Sensors & Transducers, Vol. 183, Issue 12, December 2014, pp. 1-7

Sensors & Transducers

Performance Analysis of Binary Search Algorithm in RFID

Xiangmei SONG, Henglong DONG, Zhifeng LIU, Conghua ZHOU
School of Computer Science and Telecommunication Engineering, Jiangsu University,
Zhenjiang 212013, China
Tel.: + 086 0511-88791299
E-mail: jlsxm@ujs.edu.cn

Received: 18 June 2014 /Accepted: 31 July 2014 /Published: 31 December 2014

Abstract: Binary search algorithm (BS) is a kind of important anti-collision algorithm in the Radio Frequency Identification (RFID), is also one of the key technologies which determine whether the information in the tag is identified by the reader-writer fast and reliably. The performance of BS directly affects the quality of service in Internet of Things. This paper adopts an automated formal technology: probabilistic model checking to analyze the performance of BS algorithm formally. Firstly, according to the working principle of BS algorithm, its dynamic behavior is abstracted into a Discrete Time Markov Chains which can describe deterministic, discrete time and the probability selection. And then on the model we calculate the probability of the data sent successfully and the expected time of tags completing the data transmission. Compared to the another typical anti-collision protocol S-ALOHA in RFID, experimental results show that with an increase in the number of tags the BS algorithm has a less space and time consumption, the average number of conflicts increases slower than the S-ALOHA protocol standard, BS algorithm needs fewer expected time to complete the data transmission, and the average speed of the data transmission in BS is as 1.6 times as the S-ALOHA protocol.

Keywords: RFID, Binary search algorithm, Probabilistic model checking, Discrete Time Markov Chains.

1. Introduction

RFID (Radio Frequency Identification), as one of the important technologies of the Internet of Things, is a kind of important wireless collection technology that can identify tag information quickly, timely and accurately, and can realize data exchange with it. It has been widely used in the fields such as logistics (goods reception and delivery) and storage, etc. In the real application of radio frequency identification systems, when multiple tags enter into the reader-writer's sphere of action, many of them will collide for transmitting the data to the reader-writer simultaneously, leaving the reader-writer unable to read the data. Therefore, identification efficiency caused by collision is the most important problem in the RFID technology.

Currently, the tag anti-collision algorithm frequently used in RFID includes the stochastic anti-collision algorithm based on ALOHA Mechanism [1, 2] and the definitive anti-collision algorithm based on Binary Search Algorithm, BS [3]. ALOHA is a kind of random access way which was proposed in earlier times. It is one type of probabilistic algorithm. The tags will choose a certain period of time by randomness to resend data until all of which are able to be identified, when the collision occurs because multiple tags send the data at the same time.

The binary algorithm is also called binary tree algorithm. All electronic tags have their only binary identifiers and they send the sequence numbers of signal tags to the reader-writer synchronously in the reader-writer's sphere of action to construct an ordinary binary tree. The reader-writer will repeatedly screens the leaf node of complete binary trees according to the situation of the signal collision, and finally finds out the binary tree. In the process of searching, electronic tags responding within the areas of action will be determined one by one; meanwhile, information exchange between the tags and reader-writer will be realized. As the Binary Search Algorithm not only needs to ensure that multiple tags can be identified quickly by the reader-writer, but also needs to ensure the integrity of data transmission, so a kind of more reliable and comprehensive analysis method undoubtedly becomes necessary.

Systematic formalization analysis and verification technology have been increasingly applied in the industry. The formalization analysis technology can not only verify the correctness of the system, but also make quantitative analysis on system performance, such as reliability, capacity or energy consumption. In this paper, we will analyze the Binary Search Algorithm by using an automatic formal verification technology which is called probabilistic model checking [4, 5]. The probabilistic model detection tool PRISM [6] has already made a quantitative analysis on the properties of stochastic allocation algorithm and communication protocol, for example, literature [7] has analyzed the accessibility of IEEE1394 Fire Wire, and Stylianos has made a quantitative analysis on the security of e-mail transport protocol(CEMD) in literature [8], while two random distribution algorithms namely the self-stabilizing algorithm and the dining problem of philosophers are respectively analyzed in literature [9,10].

