Formal Analysis of a Scheduling Algorithm for Wireless Sensor Networks

Maissa Elleuch[1,2], Osman Hasan[2], Sofiène Tahar[2], and Mohamed Abid[1]

[1] CES Laboratory, National School of Engineers of Sfax, Sfax University
Soukra Street, 3052 Sfax, Tunisia
maissa.elleuch@ceslab.org,
mohamed.abid@enis.rnu.tn

[2] Dept. of Electrical & Computer Engineering, Concordia University
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
{melleuch,o_hasan,tahar}@ece.concordia.ca

Abstract. In wireless sensor networks (WSNs), scheduling of the sensors is considered to be the most effective energy conservation mechanism. The random and unpredictable deployment of sensors in many WSNs in the open fields makes the sensor scheduling problem very challenging and thus randomized scheduling algorithms are used. The performance of these algorithms is usually analyzed using simulation techniques, which do not offer 100% accurate results. Moreover, probabilistic model checking, when used, does not include a strong support to reason accurately about statistical quantities like expectation and variance. In this paper, we overcome these limitations by using higher-order-logic theorem proving to formally analyze the coverage-based random scheduling algorithm for WSNs. Using the probabilistic framework developed in the HOL theorem prover, we formally reason about the expected values of coverage intensity, the upper bound on the total number of disjoint subsets, for a given expected coverage intensity, the lower bound on the total number of nodes and the average detection delay inside the network.

Keywords: Probabilistic reasoning, Theorem proving, Higher-order-logic, Wireless sensor networks, Scheduling, Coverage.

1 Introduction

Wireless sensor networks (WSNs) [24] have been proposed as an efficient solution to monitor a field without any continuous human surveillance. Such networks are composed of small tiny devices wirelessly connected over the field. The main task of sensors consists in taking measurements of the monitored event. According to these measurements, a decision procedure is made at the base station. The WSNs are extensively being deployed these days in a variety of applications like detection of natural disasters or biological attacks and military tracking.

Minimizing energy requirements for the sensor nodes is very critical given the fact that these nodes are always stand-alone and battery powered. Scheduling [14] of the nodes is one of the most widespread solutions to preserve energy. It

S. Qin and Z. Qiu (Eds.): ICFEM 2011, LNCS 6991, pp. 388–403, 2011.

consists in splitting the network on several sub-networks, which work alternatively. The biggest challenge involved in this approach is the ability to provide continuous coverage, i.e., reliable monitoring or tracking by sensors.

For inhospitable fields where the sensors are arbitrarily deployed, the trend is to use a random scheduling scheme. As the study of random scheduling algorithms for WSNs is recent, the focus is to investigate more in developing new models that can satisfy the coverage constraint. In general, a theoretical paper-and-pencil based model of the proposed scheduling algorithm is developed and analyzed. After that, performance evaluation by simulation is done in order to illustrate the theoretical results. Nevertheless, the results obtained by simulation can never be totally accurate. Thus, simulation cannot be considered as a reliable solution for the probabilistic analysis of WSNs especially when applied to validate WSNs for mission-critical applications like military, health, disaster relief and environmental monitoring.

In order to overcome the common drawbacks of simulation, formal methods [6] have been proposed as an efficient solution to validate a wide range of hardware and software systems. Formal methods increase the system reliability by rigorously using mathematical techniques to analyze the mathematical model for the given system. They have the advantage to find out subtle errors that cannot be revealed by traditional simulation. The need of formal methods in the context of WSNs is illustrated in [19]. However, formal methods seem very restricted when used to validate probabilistic systems. The random components of the system cannot be directly modeled within traditional formal tools. For example, it will be impossible to reason precisely about statistical properties, such as expectation and variance, in the case of state-based approaches. Furthermore, huge proof efforts are usually expected to be involved in reasoning about random components of a wireless system in the case of theorem proving.

Due to the recent developments in the formalization of probability theory concepts in higher-order-logic [12,7], the analysis of a variety of wireless systems with random components in a higher-order-logic theorem prover [5] can be handled with reasonable amount of proof efforts. In this paper, we propose to use the probabilistic framework developed in the HOL theorem prover [7] to formally analyze the coverage-based random scheduling algorithm of [18]. Due to the high expressiveness of the underlying logic and the inherent soundness of theorem proving, this framework overcomes the common limitations of probabilistic model checking, which are the state space explosion and the inaccuracy in the reasoning about statistical quantities. Particularly, we aim at verifying the expected values of coverage intensity, and deducing the upper bound on the total number of disjoint subsets, given expected coverage intensity for the given scheduling algorithm. We also verify the lower bound on the total number of nodes and the average detection delay inside the network.

