Journal of Computers Vol. 28, No. 2, 2017, pp. 177-187
doi:10.3966/199115592017042802013

Performability Evaluation Low-Powered Sensor Node by Stochastic Model Checking

Jun Niu, and Guang Jin

Faculty of Information Science and Engineering, Ningbo University
Ningbo, 305012, China
{niujun, jinguang}@nbu.edu.cn

Received 31 December 2015; Revised 03 June 2016; Accepted 03 July 2016

Abstract. Wireless Sensor Network (WSN) applications, there may be many low-powered sensor nodes which can communicate with each other by wireless techniques. Due to limited power supply, the satisfiability of performability properties, which include energy constraints especially, must be confirmed in design phase. This will help one to avoid implementing impracticable designs. A typical performability property may be as follows: the probability that the energy consumption of a sensor node is smaller than 5 if it operates correctly in 1 hour is bigger than 0.90. Stochastic model checking is a successful and well established technique for formally verifying performability properties of systems which exhibit stochastic behavior. This paper introduces continuous time Markov Chain (CTMC) enhanced with reward structures to model the dynamic behavior of a sensor node. The performability specification which includes energy constraints can be depicted by continuous stochastic reward logic (CSRL). Our main contribution is a verification approach of sensor nodes' performability properties, which include energy constraints especially, based on stochastic model checking technique. Moreover, we show the applicability of the approach and provide some experimental results by stochastic model checker MRMC.

Keywords: markov processes, performance evaluation, stochastic model checking, wireless sensor network

1 Introduction

Recently, Internet of Things (IoT) has engaged many attentions from many academics and industrial institutions [1-2]. It allows any physical thing to be connected to the Internet anytime, anyplace, by using any network and any service [3]. There have been many application domains of IoT such as aerospace and aviation, intelligent buildings, medical technology and healthcare, environment monitoring, people and goods transportation, agriculture and breeding, etc. [3-4].

As a networked information sensing and data gathering system, Wireless Sensor Network (WSN) is an important basic component of IoT and can be viewed as the "teleneuron" of it [2-3]. Concretely, a large number of sensor nodes in WSN serve as the bridge between physical things and Internet. In many IoT applications, there may be many small, cheap, and low-power sensor nodes which are used to monitor physical or environmental conditions and can communicate with each other via wireless technology [5]. As sensor nodes operate on limited battery power, energy supply and management is a very critical task. There have been some significant researches that focus on energy harvesting and minimizing [6] by now. Generally, the existing work mainly concentrate on energy harvesting or energy saving. The former involves that the sensor nodes replenish their energy from energy source such as solar energy, vibration energy, thermal energy, and etc, and however, the later focuses on how to save the energy when in practice by some policies in order to increase the lifetime of network given that there is no additional energy supply.

Apparently, for a majority of IoT applications, the energy supply is not boundless. In other words, it is inevitable that the usable energy cannot sustain the practical application adequately even though one can adopt some energy harvesting or energy saving techniques. Furthermore, the costs of energy harvesting or energy saving are always exorbitant relative to the cheap price of a sensor node itself. Accordingly, in design phase, it is very important that ensure the energy consumption of the considered application should fall in an acceptable interval. If not, one has to modify the design model or blueprint possibly.

Several techniques have allowed for evaluating the performability of WSN applications such as measurement, simulation, analytical and numerical methods, and etc [7-8]. All these methods can be divided into two types, i.e. test and verification. It is evident that the former method can only show the presence of errors, not their absence [9-10]. As a result, formal method is a good choice by which we should verify the performability of a wireless sensor node. Stochastic model checking is a successful formal method for checking quantitative properties of systems that exhibit stochastic behavior [11-13]. Its basic idea is to construct a random model that captures the system's dynamic behavior, and then employ it to check formally-specified properties. The strength of stochastic model checking technique bases on its exhaustiveness and completeness [11-13].

Some previous works of performability evaluation of WSN nodes are mainly focused on the time consumption. Many WSN applications are real-time, e.g., real-time embedded systems or real-time networks, and timed automata can be used to model and analyze the timing behavior of computer systems. In order to reason about the correctness of a low-power embedded system, timed automata is employed for specifying a sensor node application of low-power embedded systems, and the conformance of a real system and its expected behavior by using standard model checker Uppaal [14]. Analogously, Tschirner, Xuedong and Yi examined how to employ timed automata and Uppaal tools for validating the timing parameters of the sensor nodes such that the desired QoS requirements are satisfied [15]. Demaille, Peyronnet and Hérault employs Discrete Time Markov Chain (DTMC) modeling the sensor node behavior, and apply Approximate Probabilistic Model Checker to verify the correctness, which is written by Linear Temporal Logic (LTL) specifications [16]. Abo and Barkaouihas proposed a method for checking the performability properties of mobile WSNs, which are modeled with the stochastic Pi-calculus, by means of probabilistic model checking and PRISM [17].