During the detection of the probabilistic model, it will explore the whole state space in order to realize a systematic analysis of attributes. We model the target system S to be an appropriate one M and the properties P to be verified will be expressed by logic formulae ϕ_p. Therefore, the issue of whether the system S satisfies the properties P is converted into the issue of model checking whether $M \models \phi_p$ is true or not. Probabilistic model detection provides an effective formalized specification (including model specification and attribute specification) and precise analysis technology. The analyzed models include Discrete Time Markov Chain (DTMC) [11], Markov Decision Process (MDP) [12], Continuous Time Markov Chain (CTMC) [11] and Probabilistic Timed Automaton (PTA) [13], etc.

The operation principles of the BS algorithm are as follows: After multiple tags enter into the work area of the reader-writer, the reader-writer sends an enquiry order with restrictive condition and the tag meeting the restrictive condition answers. If a collision occurs, the restrictive condition will be changed according to the error position, and the enquiry order is sent again until a correct answer is found, and then the reading and writing operation to the tag will be finished. The operations mentioned above will be repeated on the rest tags until the reading and writing operation to all the tags is finished. After introducing the working principle of the binary system, we will model a DTMC model for it. The reason for choosing DTMC is that Binary Search Algorithm occurs with certain probability no matter in terms of the status transition of the reader-writer or the tags. We naturally consider using discrete time model, in view of the certainty of anti-collision algorithm, we adopt DTMC model after overall consideration.

2. DTMC and PCTL

Definition 1. DTMC is a six-tuple, $D = (S, P, s_0, AP, L, R)$

- S is a finite set of states here;
- $P : S \times S \rightarrow [0,1]$ is a transaction concept function, and for any state $s \in S$, there is: $\sum_{s' \in S} P(s, s') = 1$
- $s_0 \in S$ is the initial state;
- AP is a finite set of atomic propositions;
- $L : S \rightarrow 2^{AP}$ is a marking function;
- R is an reward. The reward structure of DTMC is a binary group (ρ, l), of which one $\rho : S \rightarrow R_{\geq 0}$ is a state reward vector and $l : S \times S \rightarrow R_{\geq 0}$ is a transfer reward vector.

Definition 2. (Probabilistic Computation Tree Logic PCTL [11]) The PCTL state equation in atomic proposition AP:

$\phi ::= \text{true} \mid a \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \neg\varphi \mid P_J(\varphi)$, $a \in AP$.

Herein φ is a path formula, and $J \in [0,1]$ is an interval with rational numbers as the boundaries. PCTL path formula is described as $\varphi ::= X\phi \mid F\phi \mid G\phi \mid \phi_1 U \phi_2 \mid \phi_1 R \phi_2$. Herein, ϕ , ϕ_1 and ϕ_2 are the state equations.

We use $s \models \varphi$ to show that a state $s \in S$ meets a PCTL formula ϕ . An important formula $P_J(\varphi)$ in PCTL shows that the probability that a state meets the path ϕ is in the range of J. In the path $X\phi$ of definition 2 (the next step to meet ϕ), $\phi_1 \vee \phi_2$ presents it will eventually meet ϕ_2 if ϕ_1 is true.

Two attribute instances expressed by PCTL are as follows:

$P = ?[F"finish"]$ shows the probability of the data transmission accomplished eventually by the system.

$P = ?[\neg fail_A U fail_B]$ represents the probability that A succeeds before B fails.

