Formal Verification of Real-Time Wireless Sensor Networks Protocols with Realistic Radio Links

Alexandre Mouradian, Isabelle Augé-Blum

Formal Verification of Real-Time Wireless Sensor Networks Protocols with Realistic Radio Links*

Alexandre Mouradian and Isabelle Augé-Blum
Université de Lyon, INRIA, INSA-Lyon, CITI, F-69621, France
firstname.lastname@insa-lyon.fr

ABSTRACT

Many critical applications which rely on Wireless Sensor Networks (WSNs) are proposed. Forest fire detection, landslide detection and intrusion detection are some examples. Critical applications require correct behavior, reliability, and the respect of time constraints. Otherwise, if they fail, consequences on human life and the environment could be catastrophic. For this reason, the WSN protocols used in these applications must be formally verified. Unfortunately the radio link is unreliable, it is thus difficult to give hard guarantees on the temporal behavior of the protocols (on wired systems the link error probability is very low [7], so they are considered reliable). Indeed, a message may experience a very high number of retransmissions. The temporal guarantee has thus to be given with a probability that it is achieved. This probability must meet the requirements of the application.

Network protocols have been successfully verified on a given network topology without taking into account unreliable links. Nevertheless, the probabilistic nature of radio links may change the topology (links which appear and disappear). Thus instead of a single topology we have a set of possible topologies, each topology having a probability to exist. In this paper, we propose a method that produces the set of topologies, checks the property on every topology, and gives the probability that the property is verified. This technique is independent from the verification technique, i.e. each topology can be verified using any formal method which can give a "yes" or "no" answer to the question: "Does the model of the protocol respect the property?".

In this paper we apply this method on f-MAC [23] protocol. F-MAC is a real-time medium access protocol for WSNs. We use UPPAAL model checker [10] as verification tool. We perform simulations to observe the difference between average and worst case behaviors.

*This work has been partially founded by French Agence Nationale de la Recherche under contract VERSO 2009-017.

RTNS 2013 , October 16 - 18 2013, Sophia Antipolis, France

http://dx.doi.org/10.1145/2516821.2516833.

1. INTRODUCTION

A Wireless Sensor Network (WSN) is composed of nodes deployed in an area in order to monitor parameters of the environment. Those nodes are able to send information to dedicated nodes called sinks, in a multi-hop fashion, without the need of a fixed network infrastructure. Every node is able to forward messages from other nodes. They usually run on batteries so they should consume as little energy as possible in order to increase the network lifetime. Because WSNs can contain many nodes, the financial cost of a node should be as low as possible, this leads to design nodes with poor capabilities (computation, radio, memory, etc...). For these reasons, network protocols have been designed mainly in order to reduce energy consumption and to provide autonomous network mechanisms. Nevertheless some applications need more than these characteristics. Indeed, critical applications require reliability and the respect of time constraints. Forest fire detection [30], landslide detection [22] and volcano monitoring [27] are some examples. If these applications fail, consequences on human life and the environment could be catastrophic. There is thus a need to formally verify that the WSN protocols used by these applications meet their requirements.

WSNs radio links are unreliable, thus such formal hard guarantees cannot be achieved. This is because a message may never be correctly received (even if this event has a low probability). The guarantee thus has to be given with a probability that it is verified. In the literature, formal techniques to verify probabilistic systems have been proposed [19] [13]. Nevertheless, they are some issues to adapt them to the case of wireless networks, mainly because of restrictions on the languages (lack of data types). These restrictions make the modeling of important aspects of the protocol very difficult and lead to abstract important parts of the modeled protocols.

Nonetheless, protocols have been successfully validated without tacking into account probabilistic behaviors resulting from unreliable links [29] [6]. In this paper, we propose to use such techniques and to go further by taking into account the error probability of the radio link and thus give the probability that a temporal property of the protocol is satisfied. This probability must meet the requirements of the application, otherwise the system must be changed to increase it.

We propose a method to derive the probability that a time or correctness property is satisfied from the radio channel model. From this radio model, we obtain the probability that the transmission of a packet between two nodes is suc-

cessful. The network topology is represented by a graph, we put edges between nodes that can communicate. Nodes can communicate if transmissions between them are successful, we can thus compute the probability that a given topology exists using the probabilities that transmissions on each edge are successful. We generate every possible topologies and their probabilities from a set of node positions. On each topology the time property or correctness property is checked. If the property is verified, the probability of the topology is added to the probability that the property is verified. After checking all topologies, the probability to be in a case in which the property is verified is obtained. The formal method used must give a "yes" or "no" answer to the question: "Does the model of the protocol (including the topology) respect the property?".

This method is the main contribution of the paper. It has the advantage to be independent from the formal method used to check the time or correctness property. It allows the user to choose the method that fits the best to his/her case. This method allows to find topologies where the protocol under study is faulty and it also gives the probability that these topologies appears. The user can choose to fix the more probable cases according to the level of reliability required.

We apply it to verify a time property on a real-time MAC protocol: f-MAC [23]. We also simulate f-MAC in order to show that the probability derived from the verification method is relevant.

Section 2 gives an overview of the proposed method. In section 3, the propagation model is described, and the probability that a packet is correctly received as a function of the distance between emitter and receiver is derived from this model. In section 4, the generation of the topologies and their associated probability is presented. The whole verification process is detailed in section 5. A case study on the f-MAC protocol is presented in section 6: we detail the f-MAC protocol, we apply the verification method, we compare the results to simulation ones, and we discuss the advantages and limitations of the proposed method. In section 7, we present related works. In section 8, we conclude and give perspectives.

2. METHOD OVERVIEW

The main contribution of this paper is a method to give a probability that a WSN protocol respects a property. The method we propose is divided into three main steps:

1. The probability to receive correctly a packet is derived from a realistic propagation model for WSNs.

2. The set of possible topologies is produced and a probability (computed using results of the previous step) is associated to each topology.

3. For each topology, the property is checked on the model of the protocol under study (including the current topology). If the property is verified on the current topology, its associated probability (calculated in previous step) is added to the probability that the property is verified.