The remainder of this paper is organized as follows. First, we discuss related work. Then, we present an overview of HOL probabilistic analysis foundations. Sections 4 and 5 provide the formal specification and verification of the coverage-based random scheduling algorithm, respectively. Finally, we conclude the paper.

2 Related Work

Due to its wide applicability, the random scheduling algorithm has been analyzed using various approaches in the open literature. The most commonly used approach is simulation, where a computer based mathematical model of the given algorithm is built and then evaluated through rigorous sampling. The simulation tools must essentially provide some probabilistic features in order to perform realistic simulations. In [18], a coverage-based random scheduling algorithm has been analyzed by a mathematical model, which coverage has been subsequently enhanced in [17] by eliminating some blind points. The evaluation of the two previous works within a Java simulator has restricted the monitored region to 200mx200m, the detection range to 10m, and the number of sub-networks to 6. Due to the inherent nature of simulation coupled with the usage of computer arithmetic, the probabilistic analysis results attained by the simulation approach can not be termed as completely accurate.

Probabilistic model checking is one of the first formal methods to be used for probabilistic analysis of wireless systems [22]. It has the same principle as traditional model checking: the mathematical model of the probabilistic system is exhaustively tested to check if it meets a set of probabilistic properties. This technique has been successfully used to validate many aspects of WSNs. The authors of [20] performed the formal analysis of the OGDC algorithm in the RT-Maude rewriting tool [21]. They have successfully analyzed the common performance metrics, such as, the network coverage intensity and lifetime. The probabilistic model checker PRISM [15] has also been used quite frequently for the verification of medium access control (MAC) protocols designed for WSNs, such as the S-MAC [1] and ECO-MAC [25] protocols. For the first protocol, the authors have verified, within PRISM, the reachability of packets to the sink node for a simple network model of 3-hops. They have also evaluated the expected communication latency and energy consumption of the model. Regarding the probabilistic model checking of ECO-MAC, it has especially verified properties related to the number of packet retransmissions.

In addition to its accuracy, the main advantage of probabilistic model checking method is its mechanization. However, it also suffers from some major shortcomings like the common problem of state space explosion [2] and the inability to reason accurately about statistical properties. For instance, during the verification of the OGDC [20] algorithm, the network model has been limited to 6 nodes on a surface of 15mx15m. Similarly, in [1], the network hops have been restricted to 3 and the number of scheduled subsets to 2 so that the built model can be accepted in PRISM. Finally, while verifying the ECO-MAC [25] protocol, the authors have been also obliged to readjust some parameters by a reduction factor in order to avoid a state explosion problem which was completely unpredictable. On the other hand, the reasoning support for statistical quantities in probabilistic model checker like PRISM is not so accurate. In [1], the authors have given expected values of communication latency and energy consumption by running several experiments on the proposed model of S-MAC. These values were specific to the chosen configuration and can not be considered as general

in any way. Another limitation of some classical model checkers trying to model probabilities can be also identified in [20], where the probability modeling was very approximative within the RT-Maude tool. The authors have just used a random function which is assumed to be 'good' to generate such behavior. For Uniform distributions, they have selected a sampling value generated by the same random function on a given interval. Such kind of analysis is not exhaustive and thus cannot be termed as formally verified.

In this paper, we overcome the limitations of both simulation and model checking techniques by using the probabilistic framework developed in the HOL theorem prover to validate a variant of the randomized scheduling of nodes in the context of WSNs. This framework, which is a theorem proving based probabilistic analysis framework, has already shown its practical effectiveness on a lot of case studies. Indeed, Hurd successfully verified the Miller-Rabin primality test; a well-known and commercially used probabilistic algorithm [13]. Hasan et al. verified the stop-and-wait protocol [9], a stuck-at fault model for reconfigurable memory arrays [8] and the automated repeat request (ARQ) mechanism at the logic link control (LLC) layer of the General Packet Radio Service (GPRS) standard for Global System for Mobile Communications (GSM) [10]. The HOL probabilistic framework is principally founded on Hurd's PhD thesis [12] where the formalization of some discrete random variables along with their verification, based on the corresponding PMF properties is presented [12]. In [7], Hurd's formalization framework has been extended with a formal definition of expectation. This definition is then utilized to formalize and verify the expectation and variance characteristics associated with discrete random variables that attain values in positive integers only. Statistical properties of continuous random variables have been also verified in [11]. To the best of our knowledge, none of the past works dealing with the random scheduling algorithm for WSNs or one of its variant has incorporated a formal probabilistic technique based on model checking or theorem proving.

3 Preliminaries

In this section, we describe the main theoretical elements upon which the probabilistic framework developed in the HOL theorem prover is built [7]. Particularly, we present the formalization of discrete random variables in HOL and the verified probabilistic properties that will be needed later. The general methodology that we have to follow for analyzing a wireless system within the probabilistic framework developed in the HOL theorem prover can be found in [10].

