3.11}$$

Step 5. If the stopping criterion is satisfied.
(A suitable criterion can be taken as $|J(\lambda^{(l+1)}) - J(\lambda^{(l)})| < \delta$,
where δ (>0) is sufficiently small)
Then stop
Else Go To Step 2.

Remark 3.1. The convergence of the algorithm depends on the choice of ε_l. Computing the performance index J for different ε_l's and interpolating using a cubic polynomial to obtain the minimum J and the corresponding ε_l gives faster convergence.

Remark 3.2. In step 4, only the updating of the transition rates of the AN-net, the AMI-net, the MAI-net, and the CN-net is required, while the transitions rates of the MN-net are kept unchanged.

4. Application and Numerical Simulations

In this section, we illustrate the mechanisms of applying the proposed modeling approach on real systems such as the token ring-based system. As shown in Fig. 2.a, the token ring system consists of three modules: a tagged client module, a super client subsystem and a server subsystem [32]. The complete SRN model for this system is given in Fig. 2.b. In the following, we study how to use the SRN modeling technique to solve the problem of obtaining the performance measures of the tagged client module.

For the solution of this problem, one may use the conventional technique that requires the construction of the SRN models for the other two modules (super client subsystem and server subsystem.). The other convenient way is to use our proposed approach that requires the construction of the AN-net with the logical interface nets. It is clear that the proposed technique becomes more effective as the number of system modules increases. We solve this problem using both the conventional SRN technique as well as our proposed modeling technique. Throughout this solution, we use the notation given in [32].

As shown in Fig. 2.b, the SRN model of the tagged client consists of four places: p_{TI}, p_{TA}, p_{TS} and p_{SP}, and four transitions: t_{ta}, t_{ts}, s_1 and t_{sp}. We assume the time until a client generates a request is an exponentially distributed random variable with mean $1/\lambda_c$, which corresponds to the mean firing time of t_{ta}. We assume the request transmission time is an exponentially distributed random variable with mean $1/\mu$, which to the mean firing time of t_{ts}. We assume the walk time, or the polling time (i.e. the time for the network token to move from one station to the next) is an exponentially distributed random variable with mean $1/\gamma$, which corresponds to the mean firing time of t_{sp}.

The place p_{TI} contains one PN token and represents the condition that the client is idle (in the sense, the client does not generate a network-based request). The place p_{TA} represents the condition that the client's request is waiting for the network token to arrive. The place p_{TS} represents the condition that the network token has arrived. Suppose that the place p_{TS} contains a PN token. If there is a PN token in p_{TA}, then the timed transition t_{ts} (whose mean time to fire is the mean request transmission time) is enabled; otherwise the immediate transition s_1 fires . The place p_{SP} represents one of the following conditions: (i) the client has finished transmitting its request, if this request reaches through t_{ts}, or (ii) the client has no request to transmit, if this request reaches via s_1.

To obtain any performance measure, we must know the values of the transition rates of the developed model. Therefore, we assume the following parameters in order to analyze our developed model. The network has a channel capacity of 10 Mbps and a signal propagation velocity of 2×10^8. The cable length is 400 km. The request packet length is an exponentially distributed random variable with mean 10^4 bits/packet (which corresponds to $1/\mu$ =1ms). The reply packet length is an exponentially distributed random variable with mean 1Kbytes/packet (which corresponds to $1/\beta = 0.819$ ms). The mean token length is three bytes (which corresponds to a mean walk time of $1/\gamma = \{(2/(N_c+1)) + 0.0024\}$ ms, where N_c is the number of client workstations). In our analysis, we assume N_c equals five. The mean time taken by the server to produce a reply is $1/\eta = 2$ ms. Finally, the time until a client generates a request is $1/\lambda_c = 11.376$ ms.

The reader may notice that some parameter values (e.g., the cable length, and the request packet length) may be rude. Yet, we have chosen these values to avoid stiffness [9, 10, 18] of the matrix Q, which may cause slow convergence of the proposed algorithm of Section 3. However, we have maintained all the relations between the parameters as given in [32]. To use practical values of the parameters as those presented in [32], Explicit Runge Kutta (which we have used in steps 2 and 3 of the algorithm of Section 3) can be replaced by Implicit Runge Kutta [31].

Using the tagged client module of the (complete) model shown in Fig. 2.b, we can determine the following performance measures. The first measure is the probability that the tagged client does some activity. This measure is denoted by $\tilde{X}_1(\theta)$ and defined as follows.

$$\tilde{X}_1(\theta) = \sum_{i:S_i \in S} \rho_i(\lambda,\theta)\,\pi_i(\theta) \;\; ; \forall S_i \in S : \rho_i(\lambda,\theta) = 1$$

$$S = \{S_i \in RS: ((\#(p_{TI}) \neq 0) \vee (\#(p_{TA}) \neq 0) \vee (\#(p_{TS}) \neq 0) \vee (\#(p_{SP}) \neq 0)) = true\} \qquad (4.1)$$

Where "#(p)": denotes the number of tokens in a place p.

The second measure is the mean response time at a client workstation. This measure is denoted by $\tilde{X}_2(\theta)$ and defined as follows.

$$\tilde{X}_2(\theta) \equiv {(1-\alpha)}/{\lambda_c \alpha}$$

$$\alpha \equiv \sum_{i:S_i \in S} \rho_i(\lambda,\theta)\,\pi_i(\theta);$$

$$\forall S_i \in S : \rho_i(\lambda,\theta) = 1$$

$$S = \{S_i \in RS: (\#P_{TI} \neq 0) = true\} \qquad (4.2)$$

It must be noted that the mean response time at a client workstation is the mean time that elapses from the instant a request is generated until the reply to the request is received.