   After checking all the topologies of the set, the output of the method is the probability that the protocol respects the checked property.

The following sections (3, 4 and 5) detail these steps. In section 6, we apply the method to a WSN protocol.

3. PROPAGATION MODEL

In this section, we present how to compute the probability that a packet can be correctly received between two nodes (without and with retransmissions) as a function of the distance between the emitter and the receiver. We show how this probability can be derived from the propagation model. It is the first step of the proposed method. The results of this section are necessary to compute the probability of a topology.

It is very important, for our method to be accurate, to take a realistic propagation model. The log-normal shadowing model is widely used in order to model propagation channel in the case of WSNs. Authors of [31] advocate it provides a realistic propagation model for this type of networks. With this model, the signal transmitted through the wireless channel is not only attenuated by the distance between the emitter and receiver, but it also experiences random attenuation coming from changes in the environment (moving objects). The log-normal model is defined by the following path loss formula (in dB):

$$PL_{dB}(d) = PL_{dB}(d_0) + 10\alpha \log_{10}(\frac{d}{d_0}) + X_\sigma \qquad (1)$$

Where d_0 is a reference distance, α is the path loss exponent, d is the distance between the emitter and the receiver, and X_σ is a Gaussian variable in dB with mean zero and standard deviation σ. This Gaussian variable models the randomness coming from the environment.

3.1 Link probability

From the propagation model and the modulation scheme, we can derive the probability that a packet is correctly received:

$$P_{cr}(d) = 1 - PER \qquad (2)$$

with PER the Packet Error Rate (the number of incorrectly received packets divided by the number of received packets) and d the emitter-receiver distance. The PER formulation can be found in [11], it increases with the emitter-receiver distance. Figure 1 is a plot of Equation 2 with the parameters described in Table 1, $P_{cr}(d)$ decreases when the emitter-receiver distance increases (at constant transmission power).

In the remainder of this paper we use the values of Table 1 to compute $P_{cr}(d)$. N is the packet size used in the computation of the PER. σ, α and f are used to determine the path loss. N_0 and B allow to obtain the noise level at the receiver.

3.2 Retransmissions

Often, protocols use retransmission mechanisms in order to increase communications reliability. In order to take into account such behaviors we give the probability that a packet

Table 1: Parameters of the propagation model

Symbol	Description	Value
N	Size of the packet in bits	800
σ	Standard deviation in dB	4
α	Path loss exponent	2
N_0	Noise level in dBm/Hz	-154
f	Frequency of the carrier in MHz	868
B	Bandwidth in kbps	500

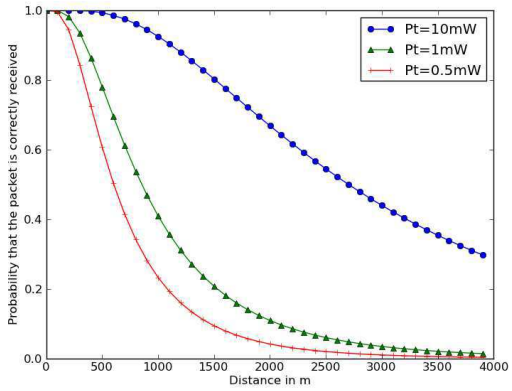

Figure 1: Probability that a packet is correctly received in function of the distance

is correctly received after K retransmissions:

$$\begin{aligned} P_{cr\_ret}(d) &= P_{cr}(d)\sum_{i=0}^{K}(1-P_{cr}(d))^i \\ &= P_{cr}(d)\frac{(1-P_{cr}(d))^{K+1}-1}{(1-P_{cr}(d))-1} \\ &= 1-(1-P_{cr}(d))^{K+1} \end{aligned} \quad (3)$$

From the expressions given in this section we can compute the probability that a link exists between two nodes, i.e. that a packet can be correctly transmitted (Equations 2 and 3). In the following section, we describe the second step: the probability assignment to topologies.

In this section we choose to derive the probability that a packet is correctly received from the log-normal shadowing model because it is widely used in the literature [31] for modeling WSNs. Nevertheless, this probability can be derive from any other model and then used in the method we propose and evaluate in the remainder of this paper.

4. TOPOLOGY PROBABILITY

In this section we describe how topology probabilities are derived from the results of the previous section. A topology is defined by a graph $G = (V, E)$ with V a set of vertices and $E \subseteq V \times V$ a set of edges. To each vertex is associated a location (x, y) on a plane. Each edge can be marked as "active" or "non-active" and has a probability associated. An active edge corresponds to a successful communication between vertices linked by the edge (the associated probability being given by Equation 2 and 3 respectively with and without retransmissions). We can notice that our method does not take into account asymetric links which can appear in reality in WSN [9], this issue will be addressed in a future work. Notations used in this section can be found in Table 2.

Topologies are derived from a basis topology, which is a graph with all edges marked as active. From the basis topology, a set of possible topologies is generated. This is done by marking a subset of edges of the basis graph as "active" and others as "non-active". Every possible graph is generated, so this produces $2^{|E|}$ graphs. Indeed, each edge of the basis topology can be "active" or not. The probability P_{E_i}, associated to the edge i, depends on the distance between the two vertices it links, and also on the number of retransmissions defined by the protocol under study. It is computed with Equation 3.