3.1 Formalization of Discrete Random Variables and Verification of their PMF

A random variable is called discrete if its range, i.e., the set of values that it can attain, is finite or at most countably infinite [23]. Discrete random variables are mathematically specified by their Probability Mass Functions (PMF) which is

the probability that a random variable X is exactly equal to some value x, i.e., $Pr(X = x)$. In higher-order-logic, discrete random variables are formalized as deterministic functions with access to an infinite Boolean sequence $\mathbf{B}^{\infty}$; a source of infinite random bits with data type $(num \rightarrow bool)$[12]. According to the result of popping the top most bit in the infinite Boolean sequence, these deterministic functions make random choices. They may pop as many random bits as they need for their computation. At the end of the computation, they return the result along with the remaining portion of the infinite Boolean sequence to be used by other functions. Thus, a random variable that takes a parameter of type α and ranges over values of type β can be represented in HOL by the function:

$$\mathcal{F} : \alpha \rightarrow B^{\infty} \rightarrow \beta \times B^{\infty}.$$

As an example, the $Bernoulli(\frac{1}{2})$ random variable that returns 1 or 0 with equal probability can be modeled as follows

```
⊢ bit = λs. (if shd s then 1 else 0, stl s).
```

where the variable s represents the infinite Boolean sequence and the functions `shd` and `stl` are the sequence equivalents of the list operation *'head'* and *'tail'*. The function `bit` accepts the infinite Boolean sequence and returns a pair with the first element equal to either 0 or 1 and the second element equal to the unused portion of the infinite Boolean sequence, which in this case is the tail of the sequence.

Random variables can also be expressed in a more compact form using the general state-transforming monad where the states are the infinite Boolean sequences.

```
⊢ ∀ a,s. unit a s = (a,s)
⊢ ∀ f,g,s. bind f g s = g (fst (f s)) (snd (f s)).
```

The HOL functions `fst` and `snd` above return the first and second components of a pair, respectively. The `unit` operator is used to lift values to the monad, and the `bind` is the monadic analogue of function application. All monad laws hold for this definition, and the notation allows us to write functions without explicitly mentioning the sequence that is passed around, e.g., function `bit` can be defined as

```
⊢ bit_monad = bind sdest (λb. if b then unit 1 else unit 0).
```

where, `sdest` gives the head and tail of a sequence `s` as a pair (*shd* `s`, *stl* `s`).

The measure theory formalization of [12] can be used to define a probability function `prob`, which transforms sets of infinite Boolean sequence to the set of real number between 0 and 1. The domain of `prob` is the set $\mathcal{E}$ of probability events. Consequently, the formalization of `prob` and $\mathcal{E}$ can be used together to prove probabilistic properties of random variables such as:

$$\vdash \texttt{prob}\ \{\texttt{s} \mid \texttt{fst (bit s)} = 1\} = \frac{1}{2}.$$

where the HOL function `fst` selects the first component of a pair and $\{x|C(x)\}$ represents a set of all elements x that satisfy the condition C.

By following the methodology described above, most of the commonly used discrete random variables which are frequently used have been specified in the HOL theorem prover. The corresponding PMF of each of these discrete random variables has been also verified. For example, HOL definitions and PMF theorems for the Bernoulli, Uniform, Binomial and Geometric random variables can be found in [12,7].

3.2 Formalization and Verification of Expectation Properties for Discrete Random Variables in HOL

The expectation of a discrete random variable, which attains values in the positive integers only, is specified as follows [16]:

$$Ex\_fn[f(R)] = \sum_{n=0}^{\infty} f(n)Pr(R = n). \tag{1}$$

where R is the discrete random variable and f represents a function of the random variable R. The function f maps the random variable R to a real value. The above definition of expectation holds only if the summation is well defined, i.e., finite. The above equation can be formalized in HOL as follows:

Definition 1

```
⊢ ∀ f R. expec_fn f R = suminf (λn.(f n)prob {s | (fst (R s)=n)}).
```

The HOL function `suminf` represents the infinite summation of a real sequence. The function `expec_fn` accepts two parameters, the function `f` of type ($num \rightarrow real$) and the positive integer valued random variable `R` and returns a real number.

The expectation of a discrete random variable that attains values in positive integers would be a particular case of the above definition where the function `f` is instantiated by the identity function (`λn.n`).