3. Abbreviations and Acronyms

3.1. Working Principle of the Protocol

BS algorithm uses the reader-writer control anti-collision method. Its basic principle is by defining a set of specified instruction sequence between reader-writer and multiple tags to choose a transponder from them and complete the data exchange between them finally. The binary anti-collision algorithm is essentially comparing all tags one by one, by multiple comparisons, the appropriate tag could be selected gradually to finish the information exchange, and then the tags will be compared one by one again until all the tags are recognized by the reader-writer. The reader-writer selects tags through four commands.

1) Request order: this order sends a referential sequence number to an electrical tag, and only when the sequence number of the tags is no more than the referential sequence number can the sequence number be sent to the reader. Thus narrow the scope of tag group which is to be identified.

2) Select order: choose a certain sequence number as the referential sequence number to send to the electronic tag. When the sequence number of the tag is the same with that of the reference, the tag is chosen.

3) Data Reading: The reader reads the data of the chosen tag.

4) Cancelling selection: a tag will come into sleep state and not respond to the reader's orders if the selection is cancelled.

3.2. DTMC Model of Protocol

The model of BS algorithm has two parts, the reader and Tag (multiple). Here we will conduct abstract modelling for the four orders of the reader-writer mentioned above. The DTMC state transition diagram of the tag is shown below as Fig. 1.

Step 1: Set s_0 as the initial state of the tag state transition diagram, which means the reader-writer send a sequence number as a parameter to the tag. At this time the tag needs to compare its own sequence number with that of the received one. If the tag's sequence number is smaller than or the same as sequence number of the reader-writer, it will send back its own sequence number to the reader-writer.

Step 2: After sending the sequences at initial state s_0, it will enter into the state s_1. This state indicates that if there is only one tag meeting the conditions, it will enter the state s_3 directly and thus completing the transfer of data. Otherwise it will enter the state s_2 when the collision occurs.

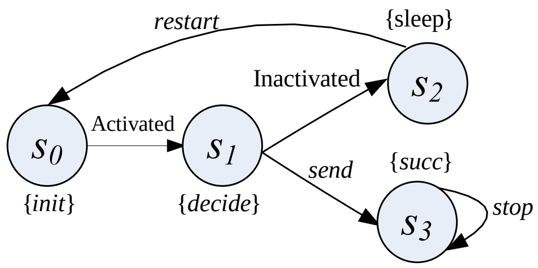

Fig. 1. Tag state transition diagram.

Step 3: Because of the collision between two or more tags, now being in state s_2 shows the collision occurs and it goes directly into the initial state s_0. Enter into the next round of selection process.

Step 4: In the state s_3, it represents that data transmission of tags is successful.

The DTMC model of the reader-writer is shown as Fig. 2. Firstly the reader-writer detects tags around and if there exist multiple tags, the reader-writer will filter tags by sending a sequence number and re-filter by changing the sequence number through a collision until all the tags are identified.

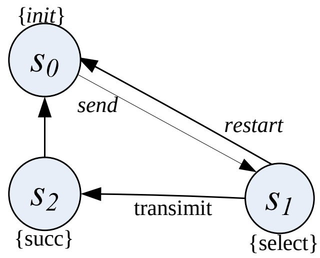

Fig. 2. State transition diagram of the reader-writer transmission of tags is successful.

Step 1: In the state s_0, the reader-writer is in initial state. At the moment the reader-writer sends a sequence number to the tags around to arrive at the state s_1. If only one tag meets the conditions, it will be selected. If a lot of tags meet the conditions, they will turn back to the initial state and be continuously selected.

Step 2: In the state s_1, the labels are in selected status. If only one label is selected, it will enter state s_2 directly, meaning that the label and data of reader-writer have been exchanged. Otherwise enter the state s_0, and labels will be selected again.

Step 3: The state s_2 indicates that a tag has been identified and data transmission is finished. At the moment, directly enter initial state s_0 to conduct the selection of next round, until all the tags are selected.