The third of these measures is the rate of request transmission by a client workstation. This measure is denoted by $\tilde{X}_3(\theta)$ and defined as follows.

$$\tilde{X}_3(\theta) = \sum_{i:S_i \in S} \rho_i(\lambda,\theta)\,\pi_i(\theta);$$

$$\forall S_i \in S : \rho_i(\lambda,\theta) = \mu$$

$$S = \{S_i \in RS: (\#P_{TA} \neq 0) = true\} \qquad (4.3)$$

The fourth of these measures is the percentage of successful request transmissions by a client workstation. This measure is denoted by $\tilde{X}_4(\theta)$ and defined as follows.

$$\tilde{X}_4(\theta) = 100 \times \left(\sum_{i:S_i \in S} \rho_i(\lambda,\theta)\, \pi_i(\theta) \right);$$
$$\forall S_i \in S : \rho_i(\lambda,\theta) = 1$$
$$S = \{S_i \in RS: (\#P_{SP} \neq 0) = true\} \qquad (4.4)$$

The fifth of these measures is the probability of receiving a reply from the server. This measure is denoted by $\tilde{X}_5(\theta)$, and defined as follows.

$$\tilde{X}_5(\theta) = \left(\sum_{i:S_i \in S} \rho_i(\lambda,\theta)\, \pi_i(\theta) \right);$$
$$\forall S_i \in S : \rho_i(\lambda,\theta) = 1$$
$$S = \{S_i \in RS: (\#P_{TS} \neq 0) = true\} \qquad (4.5)$$

Using the tagged client module of the (complete) model shown in Fig. 2.b, the numerical data given above and the SPNP package [33], we have obtained the performance measures $\tilde{X}_i(\theta); 2 \le i \le 5$ as a function of θ. These measures are shown in Figs. 4-8. We shall now use our methodology and show with the help of the proposed algorithm how one can obtain similar results as those shown in Figs. 4-8.

In order to use our proposed model for representing the token ring system, we assume the following. The size of the matrix Q is of large dimensions; hence, it is difficult to use the complete model of Fig. 2.b. The behavior of the tagged client module is known so that the corresponding SRN model can be built. This SRN model will stand for our MN-net. The probability that the tagged client does some activity (denoted by $\tilde{X}_1(\theta)$) is given. Figure 4 illustrates this measure. The other performance measures $\tilde{X}_i(\theta); 2 \le i \le 5$ are rather difficult to measure (by measurement or system designer experience). Under these assumptions, we use the EM-model and the corresponding algorithm to illustrate how one can obtain good estimates for $\hat{X}_i(\theta); 2 \le i \le 5$ of the unknown performance measures $\tilde{X}_i(\theta); 2 \le i \le 5$, given only the performance measure $\tilde{X}_1(\theta)$. Equations 4.1- 4.5 illustrate the performance measures $\hat{X}_i(\theta); 1 \le i \le 5; \theta \ge 0$.

In the following, we explain how our EM-model can be constructed for the *token ring system*.

Since the SRN model of the tagged client module represents the MN-net, the places: $p_{MN,1}$, $p_{MN,2}$, $p_{MN,3}$, and $p_{MN,4}$ can replace the places: P_{TI}, P_{TA}, P_{TS}, and P_{SP} respectively. Similarly, the transitions: $t_{MN,1}$, $t_{MN,2}$, $t_{MN,3}$, and $t_{MN,4}$ can replace the transitions: t_{ta} , t_{ts}, s_1, and t_{sp} respectively. From the SRN model of the MN-net, we can easily conclude that $OT_{MN} = \{t_{MN,2}, t_{MN,3}\}$, $IP_{MN} = \{p_{MN,1}, p_{MN,2}\}$, $OP_{MN} = \{\phi\}$, and $IT_{MN} = \{\phi\}$, where ϕ is the empty set. In other words, the input of the MN-net is from the places $p_{MN,1}$ and $p_{MN,2}$ and its output is taken from the transitions $t_{MN,2}$ and $t_{MN,4}$.

The construction of the AN-net is illustrated in Fig. 3.b, where tokens are allowed to transfer between $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$, back and forth through the transitions $t_{AN,1}, \ldots, t_{AN,6}$. The input of AN-net is from the MAI-net transitions (T_{MAI}), and its output is given to the AMI-net transitions (T_{AMI}). The main function of AN-net is to continuously supply the MN-net with tokens through the AMI-net and sink its token through the MAI-net.

In the following algorithm, we illustrate how the AN-net order (k_{AN}) can be determined. Assume that the performance measure $\tilde{X}_1(\theta)$ is known, and we need to predict the performance measures $\hat{X}_i(\theta); 2 \leq i \leq 5$. By building our equivalent model with an AN-net of $k_{AN} = j$, we can estimate $\hat{X}_i(\theta); 2 \leq i \leq 5$ as $\hat{X}_i^j(\theta); 2 \leq i \leq 5$.

Algorithm: Determine the AN-net order (k_{AN})

Step 1: Let the AN-net order be $k_{AN} = j$, and initially set j=3.

Step 2: Build the equivalent model as illustrated in Section 2 as follows. Apply the proposed technique (explained in Section 3) to the proposed Equivalent Model (EM-model) of the desired system (explained in Section 2) to calculate the transition firing rates of its nets: the MN-net, the AN-net, the MAI-net and the CN-net.

Step 3: From the available system performance measure $\tilde{X}_1(\theta)$, we estimate (based on the algorithm described in Section 3) the other required performance measures of the MN-net ($\hat{X}_i^j(\theta); 2 \leq i \leq 5$). Equations 4.1 - 4.5 illustrate some of these measures.

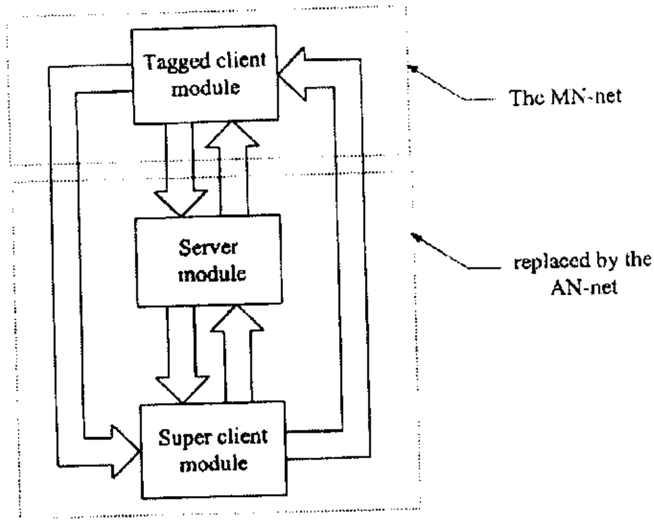

Fig. 2. a. Block diagram of the client server system.