Based on P_{E_i}, a probability P_{topo} is assigned to each topology. For a given topology, if the edge E_i is active, $P_{E_i} = P_{cr\_ret}(d_i)$, and if it is not active, $P_{E_i} = 1 - P_{cr\_ret}(d_i)$, with d_i the distance between the vertices linked by edge E_i. The probability of the topology j is given by:

$$P_{topo}(j) = \prod_{i}^{|E|} P_{E_i} \quad (4)$$

The probabilities of topologies respect the property:

$$\sum_{j}^{2^{|E|}} P_{topo}(j) = 1 \quad (5)$$

A simple proof of Equation 5 can be constructed by taking an edge E_m. E_m can be active with probability $P_{cr\_ret}(d_m)$ and non-active with probability $1 - P_{cr\_ret}(d_m)$. Let A be a set of topologies without the edge E_m and S the sum of topologies probabilities for A. Now we want to add E_m to every topologies of the set A, we want to add the active version and the inactive version. It thus doubles the number of topologies in A and the new sum of topologies probabilities is:

$$\begin{aligned} S' &= \sum_{j\in A}^{2^{|E|}} P_{topo}(j) \\ &= P_{cr\_ret}(d_m)\cdot S + (1-P_{cr\_ret}(d_m))\cdot S \\ &= S \end{aligned} \quad (6)$$

Equation 6 means that the sum of probabilities taking into account edge E_m is equal to the sum of probabilities without E_m. If we apply this to every edge, Equation 5 follows.

Figure 2 depicts an example with three nodes and two links. On Figure 2, an edge is active if there is a link between the vertices. Topology 1 is the basis topology. For example, for topology 3, $P_{E_1} = P_{cr\_ret}(d(V_1, V_2))$ and $P_{E_2} = 1 - P_{cr\_ret}(d(V_2, V_3))$ and according to Equation 4, $P_{topo}(3) = P_{E_1} \times P_{E_2}$.

To be exhaustive, the basis topology should be a clique (i.e. each node is connected to all nodes). But in this case, we would consider topologies with a very low probability of actually being observed in reality. We thus consider a probability threshold under which a link is not considered (the threshold is arbitrary and depends on the protocol under study and the wanted accuracy). The probability threshold corresponds to a distance threshold (because d is a parameter of P_{cr}), thus basis topology can be produced using a

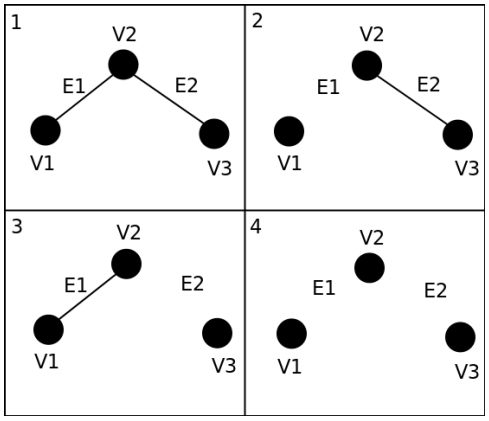

Figure 2: Example with 3 nodes and 2 links

Table 2: Notations

Symbol	Description
T	Set of topologies
T'	Set of topologies to be checked
M	Protocol model
G	Topology graph
E	Set of edges
V	Set of vertices
p	Property
P_{pv}	Probability that the property is verified
$P_{topo}(j)$	Probability of topology j

Unit Disk Graph model [5].

In this section we give the probability of a topology based on the probability that a packet (of size N) is correctly received on each "active" link (or edge) of the topology and not received on each "non-active" link with a given number of retransmissions. It thus represents the network during a limited amount of time. In the next section, we describe how to use these probabilities to give a probability that a property of a WSN protocol is verified.

5. VERIFICATION

In this section, we describe the proposed verification process and we discuss the applicability scope of such a method.

5.1 Description of the method

From the topologies and their associated probabilities, presented in the previous section, the verification process derives the probability that a property of a WSN protocol is verified. Notations used in this section can be found in Table 2.

Protocols usually have repetitive behaviors. They perform a sequence of tasks during a limited amount of time and repeat it several times during the whole network life time. In the case of WSNs, this repetitive behavior is often bounded by the duty cycle length. The duty cycle mechanism is used to reduce energy consumption, it consists in nodes alternately turning off and on their radio (respectively sleep and active modes) [21]. From the previous section, we deduce the probabilities of possible topologies for a limited amount of time (which can be the duration of the duty cycle for example). It is thus interesting to observe if a certain property is respected during a duty cycle (one active period plus one sleep period) with an associated probability. For example, one might want to verify that "with probability p all the packets do one hop toward the sink during a duty cycle".

Input: topology graph $G = (V, E)$
Output: T, the set of generated topologies

```
init T;
for E_i ∈ E marked as "active" do
    mark E_i as "non-active";
    if G ∉ T then
        add G to T;
        Generate_topologies(G);
    end
    mark E_i as "active";
end
```

Algorithm 1: Generate_topologies(G)

Based on the behavior of a protocol during one duty cycle and the positions of the nodes (from which is constructed the basis topology), we propose to generate topologies and compute their probabilities. The set of all the possible topologies is generated by the recursive procedure described by Algorithm 1. The probabilities of the topologies are calculated according to Equation 4. For each topology the property is checked. If the property is verified on a topology, the probability of this topology is added to the probability that the property is verified, P_{pv}. We can notice that, according to Equation 5, if the property is verified on all the topologies, the property is verified with probability 1 ($P_{pv} = 1$). The pseudo-code Algorithm 2 describes this verification process.

Input: basis topology G, protocol model M, property p
Output: probability P_{pv} that p holds

```
Generate_topologies(G);
Compute associated probabilities;
Select topologies to be checked T';
for T_j ∈ T' do
    Check p on M with T_j;
    if p holds then
        P_pv ← P_pv + P_topo(j);
    end
end
```

Algorithm 2: Verification process

With this process, the property is checked for all the topologies. As mentioned in the previous section, there are $2^{|E|}$ topologies generated from the basis topology. When the number of links increases the number of cases to be verified grows exponentially and so does the duration of the verification. The verification thus become unfeasible in reasonable time. In order to tackle this issue, we propose to preselect topologies for which the property has a chance to be verified, and to discard the topologies where the property cannot be verified. For example, if the property is that "the packet of each node must reach the sink in less than 5s" then we can discard all disconnected topologies. Indeed, packets from nodes which are not in the sink component have 0 probability to reach it because there is no path from them to the sink (on Figure 2, only topology 1 would be checked because 2, 3 and 4 are disconnected). For each case to be checked,

the topology is integrated in the model of the network.

