A More Realistic Energy Dissipation Model for Sensor Nodes

Raquel A.F. Mini[1 2], **Antonio A.F. Loureiro**[1], **Badri Nath**[3]

[1]Department of Computer Science – Federal University of Minas Gerais
Belo Horizonte, MG, 31270-010, Brazil

[2]Department of Computer Science – Pontifical Catholic University of Minas Gerais
Belo Horizonte, MG, 30535-610, Brazil

[3]Department of Computer Science – Rutgers University
Piscataway, NJ, 08854, USA

raquel@dcc.ufmg.br,loureiro@dcc.ufmg.br,badri@cs.rutgers.edu

Abstract. *The efficient use of the limited energy resources is the key challenge in the design of wireless sensor networks. Due to the paramount importance of the energy conservation, all protocols for these networks have to be designed taking into account their energy consumption. When simulation is used to analyze the performance of any energy-aware algorithm, it is necessary to have the information about the energy drop in a sensor node. To this end, in this paper, we propose an energy dissipation model that describes the behavior of a sensor node in terms of energy consumption. In the evaluation of our model, we compare it with an Ideal model that represents the lower bound in terms of energy consumption. Based on the energy dissipation model proposed in this paper, more realistic results can be obtained when evaluating solutions for wireless sensor networks.*

1. Introduction

Wireless sensor networks refer to a group of sensor nodes linked by a wireless medium to perform distributed sensing tasks [Pottie and Kaiser, 2000]. These nodes have limited resources, such as limited computing capability, memory and energy supplies, and they must balance these restricted resources in order to increase the lifetime of the network. Therefore, the design of these networks must consider the energy conservation as a fundamental issue and devise mechanisms for extending the network lifetime.

When simulation is used to analyze the performance of any energy-aware protocol for a sensor network, we have to know how the energy dissipation happens in the sensor nodes. To this end, in this work, we propose a State-based Energy Dissipation Model (SEDM) that tries to represent realistically the behavior of a sensor network in terms of energy dissipation. Based on this model, more realistic results can be obtained when evaluating solutions to wireless sensor networks that take into account energy aspects of the sensor devices.

To our knowledge, there is only one work that has addressed the problem of modeling the energy consumption in a sensor network [Zhao et al., 2002]. In that work, two energy dissipation models are proposed. The first one is the *uniform dissipation* model. During a sensing event, each node n in the network has a probability p of initiating a local sensing activity, and every node within a circle of radius r centered at node n consumes a fixed amount of energy e. The second one is the *hotspot dissipation* model. In this model, there are h fixed hotspots uniformly distributed on the sensor field. Each node n has a probability $p = f(d)$ to initiate a local sensing activity, and every node within a circle of radius r centered at node n consumes a fixed amount of energy e, where f is a density function, and $d = min_{\forall i}\{|n - h_i|\}$ is the distance from node n to the nearest hotspot.

The main drawback of these models is that they consider that, when an event occurs, all nodes inside its area of influence will immediately see this event. This supposition can be plausible only in situations where all nodes in the sensor field keep their sensing on during all time. This may not be an appropriate approach to deal with sensor networks. As stated in [Hill et al., 2000], the best way to save energy is to make unused components inactive whenever possible. Thus, in sensor networks, nodes that are not in use should be turned off to conserve energy.

Another limitation of the work proposed in [Zhao et al., 2002] is that they do not model any communication among the sensors, neither between the sensors and the monitoring node. In a sensor network, nodes have to communicate in two main situations. Firstly, they have to communicate in order to perform cooperative tasks that are those in which sensors communicate with each other to disseminate information related to the event. For example, the nodes can exchange information about a moving object in order to send to the monitoring node the best approximation about the object position. Secondly, nodes have to communicate in order to route the sensed information to the monitoring node. In this case, nodes near the monitoring node are used more frequently to route information collected in the entire network. Neither of these communications are modeled by the *uniform dissipation model* nor the *hotspot dissipation model.* Other problems of these models include the assumption that all nodes working in a sensing event consume the same amount of energy, and that all events have the same radius of influence.

In the State-based Energy Dissipation Model (SEDM), we propose using various operation modes with different levels of activation and with different levels of energy consumption for the sensor nodes. In the analysis of our model, we assume that the arrival of events in the network follows a Poisson process or a Pareto distribution. In addition, three other models are used to describe the behavior of events. In the first one, the event is static and has a fixed size. In the second model, the event moves around in the sensor field and, in the third one, the event is static but its area of influence increases with a given rate. The evaluation of the SEDM is contrasted with an Ideal model in which the sensor nodes have a global knowledge about the network. By global knowledge, we mean that the sensor nodes can do their best to perform a given task spending the minimum amount of energy. Thus, the SEDM is compared with an Ideal model that represents the lower bound in terms of energy consumption.