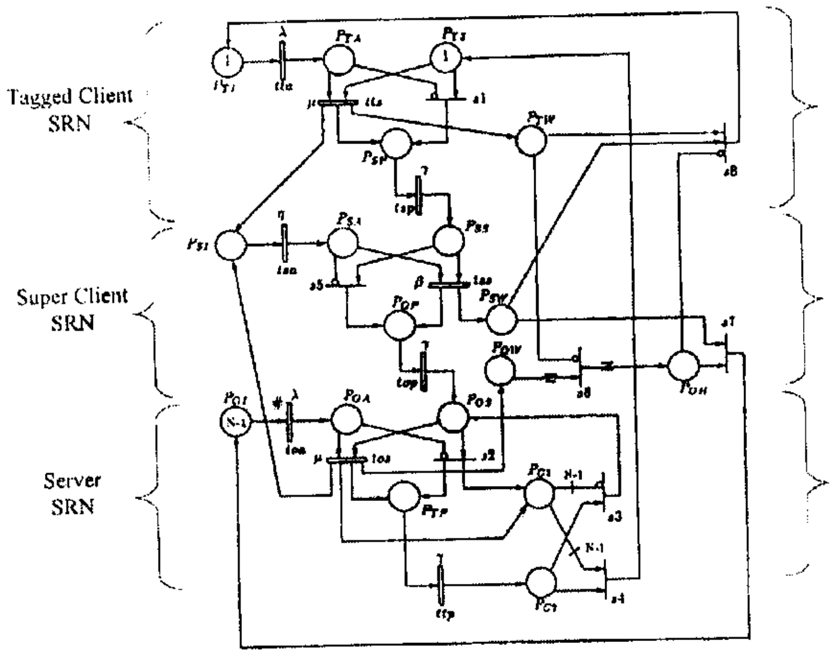

Fig. 2. b. SRN for the complete model of the token ring network (N>1).

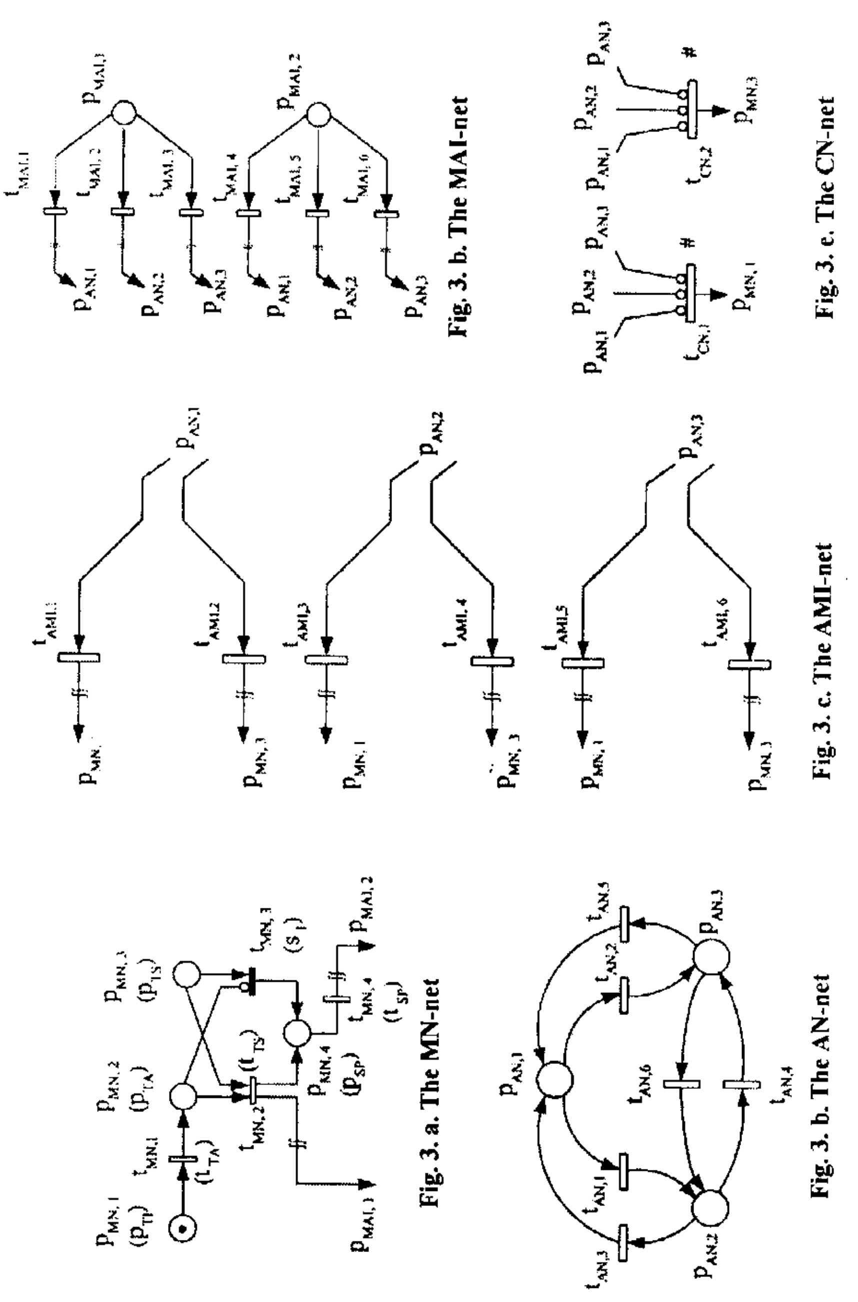

Fig. 3. a. The MN-net

Fig. 3. b. The AN-net

Fig. 3. c. The AMI-net

Fig. 3. b. The MAI-net

Fig. 3. e. The CN-net

Fig. 3. The equivalent model (EM) of the token ring network.

Step 4: J=J+1.
Step 5: Use steps 2 and 3 to calculate the required performance measures $\hat{X}_i^{j+1}(\theta); 2 \le i \le 5$.

Step 6: Compare between the values of performance measures for the two successive values of k_{AN}. Calculate the expected percentage precision $\% e$ as follows.

$$\% e \equiv \underset{i:2\to 5}{Max}\left(\sum_{k=0}^{n} \left| \frac{(\hat{X}_i^{j+1}(\theta_k) - \hat{X}_i^{j}(\theta_k))}{\hat{X}_i^{j}(\theta_k)} \right| \times 100 \right) \tag{4.6}$$

Step 7: According to the required accuracy, stop the calculation. Verify from the minimum expected percentage precision $\% \delta$ as follows.