The checking method is let to the user. Given the property and the model (including the topology) the checker has to answer "yes" or "no" to the question: "Does the model respect the property?". If the checking method is automatic (like model checking), the whole process can be automatized.

The proposed method allows to find out what are the topologies that the protocol under verification cannot handle (for example, the hidden terminal problem: two nodes, not connected, transmitting to the same node at the same time). It also gives the probability of occurrence of these bad cases. From this information the protocol designer can choose to fix the protocol or not, depending on the level of reliability needed, knowing that fixing the protocol can have a cost in term of energy consumption or end-to-end delay.

5.2 Applicability scope of the method

As mentioned in the previous subsection, the probability of a topology is only valid for a limited number of retransmission on one link (we propose to limit it to the number of retransmissions during a duty cycle). It leads to ask, if the method can capture the sequential nature of communications in WSNs, and can be applied to routing protocols. To overcome this apparent difficulty we propose to check models of routing protocols in two stages. The first is to compute, from the traffic model and the topology of the network, the worst case congestion at each node (the maximum number of packets a node will have in its buffer). Then, the second stage is to check the property: "In the worst case congestion of each node, every packet is able to do at least one hop toward the sink during one duty cycle". The fact that each packet gets closer to the sink in bounded time must be ensured by the routing protocol (if the access to the medium and the path length are bounded, then the end-to-end delay is also bounded). Our method is thus applicable to both MAC and routing protocols, but the property checked must remain in a time scope that corresponds to the number of retransmission chosen for the computation of links probability (we propose to use the duty cycle).

6. CASE STUDY: TEMPORAL VERIFICATION OF F-MAC

In this section we use the presented method to formally verify f-MAC, a real-time protocol for WSNs. We describe the protocol, we apply the verification process and we compare the results with simulation results.

6.1 f-MAC

In this section we present the f-MAC protocol. We choose the f-MAC protocol because it allows real-time medium access, the authors of [23] prove this property with perfect links. It is thus interesting to verify f-MAC behavior with unreliable links.

It is an asynchronous deterministic MAC protocol for WSNs. No time synchronization among nodes is needed, access to the medium in bounded time is guaranteed, and the bound is known. It uses the framelet approach: the data packet is transmitted several times periodically, each occurrence of the packet is called a framelet. In the case of f-MAC, the retransmissions do not serve reliability purpose, but they are used to handle the collisions. In order to guarantee that a node can access the medium, the periods of the emission of the framelets must be chosen carefully.

The authors define 4 rules which must be respected when configuring nodes:

1. Framelet length is define as: $d = \delta/2$ where δ is the f-MAC base unit.
2. The number of framelets is: $r = N_n$ with N_n the number of nodes.
3. The framelet period of node i, $t_i = k_i \times \delta$ must satisfy:
   $$k_i \cdot (r-1) < LCM(k_i, k_j) \quad \forall k_i < k_j \quad 1 \leq i, j \leq N_n$$
   with $LCM(k_i, k_j)$ the least common multiple of k_i and k_j
4. Nodes must wait for $t' = (k_{max} \cdot (r-1) + 1).\delta$ after the last framelet before sending another packet (with $k_{max} = \max_{1 \leq i \leq N_n}\{k_i\}$).

From these rules it can be deduced that the worst case transmission time for node i, noted T_i, is:

$$T_i = (r-1) \cdot t_i + t' \tag{7}$$

In fact, the authors prove that, by following these rules in the choice of the framelets periods, it is ensured that at least one framelet of each node does not collide with any other framelet. The worst case transmission time for all nodes is T_{max} deduced from Equation 7, $T_{max} = \max_{1 \leq i \leq N_n}\{T_i\}$.

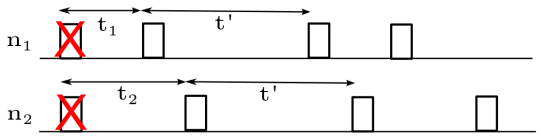

Figure 3: f-MAC example with 2 nodes

Figure 3 represents an example with two nodes. In this case, rule number 2 imposes to have two framelets to send one packet. In this example each node has two packets to send. The first framelet of the first packet is lost due to a collision but the second is correctly received because nodes n_1 and n_2 have different periods. At least one framelet of each packet is thus actually received. For the second packet, the two framelets of each node are correctly received.

F-MAC does not use a duty cycle. We thus propose to verify the protocol behavior during the maximum duration of a packet transmission, expressed by Equation 7 (instead of the behavior during a duty cycle as proposed in the previous section).

F-MAC is, up to our knowledge the only deterministic MAC WSN protocol that does not require time synchronization. We formally validate f-MAC using the proposed technique along with UPPAAL.

We produce two test basis topologies in which all the nodes are able to communicate with each others (with a given probability that depends on the distance between them). The basis topology A is represented on Figure 4 (we omit links for clarity) with 8 nodes. The coordinates of the nodes are generated randomly with a maximum distance between two nodes of 500 meters for basis topology A. Basis topology B is depicted on Figure 5, in this case the maximum distance between two nodes is 1000 meters.

Table 3: Verification results

Number of nodes	Number of topologies	Number of verified topologies	Average number of states	Duration (s)		P_{pv}	
				A	B	A	B
3	8	2	74.5	0.30	0.21	0.931	0.746
4	64	8	934	0.32	0.33	0.918	0.368
5	1024	64	10753	15.11	15.22	0.875	0.323
6	1073741824	1024	149546.6	69066.86	69173.25	0.771	0.268

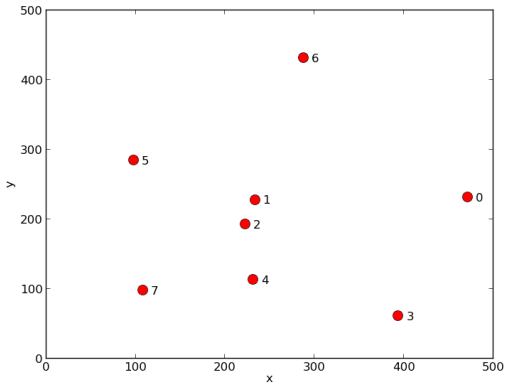

Figure 4: Basis topology A, 500m maximum

Node 0 is the sink node. For the verifications and the simulations, we use these test topologies but with node number ranging from 3 to 8 (by keeping only the first nodes). For example, topology A-3 is topology A only with nodes 0, 1, and 2.