However, the above works do not consider how to verify the performance and dependability properties which include the constraints of energy consumption. This paper introduces stochastic model checking technique to the performance and dependability evaluation of wireless sensor node, which are ubiquitous in IoT or WAN application areas. We employ reward structures to capture the energy consumption of a wireless sensor node when it running. A wireless sensor node can be modeled by Continuous Time Markov Chain (CTMC) which is enhanced with reward structures, i.e. Continuous Time Markov Reward Model (CMRM).

The remainder of this paper is organized as follows. Section 2 gives some description of the motivation and requirements and presents the foundations used in our modeling. Section 3 introduces the temporal logic CSRL Afterwards, in Section 4, we describe the stochastic model checking processes of a low-powered sensor node and present the verification results, and the paper is concluded in Section 5.

2 Preliminaries

2.1 Motivation and Requirements

In WSN, sensor nodes can sense, gather and process information from their environment and transmit the corresponding data to its neighbor nodes or base stations. Each sensor node, which is employed for monitoring bounded neighboring phenomena in collaboration with the other nodes, is small, and thus with limited resource constrained in terms of processing and data storage capacity, and energy especially. A typical wireless sensor node, which is battery-powered mostly, may consist of five main subsystems and can be described as follows [18-20]: A sensing subsystem includes one or more sensors which may be certain type and with associated A/D converters for data acquisition. A processing subsystem is with responsibility for local data processing, and a radio subsystem for wireless data communication. A power subsystem supply limited power and manage its power policy. Furthermore, sometimes there may be a location changing engine which can change the location around its environment if necessary.

The lifetime of battery-powered sensor networks depends on the number of active nodes and connectivity of the network [19]. Apparently, when running, if the energy supply of a sensor node depletes, it will cease to operate and discount from the wireless sensor network, and this may significantly impact the function or the performance of the whole WSN application [19]. Hence, we must ensure that the energy supply is efficient for a WSN application before it will be in practice. However, the energy supply cannot be infinite. In addition, in some domains, it is difficult to recharge or replace a sensor node's battery once deployed [19-20], or this may spend much money. In this paper, we do not focus on energy harvesting or energy conservation techniques, but the evaluation methods of performability properties, which include energy constraints especially, in designing model before deployed. We evaluate the energy consumption of a wireless sensor node whether obey a certain bound. If not, the design model need to be modified based on the constraints and evaluation results. According to the duality between time and reward, which means that the progress of time can be regarded as the earning of reward and vice versa [12, 21], when running, the response time and energy consumption are the main aspects which we will consider simultaneously.

Example 1: As a running example, we consider a simple wireless sensor network application. Its objective is to sense, receive and process data in a bounded area, and then to forward the information. We assume that this type of nodes have the following modules such as sensor, communication, computation, power, storage, and etc. Initially, the node is in sensing mode. When some sensor data is collected it will receive all these data, and then process them. When finishing the process procedure, it will forward the information towards some fixed and distinguished sensor nodes or base stations. Apparently, the energy consumption levels of distinguished mode are different.

In design phase, for the above wireless sensor node example, consider an event that the sensed data is received, processed and sent successfully, we may ask the following questions in natural language:

Q1: The probability of the event occurs when the maximal response time falls into [0, 10];

Q2: The probability of the event occurs when the maximal energy consumption is smaller than 20;

Q3: The probability of the event occurs when the maximal response time and energy consumption fall into the interval [0, 10] and [0, 20], respectively;

The above questions are performance and dependability properties. In the area of formal verification and analysis, states transition systems (STS) are often used as models to describe the behavior of systems. A state denotes a relative steady period, and only in this period one can observe or capture some available information of the considered system. Therefore, a state describes some information about a system at a certain moment of its behavior. In performability evaluation, the employed model is usually a probabilistic or stochastic extension style of STS.

In this paper, we will answer these questions by stochastic model checking technique. We describe the basic idea of stochastic model checking firstly, and then introduce the basic definitions of the employed models and temporal logic.

2.2 Overview of Stochastic Model Checking

Model checking is a formal verification technique that explores all possible states of considered system in a brute-force manner [10], i.e., it allows for desired properties of a considered system to be verified on the basis of suitable model of the system through inspection of all states of the system model [10]. Its notable character is that it is completely automatic and it can offer counterexamples in case a model fails to satisfy a property. It is a very attractive approach toward the correctness of information and communication systems.