If $\% e < \% \delta$
Then Go To Step 8
Else Go To Step 4

Step 8: End

By running this algorithm on our example, we have found that taking $k_{AN} = 3$ gives us $\%\delta = 5$ %. Higher orders of k_{AN} give us less percentage precision. Therefore, in our analysis, we have chosen $k_{AN} = 3$ to represent the AN-net order.

The SRN AMI-net is shown in Fig. 3.c. The input of the AMI-net is from the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$, and its output is given to IP_{MN} of the MN-net: $p_{MN,1}$, and $p_{MN,3}$. The AMI-net controls the flow of tokens from the AN-net to the MN-net in an organized form preventing formation of deadlocks in the MN-net. In other words, the AMI-net supplies the MN-net with tokens. This prevention can be done as follows. Control the flow of tokens from the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$ to $p_{MN,1}$ through the AMI-net transitions: $t_{AMI,1}$, $t_{AMI,3}$, and $t_{AMI,5}$ respectively, and from the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$ to the MN-net place $p_{MN,3}$ through the AMI-net transitions: $t_{AMI,2}$, $t_{AMI,4}$, and $t_{AMI,6}$ respectively.

From the original system, it is clear that the MN-net places: $p_{MN,2}$, and $p_{MN,3}$ must not contain a token when the place $p_{MN,4}$ has one. Similarly, the MN-net place $p_{MN,1}$ must be free of tokens when the place $p_{MN,2}$ has one. Consequently, arcs connecting the AMI-net transitions: $t_{AMI,1}$, $t_{AMI,3}$, and $t_{AMI,5}$ to the MN-net place $p_{MN,1}$ and arcs connecting the

AMI-net transitions: $t_{AMI,2}$, $t_{AMI,4}$, and $t_{AMI,6}$ to the place $p_{MN,3}$ must be associated with rules governing the flow of tokens according to the system original behavior. These constraints are translated as arc weight functions for the arcs connecting the AMI-net transitions: $t_{AMI,1}$, $t_{AMI,3}$, and $t_{AMI,5}$ to the place $p_{MN,1}$, and arcs connecting the AMI-net transitions: $t_{AMI,2}$, $t_{AMI,4}$, and $t_{AMI,6}$ to the place $p_{MN,3}$. Table 1 shows these functions.

The SRN MAI-net is shown in Fig. 3.d, where "≈" implies that the arc weight is a variable governed by a function. The input of the MAI-net is from OT_{MN} of the MN-net: $t_{MN,2}$, and $t_{MN,3}$, and its output is given to IP_{AN} of the AN-net: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$. The main function of MAI-net is to permit feedback from the MN-net to the rest of the model. It also allows the flow of tokens out of the MN-net. Another benefit of the MAI-net is to control the flow of tokens from the MN-net transitions: $t_{MN,2}$, and $t_{MN,3}$ to the AN-net, so as to avoid unboundness of PN tokens in the AN-net. This controlling action is done using the arcs for connecting the following. Each of the MN-net transitions: $t_{MN,2}$, and $t_{MN,3}$ to the places: $p_{MAI,1}$, and $p_{MAI,2}$ respectively. Each of the MAI-net transitions: $t_{MAI,1}$, $t_{MAI,2}$, and $t_{MAI,3}$ to the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$ respectively. Finally, each of the MAI-net transitions: $t_{MAI,4}$, $t_{MAI,5}$, and $t_{MAI,6}$ to the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$ respectively. These functions are also indicated in Table 1.

To avoid the effect of token absence in the AN-net, we have used the CN-net. As shown in Fig. 3.e, the CN-net consists of the transitions: $t_{CN,1}$ and $t_{CN,2}$, either of them is connected to the AN-net places: $p_{AN,1}$, $p_{AN,2}$ and $p_{AN,3}$ through inhibitor arcs. The input of the CN-net is from the AN-net places and its output is given to IP_{MN} of the MN-net: $p_{MN,1}$, and $p_{MN,3}$. In the absence of tokens in the AN-net places: $p_{AN,1}$, $p_{AN,2}$, and $p_{AN,3}$, the transition $t_{CN,1}$ or $t_{CN,2}$ supplies the MN-net places: $p_{MN,1}$, and $p_{MN,3}$ with tokens. This action prevents the generation of deadlock during the evolution of the EM-model. Since the CN-net may supply the MN-net places: $p_{MN,1}$and $p_{MN,3}$ with tokens against their logical functionality, the CN-net transitions: $t_{CN,1}$ and $t_{CN,2}$ are to be controlled with an enabling function as shown in Table 2.

Having finished the construction of the SRN EM-model, the next step is to use the proposed algorithm of Section 3 to determine the unknown transition rates so that the estimated performance measure $\hat{X}_1(\theta)$ is close enough to the given performance measure $\hat{X}_1(\theta)$ for all $\theta \in [0, T_f]$. Here, the performance measure $\hat{X}_1(\theta)$ can be determined from equation 4.1 with the set S being replaced as follows.

Table 1. Arc weights for the variable arcs of the equivalent model of the token ring based system

Arc location	Arc	Variable arc weight function (arc multiplicity)
From MN-net to MAI	${}^{t}MN.2 \rightarrow {}^{P}MAI,1$	*arc_weight=1* If (#^{P}MAI,1==1) *arc_weight=0*
	${}^{t}MN.4 \rightarrow {}^{P}MAI,2$	*arc_weight=1* If (#^{P}MAI,2==1) *arc_weight=0*
From MAI to AN-net	${}^{t}MAI,1 \rightarrow {}^{P}AN,1$ ${}^{t}MAI,2 \rightarrow {}^{P}AN,2$ ${}^{t}MAI,3 \rightarrow {}^{P}AN,3$	*arc_weight=0* If (#^{P}MAI,1>=1) If ((#^{P}AN,1==0) && ((#^{P}AN,2==0)&& (#^{P}AN,3==0)) *arc_weight=1*
	${}^{t}MAI,4 \rightarrow {}^{P}AN,1$ ${}^{t}MAI,5 \rightarrow {}^{P}AN,2$ ${}^{t}MAI,6 \rightarrow {}^{P}AN,3$	*arc_weight=0* If (#^{P}MAI,2>=1) If ((#^{P}AN,1==0) && ((#^{P}AN,2==0)&& (#^{P}AN,3==0)) *arc_weight=1*
From AMI to AN-net	${}^{t}AMI,1 \rightarrow {}^{P}MN,1$ ${}^{t}AMI,3 \rightarrow {}^{P}MN,1$ ${}^{t}AMI,5 \rightarrow {}^{P}MN,1$	*arc_weight=0* If ((#^{P}MN,1==0) && ((#^{P}MN,2==0) *arc_weight=1*
	${}^{t}AMI,2 \rightarrow {}^{P}MN,3$ ${}^{t}AMI,4 \rightarrow {}^{P}MN,3$ ${}^{t}AMI,6 \rightarrow {}^{P}MN,3$	*arc_weight=0* If ((#^{P}MN,3==0) && ((#^{P}MN,4==0) *arc_weight=1*