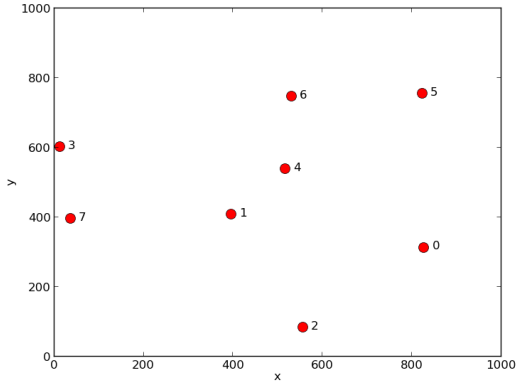

Figure 5: Basis topology B, 1000m maximum

6.2 Verification

In this section, we apply the validation process described in section 5 to the f-MAC protocol with the topologies described in the previous section.

For the verifications we use the UPPAAL model checker [10]. UPPAAL allows to check timed properties on system models. The modeling language is Timed Safety Automata [4] (referred as TA) extended with data types (notably integer and arrays of integers). Several TA can be parallely composed, forming a Network of TA. UPPAAL has been successfully used to check properties of wireless network protocols [6] [28] [29], but without taking into account the unreliable radio link. Usually the nodes are modeled as TA and the synchronizations between nodes through communication channels allow to model communications. We choose to use model checking because it is a fully automatic process, we can thus automatize the whole verification method.

In our scenario, each node has a packet to send to the sink. We verify the property that "all the packets (at least one framelet) are received by the sink before T_{max} (deduced from Equation 7)".

In the case of f-MAC, the retransmissions are not used for reliability purpose. They aim at guaranteeing a deterministic access to the medium. Thus, the retransmissions must not be taken into account in the calculation of $P_{cr}(d)$. The probability of a link is thus given by Equation 2 with the parameters of Table 1. By not taking into account the retransmissions in the calculation of the probability we assume that every framelet but one collides. This corresponds to the worst possible case. The probability obtained from the verification process is thus the minimum probability that the property is verified.

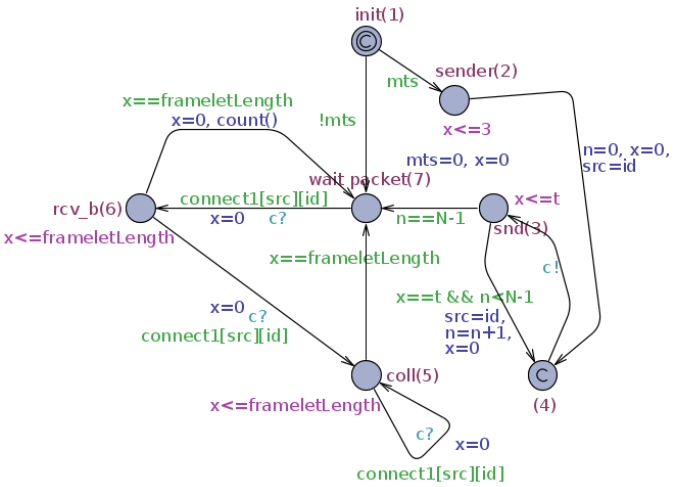

Figure 6: UPPAAL model of f-MAC

Figure 6 depicts the UPPAAL model of a f-MAC node. From the *init(1)* location, nodes with a message to send go to the *sender(2)* location and others go to *wait packet(7)*

location depending on the *mts* variable (meaning *message to send*). From the *sender(2)* location, a sender can take the outgoing transition between 0 and 3 time units. This allows to desynchronize the nodes to verify the behavior of f-MAC without synchronization. Nevertheless, this increases the state space size and leads to longer verification process. Then, senders loop between *snd(3)* and *(4)* locations until N_n framelets are sent. The period of the framelets is controlled with the invariant $x <= t$ and the guard $x == t$. The emission of a message is modeled by sending a synchronization signal on the c broadcast channel (noted $c!$). In our scenarios each node has only one packet to send, thus, after N_n framelets the node goes to *wait packet*. The sink is node 0, from the *init(1)* location, it goes directly to the *wait packet(7)* location. Nodes in *wait packet(7)* synchronize with senders only if they are connected (this behavior is called local broadcast), this is ensured by the guard *connect[src][id]*. More information on modeling local broadcast can be found in [18]. If the node is connected to the sender, it goes to location *rcv_b(6)*. It stays in this location until the end of the current framelet or until it receives another framelet. If it receives another framelet it means a collision occurred, thus the packet is not received and the node goes to *coll(5)* state. It stays in this state until the second framelet finishes, if another arrives before it loop back in the same state and wait another framelet duration. And so on so forth, until there is no more collision. It then goes back to state *wait packet(7)*. If no collision happened before the end of the current framelet, the packet is correctly received. The *count()* function counts the number of correctly received packets.

In this model the topology information is represented with an adjacency matrix. The *connect* array is a symmetric boolean array, *connect[i][j]=1* if i is a neighbor of j. We developed a program which implements Algorithm 2. It generates the topologies and it selects those which have to be verified. For each topology to be verified, it integrates the adjacency matrix in the UPPAAL network model.