This paper is organized in the following way. Section 2 presents the energy dissipation model that we propose to describe the behavior of a sensor node, and to simulate its energy drop. Section 3 describes the Ideal energy dissipation model that is used as a comparison model to evaluate the SEDM. In Section 4, we analyze the performance of our approach, and, in Section 5, we present the conclusions of our work.

2. State-based Energy Dissipation Model

In this section, we present the State-based Energy Dissipation Model, the event arrival models, and the event generation models.

2.1. Modeling

The conservation of energy is the paramount issue to be considered in the design of sensor networks. According to [Hill et al., 2000], a wireless sensor network should embrace the philosophy of getting the work done as quickly as possible and going to sleep. This can be achieved in a framework in which the nodes have different operation modes with different levels of activation, and, consequently, with different levels of energy consumption. Besides, as soon as possible, the nodes have to go to a mode that consumes less energy. Using this idea, we propose a State-based Energy Dissipation Model (SEDM) to describe how the energy is spent in a sensor node.

The kind of sensor network we will work is one in which nodes are static and homogeneous, the replacement of battery is infeasible or impossible, and also there is only one static monitoring node with plenty of energy. We suppose that nodes are deployed randomly forming a high-density network in a flat topology. In relation to the data delivery model, we simulate an event-driven network in such a way that sensors report information only if an event of interest occurs. In this case, the monitoring node is interested only in the occurrence of a specific event or set of events. In the SEDM, we suppose that the information about the events are sent to the monitoring node only at the end of each event. Also, we assume that all nodes in the area of influence of one event have to work on that event. As soon as these nodes realize that there is an event, they turn on their sensing. The communication model used is a cooperative sensor model in which the communication between the nodes is beyond the relay function needed for routing, and sensors communicate with each other to disseminate information related to the event.

The energy consumption in a sensor node can be divided in three main parts: sensing, data processing and communication [Sohrabi et al., 2000]. In a wireless sensor network, communication is the major consumer of energy. Taking the example described in [Pottie and Kaiser, 2000], the energy cost of transmitting 1 KB a distance of 100 meters is approximately 3 J. By contrast, a general-purpose processor with 100MIPS/W power could execute 300 million instructions for the same amount of energy. Thus, in wireless sensor networks, nodes have to do more processing, as opposed to exchanging raw data over the air. In this work, we model only the energy spent by communication and sensing of the node. Analyzing all possible combinations for performing sensing and communication, we find six operation modes: Mode 1: sensing off, radio off; Mode 2: sensing on, radio off; Mode 3: sensing off, radio receiving; Mode 4: sensing on, radio receiving; Mode 5: sensing off, radio transmitting; and Mode 6: sensing on, radio transmitting.

The values of power consumption for each operation mode were calculated based on information presented in [Hill et al., 2000]. In that work, the authors describe a sensor network architecture and some physical specifications, such as the values of current required by each hardware component of a sensor node. Using the values presented in that paper and knowing that sensors work with a voltage of 3 V, we can calculate the power consumption for each mode: Mode 1: 28.50 μW; Mode 2: 38.72 mW; Mode 3: 49.20 mW; Mode 4: 52.20 mW; Mode 5: 71.70 mW and Mode 6: 74.70 mW.

Nevertheless, from a practical point of view, not all operation modes are useful due to limitations in the sensor node. In general, the hardware of these nodes does not have the capability of changing between so many states. Besides, in practice, the state transitions consume energy and time. In general, a deeper sleep state consumes less power and has a longer wake-up time [Sinha and Chandrakasan, 2001]. However, in our model, we do not take into account the overhead of these state transitions. We believe that this information should be considered when an application-specific energy dissipation model is defined.

In order to simplify the model that describes the behavior of the sensor node, we use fewer operation modes, joining states that are similar. Observing the power consumption of the six operation modes, we can see that there is only a small difference between the power consumption of modes 3 and 4, and between modes 5 and 6. The difference between these modes are the sensing part that consumes only 1 mA. Therefore, in our energy dissipation model, we consider both modes 3 and 4, and modes 5 and 6, as being the same mode. With these modifications, we simplify the energy dissipation model because fewer transition functions have to be defined. Thus, in the energy dissipation model proposed in this work, the following operation modes are used:

- Mode 1: sensing off, radio off (28.50 μW);
- Mode 2: sensing on, radio off (38.72 mW);
- Mode 3: sensing on, radio receiving (52.20 mW);
- Mode 4: sensing on, radio transmitting (74.70 mW).