Table 2. Transition enabling functions for the equivalent model of the token ring based system

Transition location	Transition	Enabling function
CN-net	${}^{t}CN,1$	*enable=false* If ((#^{P}MN,1==0)&&(#^{P}MN,2==0)) *enable=true*
	${}^{t}CN,2$	*enable=false* If ((#^{P}MN,3==0)&&(#^{P}MN,4==0)) *enable=true*

$$S = \{S_i \in RS: (\# p_{MN,1} \vee \# p_{MN,2} \vee \# p_{MN,3} \vee \# p_{MN,4}) = true\} \quad (4.7)$$

Figure 4 shows how $\hat{X}_1(\theta)$ approaches $\tilde{X}_1(\theta)$ after 30 iterations. Throughout our calculations, we have taken $T_f = 100$ ms with a step of 0.1 ms. As we have explained, the MN-net stands for the tagged client module. Therefore, during the identification process, the MN-net transition rates are kept constant, while the transition rates of the

AN-net, the AMI-net, the MAI-net and the CN-net are changed continuously. Subsequently, the rates of the AMI transitions: $t_{AMI,1}, \ldots, t_{AMI,6}$, and the MAI transitions: $t_{MAI,1}, \ldots, t_{MAI,6}$ changed to a great extent, while those of the AN-net: $t_{AN,1}, \ldots, t_{AN,6}$ changed slightly. This result is sensible because the main task of the AN-net is to generate tokens, while the AMI-net and the MAI-net are responsible for controlling the flow into, and out of, the MN-net. Table 3 shows the values of the transition rates of the equivalent model before and after identification, where values before identification are chosen, by trial and error, to give a satisfactory convergence between $\hat{X}_1(\theta)$ and $\tilde{X}_1(\theta)$. This follows from the fact that the algorithm of Section 3.1 performs a local optimization method [29].

After identification, the other performance measures $\hat{X}_j(\theta)$; $2 \le j \le 5$ of the client-tagged module can be determined from the EM-model. Figures 4, 5, 6 and 7 show the estimated performance measures $\hat{X}_j(\theta)$; $2 \le j \le 5$ from the proposed EM-model as functions of time. We have also taken T_f =100 ms with a step of 0.1 ms. As shown in these figures, the estimated performance measures $\hat{X}_j(\theta)$; $2 \le j \le 5$ are close to those ($\tilde{X}_j(\theta)$; $2 \le j \le 5$) of the complete model shown in Fig. 2.b. To validate the closeness between these measures, we have defined the average percentage difference (denoted by $\%D_j$) between the performance measure $\hat{X}_j(\theta)$ obtained from the EM-model and the corresponding performance measure $\tilde{X}_j(\theta)$ obtained from the complete model shown in Fig. 2 as follows.

$$\%D_j \equiv \frac{1}{n} \sum_{i=0}^{n} \left| \frac{(\hat{X}_j(\theta_i) - \tilde{X}_j(\theta_i))}{\tilde{X}_j(\theta_i)} \right| \times 100$$

$$\theta_i = i\,\Delta\theta \quad , \quad n = {}^{T_f}\!\big/_{\Delta\theta} \tag{4.8}$$

Where $\Delta\theta$ is the unit time step of identification. As shown in Table 4, the difference ($\%D_j$; $1 \le j \le 5$) between the measures obtained from the EM-model and those of the complete model (Fig. 2b) is approximately 10 %. In other words, the percentage of closeness between both performance measures is approximately 90%.

One of the important merits of the proposed methodology is the great reduction in the number of states between the complete and equivalent models. The RS of the complete model contains 1880 states but the RS of the equivalent model contains only 101 states. This state reduction leads to reduce the dimensions of the matrix Q. Consequently, we can reduce the required storage memory. Although the matrix Q of the complete model contains 4400 non-zero elements, the matrix Q of the equivalent model contains only 679 non-zero elements.

Although we have identified a client server system consisting of five stations, the same equivalent model can be used to identify a client server system consisting of greater number of stations, which in turn implies a greater number of RS states. Then higher orders of the AN-net may be used to reach a desired accuracy. Further, we draw the attention of the reader to the fact that the proposed methodology can be applied to fault-tolerant systems to obtain their transient measures such as availability, reliability, and etc. [5, 18, 23, 31]. Unfortunately, we have not considered these kinds of system due to space limitations.

Table 3. The transition rates of the equivalent model before and after identification using the algorithm of section (3.1)

Transition	Trans. rate before ident.	Trans. rate after ident.	Transition	Trans. rate before ident.	Trans. rate after ident.
$t_{MN,1}(\lambda_c)$	87.95	87.95	$t_{AMI,3}$	200	126.60848
$t_{MN,2}(\mu)$	1000	1000	$t_{AMI,4}$	200	280.96172
$t_{MN,3}(\gamma)$	2000	2000	$t_{AMI,5}$	200	126.60848
$t_{MN,4}(s_1)$	1(weight)	1(weight)	$t_{AMI,6}$	200	280.96172
$t_{AN,1}$	200	200	$t_{MAI,1}$	200	193.00081
$t_{AN,2}$	200	200	$t_{MAI,2}$	200	193.00081
$t_{AN,3}$	200	200	$t_{MAI,3}$	200	193.00081
$t_{AN,4}$	200	200	$t_{MAI,4}$	200	203.79527
$t_{AN,5}$	200	200	$t_{MAI,5}$	200	203.79527
$t_{AN,6}$	200	200	$t_{MAI,6}$	200	203.79527
$t_{AMI,1}$	200	126.60848	$t_{CN,1}$	200	111.43852×10^{-6}
$t_{AMI,2}$	200	280.96172	$t_{CN,2}$	200	403.87229

Table 4. The average percentage difference ($\%Dj$; $1 \le j \le 5$) between $\tilde{x}_i(\theta), 1 \le i \le 5$ and $\hat{x}_i(\theta), 1 \le i \le 5$.