Table 3 presents the results of the verification process for basis topologies A and B. The number of generated topologies, the number of verified topologies, the average number of states stored during model checking, and the duration of the verification increase exponentially with the number of nodes. The probability that the property is verified decreases when the number of nodes increases, because when there are more packets to transmit it is less probable that all of them are correctly received by the sink. We were able to perform verification on topologies of up to 6 nodes, this issue is discussed in section 6.3. For topology B, the probability P_{pv} lowers rapidly between 3 and 4 nodes. This is due to the fact that node 3 is really further from the sink than nodes 1 and 2 in topology B (as can be seen on Figure 5), and the probability that the packet is correctly received depends on the distance between the emitter and the receiver. For topology A, it occurs between 5 and 6 nodes (but not as important as topology B), because node 5 is further from the sink than the others.

The f-MAC protocol respected the property for all topologies checked (i.e. topologies where all the nodes are connected to the sink). It means that, as expected, f-MAC behavior is correct (for example it is not affected by the hidden terminal problem). Thus, if the links were reliable, the probability for f-MAC to respect the property would be equal to 1.

These results allow to conclude that the protocol has a safe behavior with probability at least P_{pv}. It is useful during the system design process. Indeed, if the probability is high enough the system can be implemented, otherwise the design has to be modified.

6.3 Discussion

In this section we discuss the advantages and drawbacks of the verification method we propose. The main limit of this solution is the scalability. The number of nodes, for verification, ranges from 3 to 6 (not 8) because the verification duration explodes for networks of more than 6 nodes (for 7, 1073741824 cases would have to be checked, it would require several weeks of calculation). The number of cases to be verified is high even when preselecting cases as mentioned in section 5. Nevertheless if the verification technique is very efficient, this problem can be mitigated. Improvements of the way to model the protocol should also reduce the checking duration. Another solution to tackle the scalability problem is to verify the topologies with high probability first. If the required reliability is achieved, the process can stop after few verifications. Otherwise, if a case with high probability does not respect the property we may be able to conclude that it is not possible to achieve the required reliability. We can also notice that scalability is an issue for most of the existing verification techniques. Despite this issue, the technique could be successfully used to verify Body Sensor Networks (BSNs) protocols. Because such networks are usually composed of few nodes (less than 10, usually 5 or 6 [3]) and medical applications require reliability and the respect of time constraints.

With the presented method, the verification has to give a "yes" or "no" answer, the verified protocol thus have to be deterministic. It could be interesting to see if it is possible to compose the probability coming from the radio channel and the one coming from the protocol operation.

The main advantage of our solution is to integrate reliability information with any verification method. In this paper, we use model checking to apply our method, but one can imagine use other methods such as real-time calculus or schedulability analysis. Indeed, the operation of a MAC protocol on a given topology can be modeled as a scheduling problem (schedule of the packets on the medium) for example.

Moreover, our method is also a very useful tool for protocol designers, to find design flaws and the probability of their occurrence. It allows the designers to make decisions on which flaw has to be fixed in priority in function of the probability. If no flaw is detected (as in the case of f-MAC), the results give an insight on the reliability of the system under study. For example, it allows to compare the reliability of network deployments by producing several basis topologies and comparing the probability that the property is verified in each case.

In the next section, we simulate the f-MAC protocol using the basis topologies A and B. It allows to evaluate the pessimism of the proposed verification technique.

6.4 Simulation

Simulations are performed with the WSNet simulator [1]. WSNet is an event-driven simulator for large scale wireless networks with realistic radio propagation models. We keep

the same parameters as for validation: we use the log-normal shadowing module for simulation of the propagation. The main simulation parameters are the same as those of Table 1. In contrast to the theoretical calculations of section 3, in the case of WSNet, the *PER* is calculated for each packet because the propagation is simulated. In WSNet the probability to receive a packet also depends on the interferences produced by other nodes (which are added to the Gaussian noise).

For each number of nodes, 10000 simulations are performed. We perform two sets of simulations: average case simulations and worst case simulations. For the average case, during a simulation, each node has a packet to send to the sink. The simulation duration is $2 \times T_{max}$ (deduced from Equation 7). Nodes pick a random time between 0 and T_{max} to begin to transmit. They are thus desynchronized, as in the case of the verification model. In worst case simulations, each node sends only one packet without collision to the sink. It corresponds to the worst case of the verification method (every framelets but one are lost because of collisions).

In both cases, we monitor the percentage of simulations where all the packets meet the deadline T_{max}. This percentage is comparable to the percentage of cases where the property is verified P_{pv}.

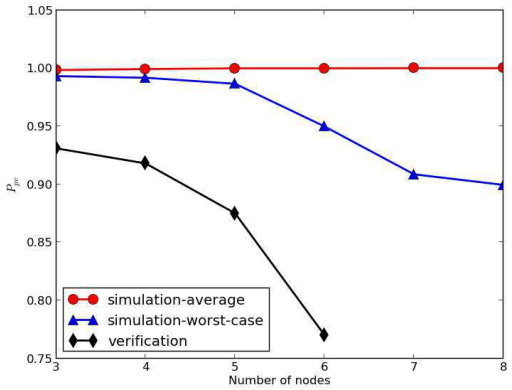

Figure 7: Simulation results for basis topology A

Figure 7 represents the percentage of simulations where the property is verified in function of the number of nodes for basis topology A. The formal verification results are also plotted. As expected from section 6.2, the simulation results are better than the verification results. This is due to the fact that, for the verification, we take the worst case of f-MAC to compute the probability that a packet is correctly received (every framelets but one collide). The worst case is a rare event and thus it does not appear often in the simulations, leading to higher probabilities. Indeed, f-MAC is asynchronous, nodes start to emit whenever they sense an event, thus many framelets may not collide. If framelets do not collide, the probability to receive correctly at least one of them increases.

Moreover, WSNet implements a radio interface with the capture effect [26]. The capture effect consists in the radio correctly receiving the packet with the highest signal level, even if there is a collision. Thus, packets are actually in collision only if the signal strength of the two packets are close. In the verification model, if two packets arrive at the same time, none is received. This phenomenon explains that P_{pv} is higher in the case of the simulations. Indeed, in this case the retransmissions serve the determinism and the reliability as well.

The results for the simulated worst case (scenarios where only one framelet is actually received) plotted on Figure 7 confirm that the verification results are conservative but not overly pessimistic.