Definition 2

```
⊢ ∀ R. expec R = expec_fn (λn.n) R.
```

For illustration purposes, the formalization of expectation of a positive valued discrete random variable was used to verify the expectation of the Bernoulli, Uniform, Binomial and Geometric random variables [7]. It was also very interesting to check the correctness of some related properties, which greatly facilitates the theorem proving based probabilistic analysis. For example, the proof of the linearity of expectation, specified in (2), has been provided in [7].

$$Ex\_fn[af(R) + b] = aEx\_fn[f(R)] + b \tag{2}$$

4 Coverage-Based Randomized Scheduling Algorithm

According to the probabilistic framework, proposed in [10], the formal analysis of wireless systems is composed of two main steps, i.e., the formalization of the given wireless system, while modeling its random components by the formalized random variables, and using this model to formally verify properties of interest as higher-order-logic theorems. In this section, we develop a HOL formalization of the coverage-based random scheduling algorithm for WSNs, which corresponds to the first step outlined above. This formalization is basically inspired by the paper-and-pencil based analytical analysis presented in [18].

4.1 Overview of the Coverage-Based Randomized Scheduling Algorithm

We consider a WSN that deploys n sensors over a field of size a. All sensors have the same task; gathering data and routing it back to the base station. The deployment of nodes over the two-dimensional field is random and thus no location information is available. The size of the sensing area of each sensor is denoted by r. A sensor can only sense the environment and detect events within its sensing range. We say that a point of the monitored field is covered when any event occurring at this point can be detected by at least one active sensor. The probability q that each sensor covers a given point is r/a. The random scheduling of the nodes assigns each sensor to one of the k sub-networks with equal probability $1/k$. During a time slot Ti, only the nodes belonging to the sub-network i will be active and can cover an occurring event. Hence, the disjoint sub-networks created will work alternatively. We denote also by: Si, the set of sensors that belongs to the sub-network i and covers a specific point inside the field, S, the set of nodes covering a specific point inside the field, and, c, the cardinality of S.

For illustration purposes, Fig. 1 shows how the scheduling algorithm splits arbitrarily a network containing eight sensor nodes to two sub-networks. The eight nodes, randomly deployed in the monitored region, are identified by IDs ranging from 0 to 7. The two sub-networks are called *S0* and *S1*. Each node chooses at random between 0 and 1 in order to be assigned to one of these two

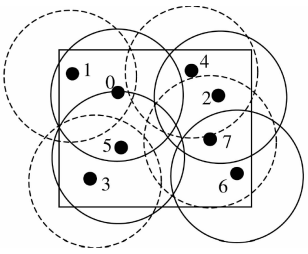

Fig. 1. An example of the randomized coverage-based algorithm [18]

sub-networks. Suppose that nodes 0; 2; 5; 6 select the number 0 and join the subset S0 and nodes 1; 3; 4; 7 choose the number 1 and join the subset *S1*. Thus, the two sub-networks will work alternatively. In other words, when the nodes 0; 2; 5; 6, which sensing ranges are denoted by the solid circles, are active, the nodes 1; 3; 4; 7 illustrated by the dashed circles will be idle and vice versa.

4.2 Formalization of the Network Coverage Intensity

The challenge in the random scheduling algorithm described below, is to select a value of k so that the energy can be saved with a good coverage. Therefore, the performance of this algorithm depends essentially on the chosen value of k. A large k will imply a lot of sub-networks which would in turn result in few nodes in each of these sub-networks, and hence a poor coverage. However, a small k will imply few sub-networks with a lot of points covered simultaneously by a lot of nodes, so a waste of energy.

The random scheduling algorithm involves several random variables. The first one distributes uniformly the nodes over the sub-networks. It is formalized by the HOL function `rd_subsets`:

Definition 3

```
⊢ (∀ k. rd_subsets 0 k = []) ∧
  (∀ c,k. rd_subsets (c+1) k = (prob_uniform k)::(rd_subsets c k)).
```

which generates recursively a list of Uniform random variables, and accepts two parameters: c, the number of sensors that covers a specific point inside the field, and k, the number of sub-networks. In this definition, we use the predefined HOL function `prob_uniform` which takes as input a natural k and generates a Uniform (k) random variable.

Let X be the random variable denoting the total number of non-empty subsets Sj. X is defined as follows:

$$X = \sum_{j=0}^{k-1} Xj. \tag{3}$$

where Xj is the Bernoulli random variable describing a non-empty subset. The variable Xj, expressed by the following HOL function, is based on the recursive HOL predicate `subset_empty` which describes an empty subset by looking for an index j in the list generated by the function `rd_subsets`.

Definition 4

```
⊢ ∀ j,c,k. subset_non_empty j c k = bernoulli_num
   (prob bern {s | fst (subset_empty j (rd_subsets c k) s) = F}).
```

The function `subset_non_empty` takes three parameters: j, a natural number, c, the number of sensors that covers a specific point inside the field, and k, the number of sub-networks. The set `{s | fst (subset_empty j (rd_subsets c k) s) = F}`, used in this function, formally models the set of events when the subset Sj is non-empty.