4. Model Verification and Analysis

4.1. Modeling Statistics

PRISM is the most successfully applied probability model-checking tool at present, which can directly support DTMC, CTMC, MDP and PTA. PRISM has been applied to the wireless communication protocol, such as, Bluetooth and ZigBee wireless communication protocol, the management plan of dynamic power and the algorithm analysis of many random distribution.

We will use this tool to complete the accurate analysis of BS algorithm. Firstly, we need to use PRISM to construct corresponding mathematical model, which is to describe BS algorithm as DTMC model with PRISM language. Then calculate the set of reachable state from initial state and the transfer matrix of the model through PRISM, and conduct quantitative model checking to a plurality of attributes (includes one or more variables).

First we set up some basic parameters of binary search algorithm model; the parameter list is shown in Table 1. Through PRISM calculation, we can get the static data of every building model, as Table 2 shows, N represents label amount, state and switching of state are the general number of DTMC model that corresponding with label. Node and leaf respectively denote the number of nodes and leaves (end-node) of Multi-terminal Binary Decision Diagram (MTBDD [9]). Build time is the time for each model building.

Table 1. Protocol parameters.

Parameters	Description	Value
N	Number of Tags	2-20
K	Number of Choice	Minimum value 1
Count	Read the label counter	Minimum value 1
U	If there are label conflicts	False(Initial value)
Sigma	Propagation time of information in the air	1 us

Table 2. Static data.

N	2	3	4	5	6
State	50	394	1146	4006	50650
Node	1153	3620	5700	7908	17084
Leaf	9	10	11	11	12
Iteration	7	8	8	9	10
Switching of State	107	926	2688	9455	119861
Build Time(s)	0.0141	0.025	0.047	0.058	0.073

From Table 2 we can know state number of BS algorithm's DTMC model and switching times have increased several times as the label number increases, and the time of building the model is long enough. This suggests that the more the number of labels are, the bigger the state space of the relevant DTMC model and the cost of time are. But the leaf number increases slowly. This suggests data structure of MTBDD is fine, and it can apply to the BS algorithm DTMC model efficiently.

4.2. Quantitative Analysis of the BS Algorithm Attributes

Reliability and rapidity are two key attributes of BS algorithm, which are the keys for the tags to be identified quickly and accurately. Reliability means the ability and possibility (probability) of system when it executes certain functions under certain condition for some time. Rapidity means the capability of transmitting certain amount of data; apparently the performance of the BS arithmetic directly influences the quality of service of the whole RFID system.

4.2.1. Reliability

A). Whether one (or more) tab can complete data transmission successfully.

Herein the first reliability we will verify is whether one (or more) tab can complete data transmission successfully, in another word, whether the probability of each tab to finish data transmission is 1. As shown in Table 3, the PCTL formula is expressed as $P=?[\,true\,U(s1=3\,\&\,r=3)]$, in which $s1=3$ and $r=3$ respectively represents the tag successfully transmits data and the reader successfully accepts data, true means begin from the original state and N means the number of the tags.

Table 3. Verification Parameters 1

PCTL Formula	Attribute description	Parameters
$P=?[\,true\,U(s1=3\&r=3)]$	Probability for a label to complete the data transmission	N=1-20

Using the attribute validation function of PRISM, we can count probability of arriving $s1=3\,\&\,r=3$ from the initial state, as the result shown in Fig. 3.

From Fig. 3, we can see that the quantity of tags increasing from 1 to 10, and the probability of tag finishing data delivery and reader receiving data is always 1, that is to say, not only the tag can always finish the transmission of data, but the reader can also successfully read the data, so as to ensure the reliability of the data transmission of BS algorithm.