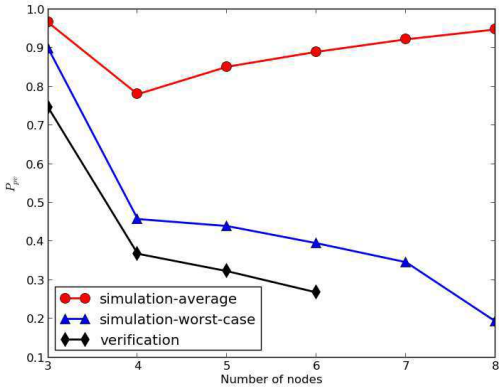

Figure 8: Simulation results for basis topology B

Figure 8 represents the percentages of simulations where the property is verified in function of the number of nodes for basis topology B. As expected, in this case the probabilities are lower, because the distances are greater. As observed in the case of the validation, the probability decreases rapidly between 3 and 4 nodes because node 3 is far away from the sink as can be seen on Figure 5. Nevertheless, in the case of the average case simulation P_{pv} increases from 4 nodes to 8 nodes. This is because, when there are more nodes, more framelets are sent (rule 2 of f-MAC). In the case of the simulations (not always the worst case), and with the capture effect, it increases the probability to receive the packet. Again, the results for the simulated worst case, which can be seen on Figure 8, confirm that the verification results are conservative but not overly pessimistic.

The verification result is a lower bound for the probabilistic behavior of the system, the simulation results are thus above the verification ones. The difference between simulation average case and verification probabilities comes from the fact that we do not often observe the worst case during the simulations. Nevertheless, we observe that the verification results are not overly pessimistic when compared to the simulated worst case.

7. RELATED WORK

In this section we describe the formalisms that have been proposed in order to do probabilistic and timed verification. We also describe formal verifications of WSN protocols proposed in the literature.

Over the years, several formalisms have been proposed to model probabilistic and timed systems. Probabilistic process algebra have been proposed [13]. A process algebra is a mathematical structure which satisfies axioms on basic operators, a process being an element of the algebra. By mod-

eling a protocol with process and operators and applying axioms, one can prove properties of the protocol. Nevertheless this is not always fully automatic and does not allow to represent explicitly time. We can notice that a process calculus for mobile ad hoc networks [25] has been proposed. It allows to model the topology and mobility of ad hoc wireless networks but no explicit time or probabilistic behaviors can be represented.

In [24], the authors present a probabilistic version of Real-Time calculus. Probabilistic arrival and service curves are defined. The compositional approach is used to derive bounds associated to probabilities. This approach allows to handle very large scale systems, because the complexity is abstracted into the arrival and service curves. Nevertheless, the bounds obtained are much less tight than those of model checking and the verification is usually not automatic.

Markov Chains (MCs) and Markov Decision Process (MDPs) can be used to model probabilistic systems and model checking can be applied to these formalisms [2]. Nevertheless, MCs lack of non-determinism, useful to represent parallel composition of nodes behavior. This issue is addressed with MDP, but they are not able to represent explicit time. Probabilistic Timed Automata [19] (PTA) have been proposed to model both probabilistic and timed behaviors. PTA are TA with modified transitions, in a location time can pass if the invariant is not violated. Any probabilistic transition which guard is satisfied can be taken. One enabled probabilistic transition is selected nondeterministically and the target location depends on the probabilities of the transition. PRISM [2] model checker allows to perform model checking on PTA models but with some limitations [15] [14]. These methods are promising but the PRISM language is limited (no arrays for example), moreover the cases treated with PRISM in the literature are usually limited to up to 4 nodes [8] [16].

The authors of [12] propose a worst case delay analysis for a wireless point-to-point transmission. They express the delay as the number of emissions necessary for the packet to reach its destination. A probability which depends on the probabilistic link model is associated with this delay. The probabilities we describe in section 3 are inspired from this work but we take a propagation model that fits better WSNs characteristics.

Various real-time protocols have been proposed for WSNs. Nevertheless few are formally verified, and nearly none formally verified tacking into account the probabilistic nature of the radio link.

In [29], authors propose Dual-Mode MAC a real-time protocol for linear WSNs. A worst case traversal time analysis is provided. The protocol is modeled and verified with UPPAAL model checker. Nevertheless, the unreliable radio link is not taken into account.

The authors of [28], check Quality of Service properties of Biomedical Sensor Networks (BSN) with UPPAAL. A protocol for data delivery in BSN is described. The authors check the absence of deadlock, the network connectivity, the packet delivery ratio, and the end-to-end delay on a network of 5 nodes. They take into account collisions but the probabilistic propagation model is not modeled for the model checking. Authors also simulate the proposed protocol with a realistic radio link. They compare the simulation results with model checking results. The model checking results are too optimistic because the realistic radio link is not taken into account.

In [6] authors use UPPAAL to verify the LMAC protocol. All connected topologies of 4 and 5 nodes are checked and authors are able to point out relevant faults. Our approach is also to check all connected topologies (deriving from a basis topology in our case), but we add a probability to each topology, which is useful to evaluate the importance of the fault.

The authors of [17] propose to verify the lifetime of a WSN by model checking. They use the IF language, it is a formal language based on TA and composition of TA. The authors compare the worst-case lifetime of two routing protocols with up to 11 nodes. Explicit time is represented, but up to our knowledge it is not possible to represent explicitly probabilistic behaviors. It is thus not possible to model unreliable links.

Real-Time Maude is a time rewrite theory formalism. In [20], the authors use it to model OGDC, a WSN protocol. The nodes are described as objects with a location, a communication range, an amount of energy and so on. This representation seems very intuitive but the definition of the behavior of a node requires a good knowledge of rewrite theory. Nevertheless, the probabilistic behavior is modeled by sampling values from a pseudo-random sequence of number. Some behaviors are omitted, the method is thus not exhaustive and rare events cannot be considered.