In order to define the random variable X, given in (3), we first define a function which recursively generates a list of Xj's by accepting the parameters: k, the length of the list, c, the number of sensors that covers a specific point inside the field, and m, the number of sub-networks. After that, a pre-defined function of the HOL probability theory, called `sum_rv_lst`, accepts this list of random variables and returns their sum as a single random variable.

Definition 5

```
⊢ (∀ c,m. subset_non_empty_lst 0 c m = [subset_non_empty 0 c m]) ∧
  (∀ k,c,m. subset_non_empty_lst k c m =
  (subset_non_empty (k+1) c m)::(subset_non_empty_lst k c m)).
```

The coverage intensity for a specific point Cp can now be defined as the average time during which the point is covered by the total length of the scheduling cycle.

$$Cp = \frac{E[X] \times T}{k \times T}. \tag{4}$$

where $E[X]$ denotes the expectation of the random variable X defined in (3). The variable Cp is formalized in HOL as follows:

Definition 6

```
⊢ ∀ c,k. cvrge_intsty_pt c k =
  (expec (sum_rv_lst (subset_non_empty_lst k c (k+1))))/(k+1).
```

The above definition specifies the coverage intensity for a specific point using the HOL function `cvrge_intsty_pt`. This function takes as parameters: c, the number of sensors that covers a specific point inside the field, and k, the number of sub-networks. Added to the function `subset_non_empty_lst`, this definition uses two other predefined HOL functions which are `expec`, for the expectation of a discrete random variable (Definition 2), and `sum_rv_lst`, for the summation over random variables. More details about these two functions can be found in the preliminaries section and in [7].

It has been shown in [18] that Cp is equal to:

$$\left[1 - \left(1 - \frac{1}{k}\right)^{c}\right]. \tag{5}$$

We recall that the variable c is initially the number of nodes covering a specific point inside the field. Covering a point or not can be assimilated to a Bernoulli trial with the probability q. If we consider the variable c among the n nodes of the network, it becomes a Binomial random variable with the following probability:

$$Pr(c = j) = \frac{n!}{j!\,(n-j)!} q^j (1-q)^{n-j}. \tag{6}$$

where q is the probability that each sensor covers a given point.

Thereafter, Cp is also a random variable. Particularly, Cp is a function of the random variable c. Since the random deployment strategy distributes independently the nodes over the area and the random scheduling makes a uniform

distribution of the same sensors, the expectation of Cp for any point inside the area is the same and its value is Cn. The variable Cn is defined as follows:

$$Cn = Ex\_fn[Cp] \tag{7}$$

where Ex_fn designates the expectation of a function of a random variable. The corresponding HOL function formalizing (7) is:

Definition 7

```
⊢ ∀ q,n,k. cvrge_intsty_network q n k =
  expec_fn (λx. 1 + (-1)×(1 - 1/(k+1))ˣ) (prob_binomial_p n q).
```

The above function `cvrge_intsty_network` accepts as inputs q, the probability that a sensor covers a point, n, the number of sensors deployed inside the field, and k, the number of sub-networks. This function specifies the expectation of a function of random variable and thus needs two parameters: the input function which basically describes the variable Cp and the random variable which is the Binomial of (6).

4.3 Formalization of the Average Detection Delay

The average detection delay is another performance metric which can be relevant in evaluating the random scheduling algorithm. It is defined as the expectation of the time elapsed from the occurrence of an event to the time when the event is detected by some sensor nodes. The average detection delay for an event arriving at any time slot with equal probability and lasting for duration longer than $(k-1) \times T$, is defined as:

$$delays = \sum_{i=1}^{k-1} \int_{0}^{T} \frac{1}{T} \times \Pr(H0 \cap H1 \cap ... \cap \overline{Hi}) \times (i \times T - t)dt. \tag{8}$$

where Hi is the event that none of the c covering sensor nodes belongs to the working subset i, $\overline{Hi}$ is the event that at least one of the c covering sensors belongs to the working subset i, T is the duration of a time slot, and k is the number of disjoint subsets.

Defining the HOL theorem corresponding to the verification of the average detection delay requires the formalization of the set $(H0 \cap H1 \cap ... \cap H(i-1) \cap \overline{Hi})$ as a higher-order-logic function. The proposed idea consists in dividing this set into two parts: the first one defines the intersection of the $(i-1)$ first events while the second models the event that *'the i^{th} working sub-network is non-empty within Ti'*.

The function `compl_intersection`, given in Definition 8, illustrates the first part of the required final set.