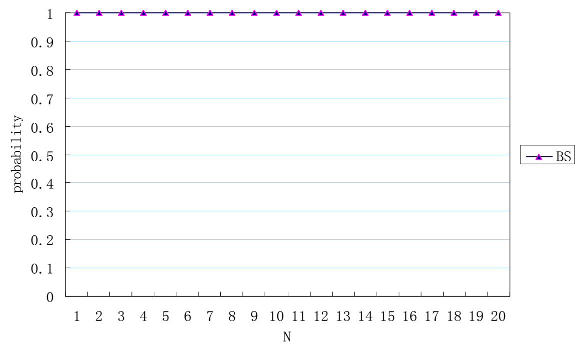

Fig. 3. Eliability of data transmission.

B). The probability of sending data successfully after selecting tag for the K time.

The times BS algorithm selects tag directly affect information transmission efficiency. As shown in Table 4, we use PCTL formula $P=?[\,true\,U(s1=3\&r=3)\&select=k]$ to verify the effects of selecting times on the data transmission credibility. The test result of model attributes is as shown in Fig. 4.

It can be seen from Fig. 4 that with the increase of label number, the number of selection times also increases but in a slow pace. Six labels call for at most five selections, which indicate that BS algorithm is highly efficient in label recognition.

Table 4. Verifying parameters 2

PCTL Formula	Attribute description	Parameters
$P=?[\,true\,U\ (s1=3\&r=3)\ \&select=k]$	The probability of successfully exchange information between the tag and reader after selecting K times.	N=2-6 K=1-10

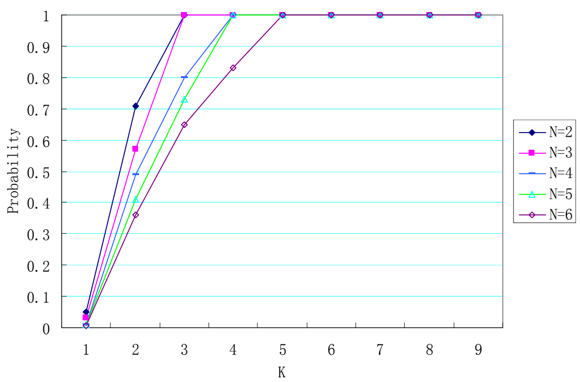

Fig. 4. The probability of sending data successfully after selecting tag K times.

C). Verification and comparison of BS algorithm and S-ALOHA protocol attribute

Based on the situation that the model suits the BS algorithm, we will make a thorough verification analysis through comparing with the S-ALOHA protocol. Given that the random waiting system of S-ALOHA possesses an attribute of uncertainty, we will establish a proper MDP model for it. Table 5 shows the static data used in building BS algorithm and S-ALOHA protocol, among which N represents the number of tags, states and Construction time respectively represent the number of states and time of the components of the relevant model.

Table 5. Comparison between the static data of BS algorithm and S-ALOHA protocol.

N	States		Construction time (s)	
	BS	S-ALOHA	BS	
2	50	426	0.0141	0.054
3	394	1752	0.025	0.077
4	1146	4285	0.047	0.124
5	4006	7827	0.058	0.153
6	50650	108214	0.073	0.229

Table 5 indicates that whether state amount or component time of BS algorithm model is much less than that of S-ALOHA. This shows that BS algorithm has less space dissipation than S-ALOHA.

BS algorithm and S-ALOHA protocol collision time plays a critical role. The more conflicts exist, the time spending in identification tags by recognizer is longer, and the efficiency of information interchange is lower. Therefore, we will verify the contrast of conflict time occurred in different label numbers, as shown in the Fig. 5.

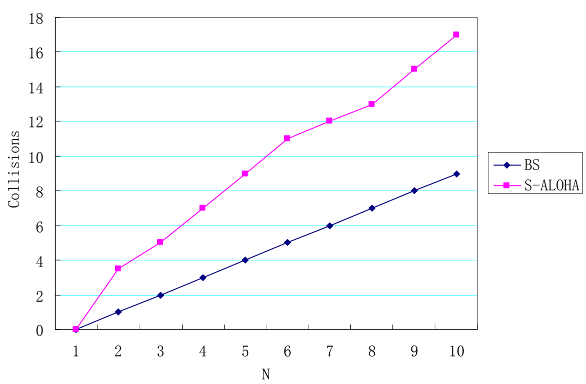

Fig. 5. Contrast of BS algorithm and S-ALOHA conflict times.