8. CONCLUSION AND FUTURE WORKS

In this paper we propose a method to take into account the unreliable radio link when verifying a real-time protocol for WSN. The probability is derived from the propagation model. In the case of WSN, we choose the widely used log-normal propagation model. The proposed method has the advantage to be "verification tool agnostic". We apply the method to the f-MAC protocol, a real-time protocol for WSNs. We use UPPAAL as the verification tool. We implemented a tool that automatizes the process and thus show the feasibility of our proposition. We compare the results of the verification with simulation results. It appears that the verification is, as expected, conservative but not overly pessimistic compared to the simulated worst case. Besides we show that f-MAC is a reliable real-time protocol for WSNs (for up to 6 nodes), as we were not able to detect faults.

In the future we plan to improve the channel model by introducing the interferences from other nodes and the impact of asymetric links in the calculation of the probability that a packet is correctly received. When the number of links in the basis topology is high, there are too many cases to check. We thus plan to select cases to check according to their probability weight until the targeted probability (required by the application) is reached. We also want to investigate if it is possible to introduce the probability coming from the protocol operation to our method. Finally, we plan to do experiments and confront verification results to experimental ones. This would give an insight on the impact of the propagation model parameters on the accuracy of the verification results. The scalability of the solution remains an issue, but even if it has to be mitigated, the method could already be applied to Body Sensor Networks.

9. REFERENCES

[1] http://wsnet.gforge.inria.fr/.

[2] http://www.prismmodelchecker.org/.

[3] IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks, 29 Feb. 2012.

[4] J. Bengtsson and W. Yi. Timed automata: Semantics, algorithms and tools. In *Lectures on Concurrency and Petri Nets*, pages 87–124. 2004.

[5] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165 – 177, 1990.

[6] A. Fehnker, L. Van Hoesel, and A. Mader. Modelling and verification of the lmac protocol for wireless sensor networks. IFM'07, pages 253–272, Oxford, UK, 2007.

[7] J. Ferreira, A. Oliveira, P. Fonseca, and J. Fonseca. An experiment to assess bit error rate in can. In *Proceedings of 3rd International Workshop of Real-Time Networks (RTN2004)*, pages 15–18, 2004.

[8] M. Fruth. Probabilistic model checking of contention resolution in the ieee 802.15. 4 low-rate wireless personal area network protocol. In *ISoLA'06*, pages 290–297. IEEE, 2006.

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report, Technical Report UCLA/CSD-TR 02, 2002.

[10] A. D. Gerd Behrmann and K. Larsen. A tutorial on uppaal. In *Formal Methods for the Design of Real-Time Systems*, volume 3185 of *Lecture Notes in Computer Science*, pages 33–35. 2004.

[11] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[12] K. Jaffrès-Runser. Worst case delay analysis for a wireless point-to-point transmission. In *RTN'12 (colocated with ECRTS'12)*, Pisa, Italy, 2012.

[13] C.-C. Jou and S. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In *CONCUR '90 Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 367–383. 1990.

[14] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic games for verification of probabilistic timed automata. *Formal Modeling and Analysis of Timed Systems*, pages 212–227, 2009.

[15] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 29(1):33–78, 2006.

[16] M. Kwiatkowska, G. Norman, and J. Sproston. Probabilistic model checking of the IEEE 802.11 wireless local area network protocol. *Process Algebra and Probabilistic Methods: Performance Modeling and Verification*, pages 1–8, 2002.

[17] L. Mounier, L. Samper, and W. Znaidi. Worst-case lifetime computation of a wireless sensor network by model-checking. PE-WASUN '07, pages 1–8, Crete Island, Greece, 2007.

[18] A. Mouradian and I. Augé-Blum. Modeling Local Broadcast Behavior of Wireless Sensor Networks with Timed Automata for Model Checking of WCTT. In *WCTT'12 (colocated with RTSS'12)*, pages 23–30, San Juan, Puerto Rico, 2012.

[19] G. Norman, D. Parker, and J. Sproston. Model checking for probabilistic timed automata. *Formal Methods in System Design*, pages 1–27, 2012.

[20] P. Olveczky and S. Thorvaldsen. Formal modeling and analysis of wireless sensor network algorithms in Real-Time Maude. *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, 2006.

[21] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. SenSys '04, pages 95–107, Baltimore, USA, 2004.

[22] M. V. Ramesh. Real-time wireless sensor network for landslide detection. In *SENSORCOMM '09*, pages 405–409, Athens, Greece, 2009.

[23] U. Roedig, A. Barroso, and C. Sreenan. f-mac: A deterministic media access control protocol without time synchronization. In *Wireless Sensor Networks*, volume 3868, pages 276–291. 2006.

[24] L. Santinelli and L. Cucu-Grosjean. Toward probabilistic real-time calculus. *SIGBED Rev.*, 8(1):54–61, 2011.

[25] A. Singh, C. Ramakrishnan, and S. A. Smolka. A process calculus for mobile ad hoc networks. *Science of Computer Programming*, 75(6):440 – 469, 2010.

[26] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *SenSys'06*, pages 237–250. ACM Press, 2006.

[27] R. Tan, G. Xing, J. Chen, W.-Z. Song, and R. Huang. Quality-driven volcanic earthquake detection using wireless sensor networks. RTSS '10, pages 271–280, San Diego, CA, USA, 2010.

[28] S. Tschirner, L. Xuedong, and W. Yi. Model-based validation of qos properties of biomedical sensor networks. EMSOFT '08, pages 69–78, Atlanta, USA, 2008.

[29] T. Watteyne, I. Augé-Blum, and S. Ubéda. Dual-mode real-time mac protocol for wireless sensor networks: a validation/simulation approach. InterSense '06, page 2, Nice, France, 2006.

[30] J. Zhang, W. Li, N. Han, and J. Kan. Forest fire detection system based on a zigbee wireless sensor network. In *Journal of Beijing Forestry University, vol. 29, no. 4*, pages 369–374, 2007.

[31] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *IEEE SECON'04*, pages 517–526, 2004.