Probability is an important phenomenon in the design and analysis of information and communication systems. It can be used to model unreliable or unpredictable behavior in order to analyze system performance and dependability. Stochastic model checking, which is an extension of traditional model checking, is a successful and well established technique for formally verifying performance and dependability of a finite state system which exhibits stochastic behavior. Initially, it involves the construction of a stochastic model of the real-life system which is to be verified. Generally, the frequently-used theoretical model may be discrete-time Markov chain (DTMC), Continuous-time Markov Chain (CTMC) or Continuous-time Markov Decision Process (CTMDP) which represents all possible configurations the system may be in and all transitions which may be triggered. The respected properties to be verified are also formally specified as formulas of a kind of stochastic temporal logic such as Continuous Stochastic Logic (CSL) and Continuous Stochastic Reward Logic (CSRL), and etc. [22-24]. These logics are capable of expressing temporal relationships between events and likelihood of them. In this paper, we use stochastic model checking as the performability evaluation approach of wireless sensor node. For the sake of simplicity, we ignore the internal or external nondeterminism [25-28]. We model the behavior of a wireless sensor node to a labeled CTMC, and the performance and dependability properties to be verified are depicted by CSRL.

2.3 Continuous Time Markov Chain enhanced with State Reward Structure

Markov property is an important mathematical property. It means that if the current state is known, the future states of the system are independent of its past states. Markov property is also named as memoryless property, and exponential distribution is the only continuous distribution which is memoryless [23]. CTMC is a very important stochastic process models which is always used in performance analysis and dependability evaluation. It is a distinct difference that a DTMC precedes in discrete steps, which means that there is no timing information about the transitions take, whereas the transitions in a CTMC can be triggered at any time instant which is governed by exponential distribution [24].

Formally, CTMC is a labeled transition system (LTS) augmented with rates additionally which describe the continuous time-steps and its exponentially distributed sojourn time. As a formal model, CTMCs have been widely used in practice as an important reasoning tool. It is suited for describing performance and dependability characteristics in many domains such as reliability models, control systems, queuing networks, biological, and etc. In this section we give an overview of continuous time Markov chains (CTMCs). For convenience, let AP be a finite set of atomic propositions. Furthermore, we will enhance CTMCs with reward structures. A reward structure can be used to represent additional information about the considered system, for example the power consumption, number of packets sent or the number of lost requests. A reward structure allows one to specify two distinct types of rewards, i.e. state rewards and transition rewards. The former are assigned to states by means of the reward functions whereas the latter are assigned to transitions. The state reward is the reward acquired in a state per time-step, i.e. a reward is incurred if the model is in state s for 1 time-step and the transition reward is acquired each time a transition occurs. In this paper, we only consider reward values which are only attached to states of the models.

Definition 1: A labeled CTMC is a tuple (S, R, L) where S is a set of states, $R{:}S{\times}S{\to}\mathbb{R}_{\geq 0}$ is the rate matrix and $L{:}S{\to}2^{AP}$ the labeling function.

Intuitively, if $R(s, s')>0$ then there exists a transition from s to s' and $1-e^{-R(s,s')\cdot t}$ is the probability that the transition $s \to s'$ can be triggered within t time units. The transition rate $E(s)=\sum_{s'\in S} R(s,s')$ denotes the total rate at which any transition outgoing from state s is taken. If $E(s)=0$, then state s is called an absorbing state. If there are more than one state s' such that $R(s, s')>0$, then there exists a competition between the transitions originating in s [22-24]. For a non-absorbing state s, i.e. $E(s)\neq 0$, the probability to move from s to a particular state s' within t time units is given by:

$$P(s,s',t)=\frac{R(s,s')}{E(s)}(1-e^{-E(s)\cdot t}).$$

Function L assigns to each state $s \in S$ the set $L(s)$ of atomic propositions such that elements of $L(s)$ are valid in s.

In order to reason the quantitative properties about the power consumption of the considered system, we need to enhance a CTMC with a state-based reward structure. For the sake of simplicity, we do not consider the impulse rewards structure which is transition-based.

Definition 2: A labeled Continuous time Markov Reward Model (CMRM) is a tuple (C, ρ) where C is a CTMC and ρ a state-based reward structure which assigns a reward rate to each state such that the residence time in a state results in an accumulation of gained reward.

Example 2: For the wireless sensor node example 1, its CMRM model is depicted in Fig. 1. States are represented by ovals and transitions by labeled edges which give the corresponding transition rate. The main state labels are depicted inside the ovals, and initial states are indicated by having an incoming arrow without source nodes. The states reward is depicted by the nearby real number. All these parameter values are obtained via empirical data. The reward structures are interpreted as energy consumption here.