We can see from Fig. 5 that with the increase of the number of labels, the number of conflicts occurred will has a slow and steady increase in BS algorithm and a rapid increase in S-ALOHA; when there are two labels, BS algorithm will send the data successfully after two conflicts at most, while the number of conflicts in S-ALOHA is close to four. When there are 10 labels, BS algorithm only has 9 conflicts and S-ALOHA has 17 conflicts. Thus, the BS algorithm performs more efficient and stable than S-ALOHA.

4.2.2. Rapidability

In BS algorithm, whether the tag can exchange information with the reader-writer quickly is crucial. Herein we adopt reward mechanism in the DTMC model of BS algorithm to calculate the expected time of exchange information between reader-writer and tag and compare it with the expected time of exchange information of S-ALOHA. In PRISM, $R\{"time"\} = [\ true\, U (s1 = 3\, \&\, r = 3)]$ refers to the time the tag completes the data in the end. Fig. 6 shows the expected time of sending data with different number of tags by using BS algorithm and S-ALOHA.

As shown in Fig. 6, we can see that the number of tags is from 1 to 20. The expected time by using BS algorithm is less than by using S-ALOHA algorithm to complete the transmission of the data. The expected time difference is 200 units of time when the number of tags is 20, which illustrates that BS algorithm's efficiency is higher that of S- ALOHA when dealing with a few tags.

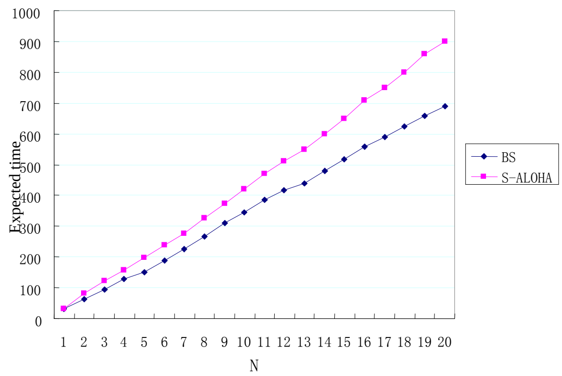

Fig. 6. The expected time of BS algorithm and S-ALOHA under the condition of different number of tags.

We continue to verify the effect of system model expected time by sending different size of data packets. As shown in Fig. 7, in the case of two labels, we set different amount of data to analyze the expected time of system finishing data transmission.

The horizontal axis of the coordinate refers to the size the data package sent and the vertical axis refers to the expected time to complete corresponding size of data package. As is shown in Fig. 7, with the increase of data size, the expected time for BS algorithm and S-ALOHA to complete data transmission also increase; but the speed increase of S-ALOHA is significantly greater and the growth rate is approximately 1.6 times of the speed increase of BS algorithm, which indicates that the average speed for BS algorithm to send data is 1.6 times faster than that of S-ALOHA.

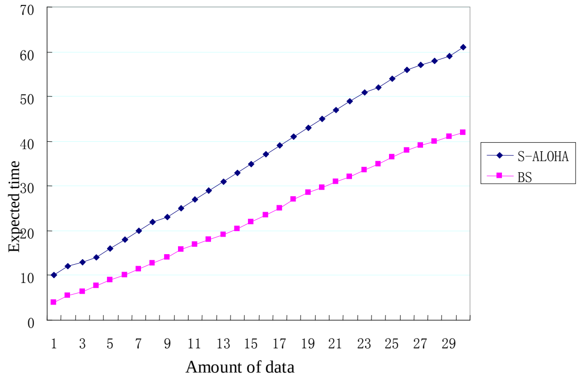

Fig.7. The expected time for BS algorithm and S-ALOHA to send different data volume.