Definition 8

```
⊢ ∀ i,c,k. compl_intersection i c k =
          bind (indep_rv_list (subset_non_empty_rv_list i c k))
                                        (λx. unit (disj_list x)).
```

This builds the intersection of events describing the $(i-1)$ first empty subsets. The idea is to first make a list of the required random variables (function `subset_non_empty_rv_list`) by satisfying the independence criteria (function `indep_rv_list`), and then create the conjunction of all the elements of the list as required. The function `compl_intersection` takes as parameters: i, a natural index, c, the number of sensors that covers a specific point inside the field, and k, the number of sub-networks. The HOL definitions of the the two functions used within the function `compl_intersection` can be found in [4].

The second part of the final set is described by the Bernoulli random variable used in (3) which also expresses the event of an empty subset. Thus, the final set is described by the following HOL function `final_set` which takes the same parameters as the function `compl_intersection`.

Definition 9

```
⊢ ∀ i,c,k. final_set i c k =
                  bind (compl_intersection i c k) (λx.
                  bind (subset_non_empty (k-i-1) c (k-i)) (λy.
                  unit (¬x ∧ (y = 1)))).
```

5 Formal Verification of the Random Scheduling Algorithm

We use the defined HOL functions in order to formally verify the main statistical properties regarding the network coverage intensity and the average detection delay. We have described the verified theorems in a backward chaining approach, i.e., we present the main goal first and then the corresponding proofs.

5.1 Formal Verification of the Network Coverage Intensity

We have already noticed from the specification section that the network coverage intensity is defined as a statistical measure of the coverage intensity for a specific point (see (7)). Hence, we need to verify first that the coverage intensity for a specific point, defined in (4), is really equal to the expression given in (5). The HOL theorem corresponding to this property can be expressed as follows:

Theorem 1

```
⊢ ∀ c,k. cvrge_intsty_pt c k = 1 - (1 - (1/(k+1)))^c.
```

The verification of the above theorem is based on Theorem 2, which gives the expectation of the random variable specified in (3).

Theorem 2

```
⊢ ∀ c,k. expec (sum_rv_lst (subset_non_empty_lst k c (k+1))) =
         (k+1)×(1 - (1 - (1/(k+1)))^c).
```

The proof of Theorem 2 is mainly based on the application of the expectation property stating that the expectation of the sum of discrete random variables

is equal to the sum of their respective expectation, and the verification of the expectation of each element of the list `subset_non_empty_lst` [4].

Next, we have to verify the second main theorem related to the network coverage intensity Cn. It has been shown in [18] that Cn is equal to:

$$1 - \left(1 - \frac{q}{k}\right)^n. \tag{9}$$

which is formalized in HOL by the following theorem:

Theorem 3

```
⊢ ∀ n,q,k. (0 ≤ q) ∧ (q ≤ 1) ∧ (1 ≤ n) ⇒
           (cvrge_intsty_network q n k = (1 - (1-(q/(k+1)))ⁿ)).
```

The proof of Theorem 3 is primarily based on the application of the linearity of expectation property (see (2)) which further requires the independence of the Binomial random variable, already verified in [7], and the proof of the finite summation of the corresponding function multiplied by the probability. Besides that, the proof of Theorem 3 needed a lot of mathematical reasoning related to the real summation especially for the Binomial theorem for reals which was not available in the existing HOL libraries and thus, we had to prove it.

Theorem 3 gives a clear relationship between the network coverage intensity, the number of nodes n and the number of disjoint sub-networks k. As a result, two important corollaries can be deduced. Given a number k, we require that the minimum of the network coverage intensity Cn is t, and we can deduce the lower bound on the necessary number of sensor nodes in the whole network which is:

$$n \geq \left\lceil \frac{\ln(1-t)}{\ln\left(1 - \frac{q}{k}\right)} \right\rceil. \tag{10}$$

The above corollary has been successfully verified in HOL by using intermediate results associated to the two mathematical functions of power and logarithm.

Similarly, we can deduce that for a given n and providing a network coverage intensity of at least t, the upper bound on the number of disjoint subsets k is:

$$k \leq \frac{q}{1 - e^{\frac{\ln(1-t)}{n}}}. \tag{11}$$

The proof of the above corollary was straightforward and is based on pre-verified theorems from the two HOL theories of real and exponential.

The second corollary, given in (11), is very useful in dynamically adjusting the coverage of a sensor network after it is deployed. When the total number of sensor nodes is fixed, the network coverage intensity can be adjusted by changing the number of disjoint subsets k. A simple message flooding can be done to inform all sensor nodes about the new value of k.