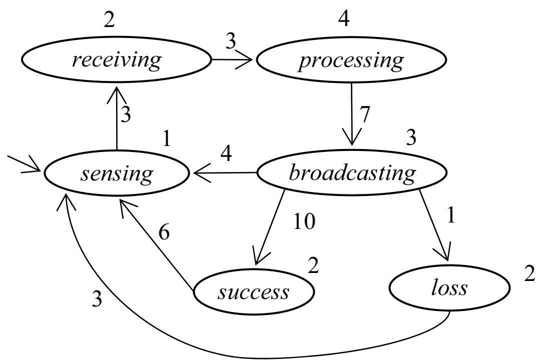

Fig. 1. CMRM model of simple wireless sensor node

Definition 3: An infinite path is an infinite sequence

$$s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \cdots \xrightarrow{t_n} s_n \xrightarrow{t_n} s_{n+1} \xrightarrow{t_{n+1}} \cdots$$

where $R(s_i, s_{i+1})>0$ for i=0, …, n-1,…. A finite path in M is a finite transition sequence

$$s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \cdots \xrightarrow{t_n} s_n$$

where $R(s_i, s_{i+1})>0$ for i=0, …, n-1.

A path in a CMRM M represents an execution which indicates one possible behavior of the system being modeled. Let *Paths* (M, s) denote all infinite and finite paths starting in s in model M and *Paths* (M) all paths in M. *Paths* (M) describes the dynamic behavior of the system under consideration which is modeled by M. In Fig. 1, the event depicted in example 1 can be showed by the paths set in which the path starts from the sensing state and finally arrives at the same state only via the success state.

Definition 4: For a finite path σ in CMRM M, let $\sigma[i] = s_i$ denote the (i+1)-st state, $\delta(\sigma, i) = t_i$ the resident time in state $\sigma[i]$ and $\tau(\sigma) = \sum_{m=0}^{j-1} t_m$ the completion time in path σ. Let $\sigma@t = \sigma[i]$ denote the state in σ at time t where i the smallest index with $t \leq \sum_{j=0}^{i} t_j$. For $t = \sum_{j=0}^{k-1} t_j + t'$ with $t' \leq t_k$ we define $y(\sigma, t) = \sum_{j=0}^{k-1} t_j \cdot \rho(s_j) + t' \cdot \rho(s_k)$, the cumulative reward along σ up to time t [21].

A path in a CMRM specifies a certain behavior. For quantitative analysis, we must define the probability measures on a set of paths and this can be achieved by traditional approach [23-24]. Let *Paths* (M, s) be the sample space. An initial distribution α yields a probability measure $\Pr_a^M$ on paths as follows. In this paper, we assume that there is only one initial state, and thus $\Pr_a^M$ can be written by $\Pr_s^M$. Assumed that $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \cdots \xrightarrow{t_n} s_n$ is a finite path and $I_0,\ldots, I_{k-1}$ non-empty intervals in $\mathbb{R}_{\geq 0}$. Let $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ denote the cylinder set consisting of all paths $\sigma \in Path(M, s_0)$ such that $\sigma[i] = s_i (i \leq k)$ and $\delta(\sigma, i) \in I_i (i < k)$. Let $F(Path(M, s))$ denote the smallest σ-algebra which contains all sets $C(s\ , I_0, \ldots, I_{k-1}, s_k)$ where $s_0,\ldots, s_k$ ranges over all state sequences with $s = s_0$, $R(s_i, s_{i+1})>0$,($0 \leq i < k$), and $I_0, \ldots, I_{k-1}$ ranges over all sequences of non-empty intervals in $\mathbb{R}_{\geq 0}$. The probability measure $\Pr_s^M$ on $F(Path(M, s))$ is the unique measure defined by induction k by [23-24]:

$$\begin{cases} \Pr_s^M(C(s_0)) = a(s_0) \\ \Pr_s^M(C(s_0,\ldots,s_k, I', s')) = \Pr_a(C(s_0,\ldots,s_k)) \cdot P(C(s_k, s') \cdot . \\ \qquad (e^{-E(s_k)\cdot Inf\, I'} - e^{-E(s_k)\cdot Sup\, I'}) \ \ for\ k \geq 0 \end{cases}$$

For a CTMC, we may consider two types of state probabilities, i. e., steady-state probabilities where the system is considered on the long run, and transient probabilities where the system is considered at a given time instant t.

Definition 5: In a CTMC, the transient probability to be in state s' at time t given the initial distribution α are defined as $\pi^M(a,s',t) = \Pr_s^M(\sigma \in Paths(M,s) \mid \sigma@t = s')$ and the steady-state probabilities are defined as $\pi^M(a,s') = \lim_{x\to\infty} \pi^M(a,s',t)$. So, for $S' \subset S$, $\pi^M(a,S') = \sum_{s'\in S'} \pi^M(a,s')$.