$\%D_1$	0.938%
$\%D_2$	8.049%
$\%D_3$	4.953%
$\%D_4$	4.816%
$\%D_5$	9.673%

5. Conclusion

In this paper, we have developed a new approach for solving the modeling problem of discrete event systems. The proposed model consists mainly of two nets. The first main net is denoted by the MN-net. The MN-net describes the basic structure of the interesting part(s) of the DEDS that allows us to model its dynamic behavior and evaluate its performance measures. The second main net acts as an auxiliary net (denoted by AN-net). The AN-net feeds the MN-net by the required tokens. To allow the interaction between the MN-net and the AN-net, we have introduced three interface nets to the overall model. The first interface net (denoted by MAI-net) works as an interface between the MN-net and the AN-net. The second interface net (denoted by AMI-net) works as an interface between the AN-net and the MN-net. The MAI-net controls token flow between the MN-net and the AN-net, whereas the AMI-net controls token flow between the AN-net and the MN-net. The third interface net (denoted by CN-net) has been introduced to protect the overall model against deadlocks. An SRN model has represented the AN-net with an arbitrary number of places. This number defines the order of the AN-net. The choice of the AN-net order depends mainly on how accurate the overall model behaves with respect to the original DEDS model.

To efficiently use the proposed modeling methodology, the following parameters must be known.

(i) At least one performance measure for the interest part(s) or the known part(s) of the overall system.
(ii) The complete knowledge of the basic structure of this system part(s) to allow building its MN-net in an easy way.

In order to illustrate the effectiveness of the proposed approach, we have investigated the modeling problem of the token ring system. This problem has been tackled using two different approaches, and their performance measures have been compared. In the first approach, we have used the conventional SRN technique to build a complete model for the token ring system to determine the performance measures

$\tilde{X}_i(\theta); 1 \le i \le 5$. In the second approach, we have used our proposed modeling technique to build a model for the token ring system under the following assumptions. The tagged client module behavior is known, the performance measure $\tilde{X}_1(\theta)$ is given, and the performance measures $\tilde{X}_i(\theta); 2 \le i \le 5$ are unknown or difficult to determine.

Since some of the transition rates of the proposed model are unknown, we have used the algorithm proposed in Section 3 to determine these unknown rates. Therefore, the estimated performance measure $\hat{X}_1(\theta)$ obtained is close enough to the performance measure $\tilde{X}_1(\theta)$. Fig. 4 shows that the estimated performance measure $\hat{X}_1(\theta)$ converges to the performance measure $\tilde{X}_1(\theta)$ in about 30 iterations. Thus, the computed transition rates have been used to determine the estimated measures $\hat{X}_i(\theta); 2 \le i \le 5$. Figs. 5, 6, 7 and 8, show that the measures obtained via our proposed methodology are fairly close to those computed using the conventional SRN complete model. From these results, we conclude that the proposed modeling technique is much easier to handle due to the following facts.

1. The use of the proposed model reduces to a great extent the number of RS states. The RS of the SRN complete model contains 1880 states whereas those of the proposed model are only 101 states.
2. The size of the matrix Q for the proposed model is very much smaller than that of the SRN complete model. The Q size reduction leads to minimize the storage memory required.

Although we have identified a client server system consisting of only three modules, the same approach can be used for identifying systems consisting of a larger number of modules (i.e. a larger number of states). To efficiently solve the systems that have a large number of states, we may increase the AN-net order. This solution leads to the desired accuracy.

It is important to note that the application of the proposed modeling technique does not depend on whether the desired system has homogenous or heterogeneous behavior but on the interaction between the proposed nets that constitute the modeled systems. Such nets are the AN-net and the MN-net. For more explanation, we assume that we need to model five - heterogeneous clients. If we know the dynamic behavior of

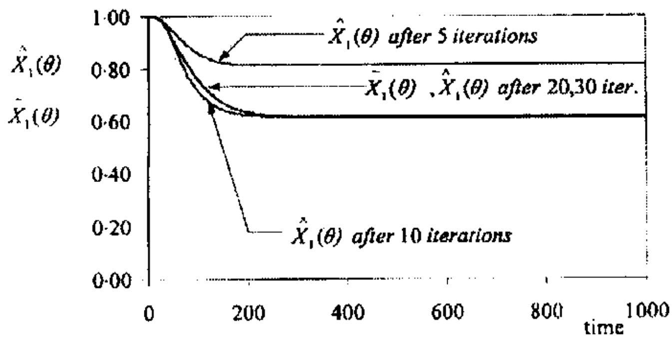

Fig. 4. Probability that the tagged client does some activity (each time unit = 0.1 ms).

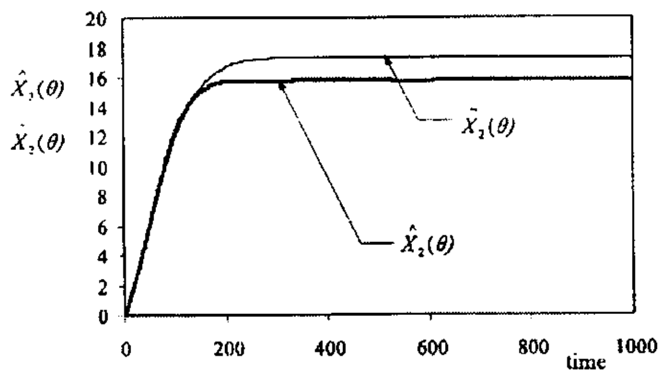

Fig. 5. Mean response time at a client WS. (ms) (each time unit = 0.1 ms).

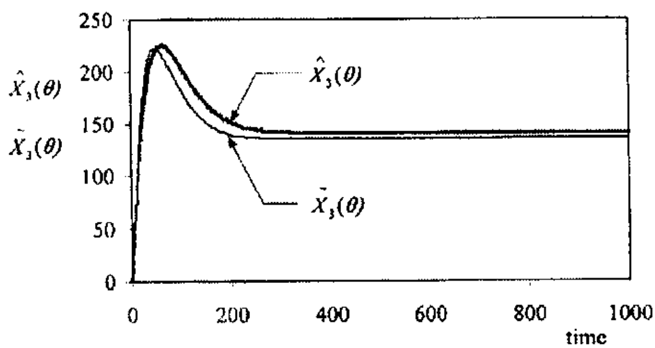

Fig. 6. Rate of request transmission (each time unit = 0.1 ms).