5. Summary

This article employs a supermatic formalization method: Probabilistic model checking, quantitatively analyzing the binary searching algorithm in RFID. According to the principles of BS algorithm, establish the discrete time Markov chain (DTMC) model, which supports decisive selection, then conduct analysis to the reliability and rapidability of BS algorithm, and the conclusion reveals that: 1) the tag delivery conflict times in BS algorithm are less than that of S-ALOHA apparently, which shows that BS algorithm has higher efficiency in dealing with the information exchange between tag and reader; 2) when multiple tags are communicating with the reader at the same time, the expected time of the BS algorithm and S-ALOHA have difference of 200 time units at the most. Sending the data packages with different volumes, the average speed of BS algorithm is also 1.6 times faster than that of S-ALOHA.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61300228, 61300229, 61003288) and Natural science fund for colleges and universities in Jiangsu Province (No. 12KJD580002).

References

[1]. F. Baccelli, B. Blaszczyszyn, and P. Mühlethaler, An Aloha Protocol for Multihop Mobile Wireless Networks, *IEEE Transactions on Information Theory*, Vol. 52, No. 2, February 2006, pp. 421-436.

[2]. Su-Ryun Lee, Sung-Don Joo, Chae-Woo Lee, An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification, in *Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems, Networking and Services*, 17-21 July 2005, pp. 166-172.

[3]. K. Parveen, S. A. Khader, M. A. Rabbani, Analysis of bit grouping algorithm for collision resolution in passive RFID tags, *International Journal of Engineering Science and Technology*, Vol. 2, No. 9, 2010, pp. 4192-4204.

[4]. C. Baier, E. Moritz-Hahn, B. R. Haverkort, H. Hermanns, and J.-P. Katoen, Model Checking for Performability, *Mathematical Structures in Computer Science*, Vol. 23, 2013, pp. 751–795.

[5]. M. Kwiatkowska, G. Norman and D. Parke, PRISM: Probabilistic Model Checking for Performance and Reliability Analysis, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 36, No. 4, 2009, pp. 40-45.

[6]. M. Kwiatkowska, G. Norman, D. Parker, PRISM2.0: A tool for Probabilistic model checking, in *Proceedings of 1st IEEE International Conference on Quantitative Evaluation of Systems (QEST'04)*, 2004, pp. 322–323.

[7]. C. Daws, M. Kwiatkowska, G. Norman, Automatic verification of the IEEE1394 root contention protocol with KRONOS and PRISM, *International Journal on Software Tools for Technology Transfer*, Vol. 5, No. 2, 2004, pp. 221–236.

[8]. B. Stylianos, P. Sophia, A. Nikolaos, P. Georgios, K. Panagiotis, Quantitative analysis of a certified email protocol in mobile environments: A probabilistic model checking approach, *Computers & Security*, Vol. 30, No. 4, 2011, pp. 257-272.

[9]. M. Kwiatkowska, G. Norman, D. Parker, Probabilistic verification of Herman's self-stabilisation algorithm, *Formal Aspects of Computing*, Vol. 24, No. 4, 2012, pp. 661-670.

[10]. M. Duflot, L. Fribourg, C. Picaronny, Randomized dining philosophers without fairness assumption, *Distributed Computing*, Vol. 17, No. 1, 2004, pp. 65-76.

[11]. M. Kwiatkowska, G. Norman, and D. Parker, Stochastic Model Checking, *Lecture Notes in Computer Science*, Vol. 4486, 2007, pp. 220-270.

[12]. M. Kwiatkowska, G. Norman, D. Parker, Stochastic Model Checking, *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation*, 2007.

[13]. G. Norman, D. Parker, and J. Sproston, Model Checking for Probabilistic Timed Automata, *Formal Methods in System Design*, Vol. 43, No. 2, September 2013, pp. 164-190.