3 The Temporal Logic CSRL for Performability Specifications

A CMRM model is a stochastic model and its performability specifications which to be verified need to be characterized by stochastic temporal logic. In order to reason about energy-based as well as time-based constraints behavior of wireless sensor node, the employed temporal logic has to depict time and reward constraints at one time conveniently. Continuous Stochastic Reward Logic (CSRL) is a specification formal language for performability measures over CMRMs, and it allows one to specify properties over states as well as paths. Especially, it can append time and reward intervals to path formula which means that the behavior satisfying corresponding time and reward constraints. This section presents the syntax and semantics of the CSRL. In CSRL, two kinds of formulas are distinguished, i.e. state formulas and path formulas. A state formula depicts the properties which need to be satisfied in some states and a path formula in some paths.

Definition 6: Let a real number $p \in [0, 1]$ be a certain probability value, *AP* be a fixed set of considered atomic propositions, $\infty \in \{<, <=, >=, >\}$ be a comparison operator, $ap \in AP$, *I* and *J* intervals of non-negative real number. The syntax of CSRL state formulas Φ and path formulas φ can be defined inductively as follows:

$$\Phi ::= tt \mid ap \mid \neg\Phi \mid \Phi \vee \Phi \mid S_{\infty p}(\Phi) \mid P_{\infty p}(\varphi)$$

$$\varphi ::= X_J^I \Phi \mid \Phi U_J^I \Psi$$

The connectives $\neg$ and $\vee$ have their original meanings in linear temporal logic LTL, and other logical connectives can be derived in an obvious way. The steady-state operator $S_{\infty p}(\Phi)$ asserts that the probability of being in a Φ state in the long run obeys the bound ∞p. The operator $P_{\infty p}(\varphi)$ asserts that the probability of all paths which satisfy the path formula φ obeys the bound ∞p. The path formula $\Phi U_J^I \Psi$ holds if Ψ is satisfied at some time instant in the interval I and the earned cumulative reward up to *r* lies in *J* and Φ holds at all preceding time instants. The path formula $X_J^I \Phi$ holds if the next state satisfies Φ at some time instant in the interval *I* and the earned cumulative reward up to *r* lies in *J*.

State formulae are interpreted over the states of a CMRM, and path formulae are interpreted over the paths of a CTMC [11].

Definition 7: CSRL state formulas are interpreted over labeled CMRMs by a satisfaction relation $\models_s$ between a state *s* and a state formula Φ. A satisfaction relation $\models_s$ is called valid iff a state formula is satisfied in a state *s*.

$$
\begin{array}{lll}
s \models_s true & iff & all\; s \in S \\
s \models_s ap & iff & ap \in L(s) \\
s \models_s \neg\Phi & iff & s \not\models \Phi \\
s \models_s \Phi_1 \wedge \Phi_1 & iff & s \models_s \Phi_1 \; and \; s \models_s \Phi_2 \\
s \models_s \mathcal{S}_{\infty p}(\Phi) & iff & \sum_{s'\in Sat(\Phi)} \infty p \\
s \models_s \mathcal{P}_{\infty p}(\varphi) & iff & \Pr ob_s^M(\varphi) \infty p
\end{array}
$$

where $\Pr ob_s^M(\varphi) = \Pr_s^M\{\sigma \in Paths(M,s) \mid \sigma \models_p \varphi\}$ and $\models_p$ is the satisfaction relation between paths and path formulas.

Definition 8: CSRL path formulas are interpreted over the finite and infinite paths of a CMRM model by a satisfaction relation $\models_p$ between a path σ and a path formulaφ. A satisfaction relation $\models_p$ is called valid iff a path formula is satisfied in a path:

$$\begin{aligned}\sigma \vDash_p \mathcal{X}_J^I \Phi \quad iff \quad & |\omega| > 1 \wedge \delta(\sigma,0) \in I \wedge \\ & y(\sigma,\delta(\sigma,0)) \in J \wedge \sigma[1] \vDash_s \Phi\end{aligned};$$

$$\begin{aligned}\omega \vDash_p \Phi U_J^I \Psi \quad iff \quad & \exists t \in I.(\sigma @ t \vDash_s \Psi \\ & \wedge (\forall t' \in [0,t).\sigma @ t' \vDash_s \Phi) \wedge y(\sigma,t) \in J)\end{aligned}.$$

In order to satisfy a next formula $X_I^J \Phi$, a path σ must have at least length 1 and next state must satisfies state formula Φ. The residence time of the first state has to fall in the interval *I*, and the reward accumulated in the first state in the interval *J*. A path σ satisfies $\Phi U_J^I \Psi$ if there is a time $t \in I$ such that: The state of the path at time *t* satisfies Ψ; the state of the path at all times *t'* before *t* satisfies Φ; the accumulated reward at time *t* falls in *J*.