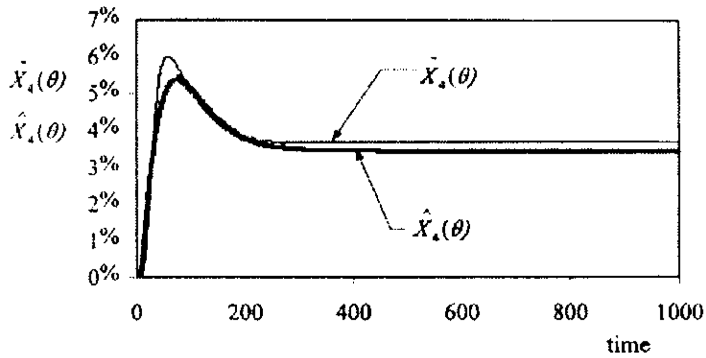

Fig. 7. Percentage of successful request transmission (each time unit = 0.1 ms).

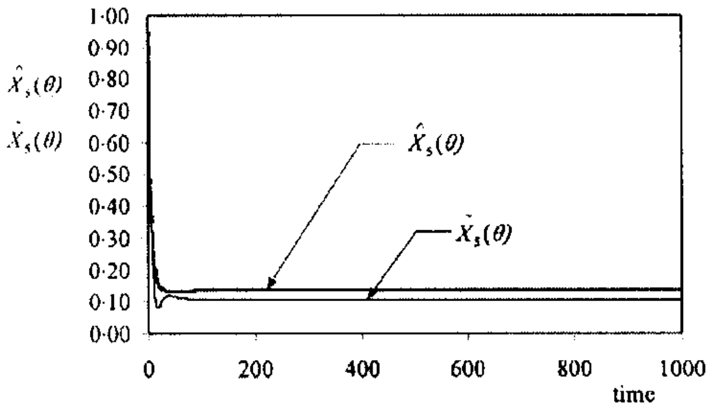

Fig. 8. Probability of receiving a reply (each time unit = 0.1 ms).

only two clients among them, then (i) we designate the MN-net to represent these two clients and, (ii) the AN-net to represent the rest of clients. Subsequently, we construct the interface nets between them by the MAI-net and the AMI-net. These steps can be also applied on homogeneous systems as we have illustrated in our research work. In other words, these modeling steps do not depend on whether the system has heterogeneous or homogenous behavior.

References

[1] Capkovic, F. "A Solution of DEDS Control Synthesis Problems." *Proceedings of the IFAC Conference on Control System Design*, Bratislava, Slovakia, (2000), 337-342.

[2] Capkovic, F. and Capkovic, P. "A New Tool for Knowledge-based Control Synthesis of DEDS by Means of Simulation." *Proceedings of the 2001 European Simulation Multi-Conference: Modeling and Simulation*, Prague, Czech Republic, (2001), 179-186.

[3] Capkovic, F. "An Approach to Model and Control of Discrete-event Dynamic Systems." *Proceedings of the European Control Conference* (ECC'2001), Porto, Portugal, (2001), 1007-1012.

[4] Buchholz, P. "An Adaptive Importance Sampling Approach for the Transient Analysis of Markovian Queuing Networks." *European Transactions on Telecommunications* 13, No. 4 (2002), 317-330.

[5] Bradley, J. T., Knottenbelt, W. J., Harrison, P. G. and Vowden, C. J. "Transient and Passage-time Distributions in Semi-markov Processes." *Proceedings of the 18th UK Performance Engineering Workshop (UKPEW2002)*, Glasgow, UK, (2002), 153-162.

[6] Ciardo, G., Blakemore, A., Chimento, P. F. J., Muppala, J. K. and Trivedi, K. S. "Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets." In: Meyer, C. and Plemmons, R. J. (Ed.) *Linear Algebra, Markov Chains, and Queuing Models*, Volume 48 of IMA Volumes in Mathematics and its Applications, Springer-Verlag, (1993), 145-191.

[7] Ramani, S., Trivedi, K.S. and Dasarathy, B. "Performance Analysis of the CORBA Event Service Using Stochastic Reward Nets." *Proceedings of the 19th Symposium on Reliable Distributed Systems*, Nurnberg, Germany, (2000), 238-247.

[8] Nicole, D. M. "Discrete-event System Simulation in Performance Evaluation." *Lecture Notes in Computer Science*, 1769, Springer-Verlag, (2000), 443-457.

[9] Davies, I., Knottenbelt, W.J. and Kritzinger, P.S. "Symbolic Methods for the State Space Exploration of GSPN Models." *Proceedings of the 12th International Conference on Modeling Techniques and Tools* (TOOLS 2002), London, UK, (2002), 188-199.

[10] Miner, A.S. "Efficient State Space Generation of Gspns Using Decision Diagrams." *Proceedings 2002 of the International Conference on Dependable Systems and Networks* (DSN 2002), Washington, DC, USA, (2002), 637-646.

[11] Ballarini, P., Donatelli, S. and Franceschinis, G. "Prametric Stochastic Well-Formed Nets and Compositional Modeling." *Proceedings of the 21st International Conference on Application and Theory of Petri Nets*, Aarhus, Denmark, (2000), 215-221.

[12] Koriem, S.M. "R-nets for the Performance Evaluation of Hard Real-time Systems." *Journal of Systems and Software*, 46, No.1 (1999), 41-58.

[13] Koriem, S.M. "Fast and Simple Decomposition Techniques for the Reliability Analysis of Interconnection Networks." *Journal of Systems and Software*, 45, No.2 (1999), 155-171.

[14] Schneeweiss, W. G. "Tutorial: Petri Nets as a Graphical Description Medium for Many Reliability Scenarios. *IEEE Transactions on Reliability*, 50, No.2 (2001), 159-64.

[15] Vittorini, V., Franceschinis, G., Gribaudo, M., Iacono, M. and Mazzocca, N. "DrawNet++: Model Objects to Support Performance Analysis and Simulation of Complex Systems." *Proceedings of the 12th International Conference on Modeling Tools and Techniques for Computer and Communication System Performance Evalution* (TOOLS 2002), London, UK, (2002), 335-348.