5.2 Formal Verification of the Average Detection Delay

It has been shown in [18] that the average detection delay for an event, occurring at a point covered by c sensor nodes and lasting for duration longer than $(k-1) \times T$, is equal to:

$$delays = \frac{T}{2} \times \left[\left(\frac{k-1}{k} \right)^{c} + 2 \times \sum_{i=2}^{k-1} \left(\frac{k-i}{k} \right)^{c} \right]. \tag{12}$$

We have successfully verified the theorem formalizing the above equation. The proof has been based on an important result, verified in Theorem 4, along with some reasoning based on derivatives, and the corresponding details can be found in [4].

Theorem 4

```
⊢ ∀ i c k. (2 ≤ k) ∧ (1 ≤ (k - i)) ⇒
           (prob bern {s | fst (final_set i c k s) = T} =
           product 0 i (λj. (1-(1/(k-j)))ᶜ)×(1-(1-1/(k-i))ᶜ)).
```

This theorem reduces the probability of a set of independent events to the product of their respective probabilities. The function `product`, used in the above theorem, is a recursive function that gives the product of a sequence of elements of the same function. The proof of Theorem 4 required reasoning related to the transformation of probabilistic sets and to the independence theorem of probability. Under some assumptions, this last theorem transforms the probability of the intersection of two independent events into the product of their respective probabilities.

Our results demonstrate the effectiveness of the probabilistic theorem proving based approach for the verification of randomized scheduling algorithms for WSNs. We have been able to formally verify the most important probabilistic properties of interest associated with the network coverage intensity and the average detection delay. While other techniques, like simulation and model checking, are restricted by the number of simulated nodes n, the number of disjoint subsets k, the sensing range r, and the surface a, our results are completely generic, i.e., the verified theorems are universally quantified for all values of n, k, r and a.

Moreover, the inherent soundness of theorem proving certifies that the obtained results are 100% accurate. Based on the discussion in Sections 1 and 2 of this paper, it is clear that other techniques can never have this flexibility. Indeed, previous simulation work have given non-exhaustive results which are valid for specific network configurations. Similarly, probabilistic model checking have been frequently forced to restrict the values of the two first parameters in order to avoid a state space explosion problem. Finally, compared to probabilistic model checkers, a major novelty provided in this paper is the ability to perform formal and accurate reasoning about statistical properties of the problem. Hence, it was possible to verify the network coverage intensity which is a statistical measure of the coverage intensity for a specific point. This possibility is mainly due to the strong theoretical support for probability modeling available within the HOL probabilistic framework and the high expressibility of higher-order logic.

The above mentioned additional benefits, associated with the theorem proving approach, are attained at the cost of the time and effort spent, while formalizing

the randomized scheduling algorithm and formally reasoning about its properties, by the user. This analysis consumed approximately 200 man-hours and 1500 lines of HOL code by an expert user.

The major challenges faced in this work include the learning of the HOL probabilistic framework that primarily requires prior familiarization with the theorem proving technique and a good background on the probability theory. Higher-order-logic formalization also required a lot of intuition in selecting the right random variables. Similarly, an exhaustive set of assumptions is required for the verification as missing any assumption leads to verification failure due to the inherent soundness of the underlying theorem proving approach. Nevertheless, the fact that we were building on top of already verified probability theory related results helped significantly in this regard. In this paper, a lot of intermediate results have been omitted in order to meet page limits. The interested reader can refer to [4] for more details about all the theorems.

6 Conclusions

Due to the deployment constraints of WSNs, we are more motivated to provide algorithms characterized by a probabilistic behavior. Such a characteristic is impossible to cover using classic validation procedures like simulation, which do not ascertain 100% accuracy. The purpose of this paper was to provide a reliable analysis by using an accurate formal probabilistic reasoning based on the general purpose HOL theorem prover. We formally analyzed the coverage and the average detection delay of a scheduling algorithm designed for randomly deployed wireless sensor networks. We particularly verified the expected values of the coverage intensity, the upper bound on the total number of disjoint subsets, the lower bound on the total number of nodes and the average detection delay inside the network.

To the best of our knowledge, this paper presents the *first* formal analysis of a randomized scheduling problem using a probabilistic formal method. Obtained results have the advantages to be exhaustive and completely generic, i.e., valid for all parameter values, which cannot be attained in simulation or probabilistic model checking based approach. In addition, the successful formal reasoning about statistical properties clearly demonstrates the practical effectiveness of the proposed approach compared to probabilistic model checking, where such a feature is not available.

It is important to note that the usability of the HOL probabilistic framework for the WSN context is not limited to the current case study. Indeed, the whole framework can be efficiently used to formally analyze several probabilistic routing algorithms for WSNs. One such example is the Reverse Path Forwarding (RPF) algorithm [3]. Once the HOL probabilistic framework is enriched with possibilities to reason about statistical properties of multiple continuous random variables, it will be promising to extend the formal analysis of the coverage-based scheduling algorithm. We can, for example, think to formally verify the network lifetime which is a crucial aspect in the WSNs context or the impact of clock asynchrony on the coverage quality.