4 Stochastic Model Checking Low-Powered Sensor Node

The stochastic model checking procedure for CSRL specifications in CMRM is similar to that for CTL in LTS. A stochastic model checkers accepts a description of a considered model, represented as a extended state transition system, and a specification, typically a formula in some temporal logic, and return 'yes' or 'no', indicating whether or not the model satisfies the specification. Ignoring reasoning details, this can be reduced to computing the satisfactory sets of all sub-formulas of the specification. The computing rules of satisfactory set for CSRL specifications in CMRM are as follows:

$$\begin{aligned}&Sat(true) = S,\\ &Sat(false) = \varnothing,\\ &Sat(a) = \{s \mid a \in L(s)\},\\ &Sat(\neg\Phi) = S \setminus Sat(\Phi),\\ &Sat(\Phi_1 \vee \Phi_2) = Sat(\Phi_1) \bigcup Sat(\Phi_2),\\ &Sat(\mathcal{S}_{\infty p}(\Phi)) = \{s \in S \mid \pi^M(s, Sat(\Phi)) \infty p\},\\ &Sat(\mathcal{P}_{\infty p}(\varphi) = \{s \in S \mid \Pr ob_s^M(s,\varphi) \infty p\}\end{aligned}$$

For basic operators and $\mathcal{S}_{\infty p}(\Phi)$, the satisfactory set can be computed by the same procedure as for CSL [11]. The treatment of formulas of type $\mathcal{P}_{\infty p}(\varphi)$ is more complicated. According to its semantics rule, for each state s we have to compute the probability $\Pr ob_s^M(\varphi)$ and check whether it obeys the specified bound ∞p.

Markov Reward Model Checker (MRMC) [13] is a model checker for discrete-time and continuous-time Markov reward models. It is distributed under the GNU General Public License (GPL) and requires the GNU Scientific Library (GSL) which is a collection of numerical routines for scientific computing. It supports reward-based specifications such as PRCTL [10] and CSRL, and allows for the automated verification of properties concerning long-run and instantaneous rewards as well as cumulative rewards. In particular, it supports to check the reachability of a set of goal states under a time and an accumulated reward constraint.

In this paper, the adopted versions are MRMC 1.5 and GSL 1.9, respectively. Additionally, in order to install GSL on Windows we installed Cygwin firstly and then to perform GSL installation procedure using the Cygwin shell. We have installed MRMC on a notebook with an Intel(R) Core(TM) i3 2.53GHz processor and 1.86 GB main memory to check the formalized requirements. MRMC is a command-line tool that supports an easy input format. Its basic input files include *.lab file which indicates the state labels with atomic propositions, *.rew file which specifies the state-based reward structure and *.tra file which describes the rate matrix. For example 1, Table 1 shows three basic input files which must be loaded before execution the verification. The command line syntax is $./bin/mrmc cmrm c.tra c.lab c.rew where the first parameter indicates the type of input model. We consider the CSRL formula $\mathcal{P}_{\geq 0.90}(normal\ U_{[0,J]}^{[0,I]}\ (success))$. Here, *I* and *J* are nonnegative real numbers which denote the upper bound of corresponding constraint interval.

Table 1. The input files of Fig. 1

c.lab	c.tra	c.rew
# DECLARATION	STATES 6	1 1
sensing receiving processing	TRANSITIONS 8	2 2
broadcasting success loss normal	1 2 3	3 4
#END	2 3 3	4 3
1 sensing normal	3 4 7	5 2
2 receiving normal	4 1 4	6 2
3 processing normal	4 5 10	
4 broadcasting normal	4 6 1	
5 success	5 1 6	
6 loss	6 1 3	

When starting from the normal state and finally into the success state in I time-units (I=1, 2,…, 10), Fig. 2 reports the probability results when reward constraint is varying. The results show that the probability becomes larger along with the increasing of maximum reward constraint. This is to be expected since, for larger reward constraint, the likelihood of satisfying the path formula and a fixed time constraint will become larger as well. When the constraint on the maximum reward arrives at some threshold, the probability increases gently and finally converge at some point. For a fixed reward constraint, if the time constraint increases, the probability increases too. When the time and reward constraints increase synchronously, the probability converges at 0.90. Fig. 3 reports similar results.

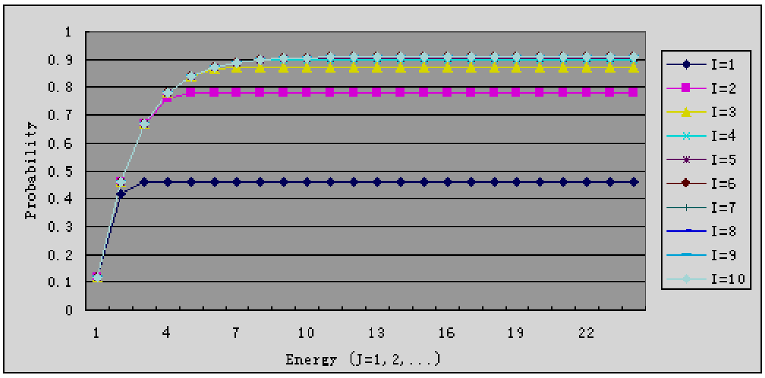

Fig. 2. Probability of the event before time I with accumulated energy less than J