[16] Park, J., Reveliotis, S., Bodner, D. and McGinnis, L.F. "A Distributed Event Control Architecture for Flexibly Automated Manufacturing Systems." *International Journal of Computer Integrated Manufacturing*, 15, No.2 (2002), 109-126.

[17] Hadjicostis, C.N. and Verghese, G.C. "Encoded Dynamics for Fault-tolerance in Linear Finite-state Machines." *IEEE Transactions on Automatic Control*, 47, No. 1 (2002), 189-192.

[18] Goseva-popstojanova, K. and Trivedi, K.S. "Stochastic Modeling Formalisms for Dependability, Performance and Performability." *Lecture Notes in Computer Science*, 1769, Springer-Verlag, (2000), 403-422.

[19] Buchholz, P. "An Adaptive Aggregation/Disaggregating Algorithm for Hierarchical Markovian Models." *European Journal of Operational Research* 116, No. 3 (1999), 545-564.

[20] Capra, L., Dutheillet, C., Franceschinis, G. and Ilie, J.M. "Exploiting Partial Symmetries for Markov Chain Aggregation." *Electronic Notes in Theortical Computer Science* 39, No. 3, (2000), 287-295.

[21] Marsan, M.A., Gaeta, R. and Meo, M. "Accurate Approximate Analysis of Cell-based Switch Architectures." *Performance Evaluation*, 45, No.1 (2001), 33-56.

[22] Knottenbelt, W.J., Harrison, P.G., Mestern, M.A. and Kritzinger, P.S. "A Probabilistic Dynamic Technique for the Distributed Generation of Very Large State Spaces." *Performance Evaluation Journal* 19, No. 1-4 (2000), 217-148.

[23] Ezpeleta, J. *Petri Nets for System Engineering: A Guide to Modeling, Verification and Applications*. Springer-Verlag, (2001).

[24] Aoumeur, N. and Saake, G. "Towards an Adequate Framework for Specifying and Validating Runtime Evolving Complex Discrete-event Systems." Proceedings of the 1st *Workshop on Modeling of Objects Components, and Agents*, Aarhus Unveristy, Denmark, DAIMI PB-553, (2001), 1-20.

[25] Esser, R. and Janneck, J.W. "Moses: A Tool Suite for Visual Modeling of Discrete-event Systems." *Symposium on Visual/Multimedia Approaches to Programming and Software Engineering*, HCC'01, Stresa, Italy, (2001), 416-423.

[26] Juhas, G. "A Unified Approach to Model and Control of a Class of Discrete- event and Hybrid Systems Via Algebraically Generalized Petri Nets." *Proceedings of the IFAC Conference on Control System Design*, Bralislava, slovakia, (2000), 377-382.

[27] Yoo, T.-S. and Lafortune, S. "On the Computational Complexity of Some Problems Arising in Partially-observed Discrete-event Systems." *Proceedings of the American Control Conference, Arlington*, VA, USA, (2001), 307-312.

[28] Takai, S. "Supervisory Control of Partially Observed Discrete Event Systems with Arbitrary Control Patterns." *International Journal System Science*, 31, No.5 (2000), 649-656.

[29] Teo, K.L., Goh, C.J. and Wong, K.H. "A Unified Computational Approach to Optimal Control Problems." *Longman Scientific and Technical*, London, England, (1991).

[30] Fischer, M. and Kemper, P. "Distributed Numerical Markov Chain Analysis." *Proceedings of the 8th Euro PVM/MPI 2001*, Santorini (Thera) Island, Greece, Springer-Verlag, LNCS 2131, (2001), 272-279.

[31] Reibman, A. and Trivedi, K.S. "Numerical Transient Analysis of Markov Models." *Computers and Operations Research* 15, No.1, (1988), 19-36.

[32] Ibe, O.C., Choi, H. and Trivedi, K. S. "Performance Evaluation of Client-Server Systems." *IEEE Transactions on Parallel and Distributed Systems* 4, 11, (1993), 1217-1229.

[33] Hirel, C. , Trivedi, K.S. and Tuffin, B. *SPNP Version 6.0. Lecture Notes in Computer Science*, 1786, Springer-Verlag, (2000), 354-357.

تكنيك جـــديد للنمذجة معتمدًا على نظريات شبكات بتري لتحليل كفاءة أداء أنظمة الحاسبات ذات الحالات الحركية المميزة

ســمير م. كريم محمد، بي. أي.دبوس* و و. س. الكيلاني**

قسم هندسة النظم والحاسبات، كلية الهندسة، جامعة الأزهر، القاهرة،

**المعهد العالي للتكنولوجيا، مدينة العاشر من رمضان، القاهرة، و **قسم تقنيات المعلومات،*

كلية الحاسب والمعلومات، جامعة المنوفية، شبين الكوم، جمهورية مصر العربية

(قدّم للنشر في ١٤/١٠/٢٠٠١م؛ وقبل للنشر في ٣ /٤/٢٠٠٢م)

ملخص البحث. في جميع الدراسات والبحوث السابقة كان التفكير في عمل نموذج لإحدى نظم الحاسبات العملية الواقعية يتطلب معرفة السلوك الحركي لكل جزء من أجزاء هذا النظام. ولكن المشكلة الحقيقية تظهر إذا أردنا أن نصمم نموذجاً لهذا النظام ونحن نعرف منه (فقط) السلوك الحركي لجزء أو أكثر دون باقي أجزائه. هذه المشكلة تم دراستها رياضيا بواسطة نظريات التحكم. طبقا لبحثنا ومعرفتنا فلقد وجدنا أن هذه المشكلة لم يتم دراستها بواسطة نظريات شبكات بتري للنمذجة. ولهذا السبب فإننا في هذا البحث نقترح تكنيك جديد للنمذجة معتمدا على اتحاد نظريات شبكات بتري للنمذجة مع نظريات التحكم الرياضية. في هذا التكنيك المقترح، تم تصميم نموذج لشبكة بتري مكافأ للنظام المطلوب عمل نموذج له. هذا النموذج يتكون من شبكتين رئيستين وشبكتين مواجهتين لهما . تمثل إحدى الشبكتين الرئيستين الجزء المهم و المعلوم من النظام المطلوب عمل نموذج له. تمثل الشبكة الرئيسية الأخرى باقي أجزاء هذا النظام. لمعرفة مدى كفاءة التكنيك الجديد المقترح ، تم تطبيقه على إحدى أجهزه الحاسبات العملية الواقعية.