References

1. Ballarini, P., Miller, A.: Model Checking Medium Access Control for Sensor Networks. In: International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, pp. 255–262. IEEE Press, New York (2006)
2. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (2000)
3. Dalal, Y., Metcalfe, R.: Reverse Path Forwarding of Broadcast Packets. Commun. of ACM 21(12), 1040–1048 (1978)
4. Elleuch, M., Hasan, O., Tahar, S., Abid, M.: Formal Probabilistic Analysis of the Coverage-based Random Scheduling Algorithm for WSNs. Technical Report. ENIS, Sfax University, Tunisia (2011), `http://www.ceslab.org/publications/TR_FPARSAWSN_v1.3.pdf`
5. Gordon, M.J.C.: Mechanizing Programming Logics in Higher-Order Logic. In: Current Trends in Hardware Verification and Automated Theorem Proving, pp. 387–439. Springer, Heidelberg (1989)
6. Gupta, A.: Formal Hardware Verification Methods: A Survey. Formal Methods in System Design 1(2-3), 151–238 (1992)
7. Hasan, O.: Formal Probabilistic Analysis using Theorem Proving. PhD Thesis, Concordia University, Montreal, QC, Canada (2008)
8. Hasan, O., Tahar, S., Abbasi, N.: Formal Reliability Analysis using Theorem Proving. IEEE Transactions on Computers 59(5), 579–592 (2010)
9. Hasan, O., Tahar, S.: Performance Analysis and Functional Verification of the Stop-and-Wait Protocol in HOL. Journal of Automated Reasoning 42(1), 1–33 (2009)
10. Hasan, O., Tahar, S.: Probabilistic Analysis of Wireless Systems using Theorem Proving. Electronic Notes in Theoretical Computer Science 242(2), 43–58 (2009)
11. Hasan, O., Abbasi, N., Akbarpour, B., Tahar, S., Akbarpour, R.: Formal Reasoning about Expectation Properties for Continuous Random Variables. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 435–450. Springer, Heidelberg (2009)
12. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, Cambridge, UK (2002)
13. Hurd, J.: Verification of the Miller-Rabin Probabilistic Primality Test. J. of Logic and Algebraic Programming 50(1-2), 3–21 (2003)
14. Jain, S., Srivastava, S.: A Survey and Classification of Distributed Scheduling Algorithms for Sensor Networks. In: International Conference on Sensor Technologies and Applications, pp. 88–93. IEEE Press, New York (2007)
15. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: Probabilistic Model Checking for Performance and Reliability Analysis. ACM SIGMETRICS Performance Evaluation Review 36(4), 40–45 (2009)
16. Levine, A.: Theory of Probability. Addison-Wesley series in Behavioral Science, Quantitative Methods. Addison-Wesley, Reading (1971)
17. Lin, J.W., Chen, Y.T.: Improving the Coverage of Randomized Scheduling in Wireless Sensor Networks. IEEE Transactions on Wireless Communications 7(12), 4807–4812 (2008)
18. Liu, C., Wu, K., Xiao, Y., Sun, B.: Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 17(6), 562–575 (2010)

19. McIver, A.K., Fehnker, A.: Formal Techniques for the Analysis of Wireless Networks. In: International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, pp. 263–270. IEEE Computer Society, Washington, DC, USA (2006)
20. Ölveczky, P.C., Thorvaldsen, S.: Formal Modeling and Analysis of the OGDC Wireless Sensor Network Algorithm in Real-Time Maude. In: Bonsangue, M.M., Johnsen, E.B. (eds.) FMOODS 2007. LNCS, vol. 4468, pp. 122–140. Springer, Heidelberg (2007)
21. The Real-Time website, `http://heim.ifi.uio.no/peterol/RealTimeMaude/`
22. Rutten, J., Kwaiatkowska, M., Normal, G., Parker, D.: Mathematical Techniques for Analyzing Concurrent and Probabilisitc Systems. CRM Monograph Series, vol. 23. American Mathematical Society, Providence (2004)
23. Yates, R.D., Goodman, D.J.: Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers. Wiley, Chichester (2005)
24. Yick, J., Mukherjee, B., Ghosal, D.: Wireless Sensor Network Survey. J. Computer Networks 52, 2292–2330 (2008)
25. Zayani, H., Barkaoui, K., Ben Ayed, R.: Probabilistic Verification and Evaluation of Backoff Procedure of the WSN ECo-MAC Protocol. J. of Wireless & Mobile Networks 2(2), 156–170 (2010)