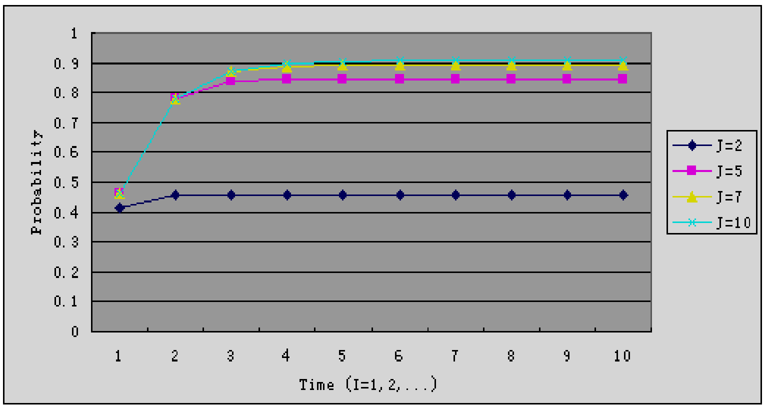

Fig. 3. Probability of the event with accumulated energy less than J before time I

In addition, we also report the probability regions when time and energy constraints varying in Fig. 4 by 3-dimensional overlooked view, and the above two regions imply that the probability is greater than 0.90. It indicates that if the time constraint falls into the interval [4-5] and the reward constraint falls into the interval [8-9], the probability meets the performance and dependability property, namely the expected state formula $\mathcal{P}_{\geq 0.90}(normal\ U_{[0,J]}^{[0,I]}\ (success))$.

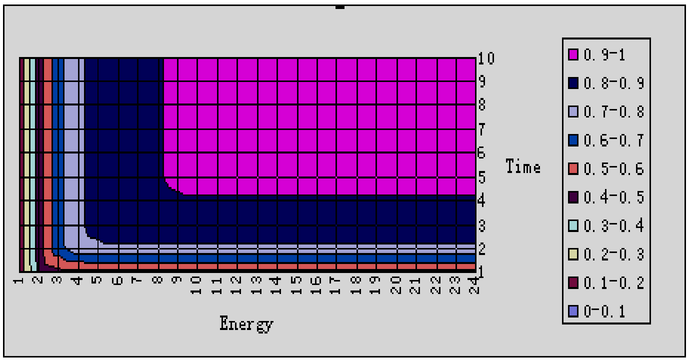

Fig. 4. 3-dimensional overlooked view of probability distribution when time and energy constraints varying

The above experiment shows the probability distribution when time and energy constraints varying. Afterwards, it provides the evaluation evidence for a performability formula. For fixed time and energy constraints I and J, we can deduce that the formula is valid or not. If the latter, we have to modify the considered model according to the experiment results, even the generated counterexample based on the model checking procedure [10]. When model checking, the possible emerging state explosion problem can be treated by state space reduction technique such as partial order reduction, bisimulation and etc. [13].

5 Conclusions and Future Work

This paper has presented a performance and dependability verification approach of a wireless sensor node by stochastic model checking technique. In order to reason about the performability properties which include energy consumption constraint, the dynamic behaviors of a sensor node is modeled by a continuous time markov reward model, which is enhanced with reward structure in order to depict the energy consumption, and the performability are specified by continuous stochastic reward logic formulas. Thus, we can obtain the verification results via Markov Reward Model Checker MRMC. It can ensure that only the practicable designing should be taken into account. In addition, in design-time, this work can provide beneficial evidence to modify the designing and save costs. From the verification results of the example, it is obvious that the upper bounds of the time or reward constraints will affect the probability of certain specifications, and the probability will be steady when they satisfy some relation. In the future, for sensor nodes, we plan to consider how to get an equilibrium point between time and energy constraints at which the probability will be convergent, and provide the theoretical proof. As a result, for a given performability formula, we may obtain the verification conclusion according to the formula itself, and this may help us to avoid the model checking procedure firstly in some sense.

Acknowledgement

This research was supported by the National Natural Science Foundation of China under grant No. 61272107, 51204185, the Zhejiang Provincial Natural Science Foundation of China under Grant No. LY15F020010, the Zhejiang Public technology applied research projects under Grant No. 2014C31059, the open Project of the State Key Laboratory of Software Engineering under Grant No. SKLSE2014-10-05, the open Project of the Key Laboratory of Ministry of Education Embedded System and Service Computing under grant No. ESSCKF201302.

References

[1] C. Perera, A. Zaslavsky, P. Christen, D. Geogakopoulos, Sensing as a service model for smart cities supported by internet of things, Transactions on Emerging Telecommunications Technologies 25(1)(2014) 81-93.

[2] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, European Commission. Directorate-General for the Information Society and Media, Vision and Challenges for Realising the Internet of Things, Cluster of European Research Projects on the Internet of Things, Brussels, 2010.

[3] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I.S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, P. Doody, Internet of Things-Strategic Research Roadmap, Cluster of European Research Projects on the Internet of Things, Brussels, 2009.

[4] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Computer Networks 54(15)(2010) 2787-2805.

[5] P. C. Olveczky, S. Thorvaldsen, Formal modeling, performance estimation, and model checking of wireless sensor network algorithms in real-time maude, Theoretical Computer Science 410(2-3)(2009) 254-280.

[6] W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38(4)(2002) 393-422.

[7] J. Green, S. Bhattacharyya, B. Panja, Real-time logic verification of a wireless sensor network, in: Proc. 2009 World Congress on Computer Science and Information Engineering, 2009.

[8] J.K. Jacoub, R. Liscano, J.S. Bradbury, A survey of modeling techniques for wireless sensor networks, in: Proc. International Conference on Sensor Technologies and Applications, 2011.

[9] M. Clarke, O. Grumberg, D.A. Peled, Model Checking, MIT Press, Cambridge, 2000.

[10] C. Baier, J. -P. Katoen, Principles of Model Checking, MIT Press, Cambridge, 2008.

[11] M. Kwiatkowska, G. Norman, D. Parker, Stochastic model checking, in: Proc. Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation, 2007.

[12] B. Haverkort, L. Cloth, H. Hermanns, J.-P. Katoen, C. Baier, Model checking Performability properties," in: Proc. DSN 2002, 2002.

[13] J.-P. Katoen, Perspectives in probabilistic verification, in: Proc. IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering, 2008.

[14] M. Woehrle, K. Lampka, L. Thiele, Exploiting timed automata for conformance testing of power measurements, in: Proc. FORMATS 2009, 2009.

[15] S. Tschirner, L. Xuedong, W. Yi, Model-based validation of QoS properties of biomedical sensor networks, in: Proc. ACM & IEEE International Conference on Embedded software, 2008.

[16] Demaille, S. Peyronnet, T. Hérault, Probabilistic verification of sensor networks, in: Proc. 4th IEEE International Conference. on Computer Science, 2006.

[17] R. Abo, K. Barkaoui, A performability analysis of mobile wireless sensor networks with probabilistic model checking, in: Proc. Wireless Advanced, 2011.

[18] P. Ballarini, A. Miller, Model checking medium access control for sensor networks, in: Proc. the Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, 2006.

[19] S. Coleri, M. Ergen, T.J. Koo, Lifetime analysis of a sensor network with hybrid automata modeling, in: Proc. the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[20] P.C. Ölveczky, S. Thorvaldsen, Formal modeling and analysis of the ogdc wireless sensor network algorithm in real-time

maude, in: Proc. the 9th IFIP WG 6.1 International Conference on Formal Methods for Open Object Based Distributed Systems, 2007.

[21] C. Baier, B.R. Haverkort, J.-P. Katoen, H. Hermanns, On the logical specification of performability properties, in: U. Montanari, J.D.P. Rolim, E. Welzl (Eds.), Automata, Languages and Programming. ICALP 2000. Lecture Notes in Computer Science, Vol. 1853, Springer, Berlin, 2000, pp.780-792.

[22] C. Baier, L. Cloth, B.R. Haverkort, M. Kuntz, M. Siegle, Model checking Markov chains with actions and state labels, IEEE Transactions on Software Engineering 33(4)(2007) 209-224.

[23] C. Baier, B.R. Haverford, J.-P. Katoen, Model checking continuous-time Markov chains by transient analysis, in Proc. International Conference on Computer Aided Verification, 2000.

[24] C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, Model-checking algorithms for continuous-time Markov chains, IEEE Transactions on Software Engineering 29(6)(2003) 524-54.

[25] Baier, C., Hermanns, H., Katoen, H.R. Haverkort, Efficient computation of time bounded reachability probabilities in uniform continuous-time Markov decision processes, Theoretical Computer Science 345(1)(2005) 2-26.

[26] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley and Sons, New York, 1994.

[27] M.R. Neuhäußer, J.P. Katoen, Bisimulation and logical preservation for continuous-time Markov decision processes, Lecture Notes in Computer Science 4703(2007) 412-427.

[28] N. Wolovick, S. Johr, A characterization of meaningful schedulers for continuous-time Markov decision processes, Lecture Notes in Computer Science 4202(2006) 52-367.