These values will be used throughout all simulations.

In the SEDM, in each instant of time, each node is in one of the four operation modes. The transitions between these modes are described by the diagram of Figure 1. In this diagram, the operation modes are represented by states 1, 2, 3 and 4. In addition, it was necessary to represent more two states $2'$ and $3'$. The state i' also represents the operation mode i. As example, both states 2 and $2'$ represent the operation mode 2, the only difference is that when a node goes to state 2, it always starts a timer, while in state $2'$, it verifies if is there any event for it. Thus, in terms of energy consumption, state i is exactly the same as state i'. The only difference is that the behavior of a node that goes to state i is different from the one that goes to state i'.

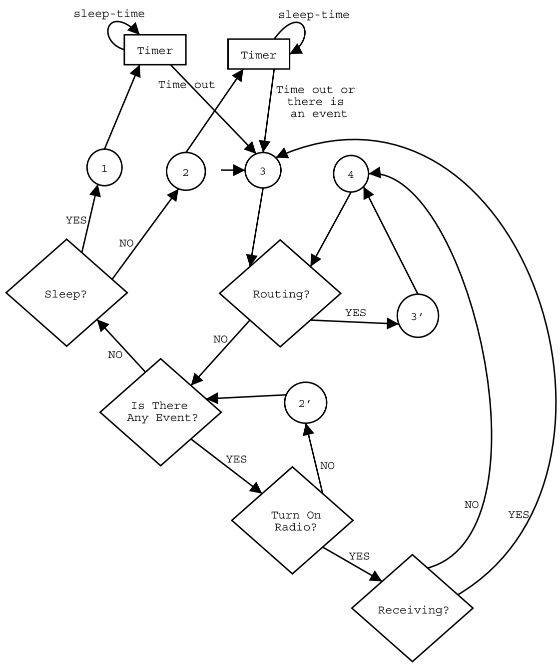


Figure 1: Diagram of the State-based Energy Dissipation Model.

In the SEDM, at the beginning of simulation, all nodes go to state 3. When a node is in state 3, its radio is on, and thus it can work as a router, transmitting packets in the direction to the monitoring node. If the node has to act as a router, it goes to state 3, and then to state 4. If the node does not have to work as a router, it verifies if there is any event for it. If there is no event for this node, it verifies if it goes to state 1 or 2. When there is an event for it, the node verifies if it is necessary to turn on its radio. This task is useful to model the cooperation needed in the applications envisioned for sensor networks. If the node has to turn on its radio, another test is executed to see if this node has to turn on its radio in the receiving or in the transmitting mode.

When a node goes to state 1, it will be sleeping for *sleep-time* seconds. During this period, this node will be saving energy but it will not be able to communicate or to sense any event. After *sleep-time* seconds, the node wakes up and goes to state 3 to see if there is any event for it or if there is any node trying to communicate with it. If a node goes to state 2, it will be in this state for *sleep-time* seconds, but unlike state 1, a node that is in state 2 can see the occurrence of an event because, in this state, the sensing is on. If an event occurs during the *sleep-time* seconds, or the time *sleep-time* ends, the node goes to state 3 to see if there is any node trying to communicate with it.

In the model of Figure 1, there are five tests. The answers for these tests are guided by parameters of the model. Bellow, we describe these parameters.

- "Routing?" – When an event ends, all nodes in the straight line between the center of this event and the monitoring node have their *routing-bit* set to one. When a node goes to either state 3 or 4, it checks this *routing-bit* to see if it is necessary to work as a router. The "Routing" test only checks the *routing-bit* to see if it is necessary to turn on the radio in order to route packets to the monitoring node. Using this test, we model the communications needed to route the sensed information to the monitoring node.
- "Is There Any Event?" – When a node is in the operation modes 2, 3 or 4, its sensing is on, and thus it can see if there is an event for it. This test checks it.
- "Sleep?" – This answer is obtained using the parameter *sleep-prob*. With probability *sleep-prob*, the node goes to state 1, and with $(1-$*sleep-prob*$)$, it goes to state 2. The greater the value of *sleep-prob*, the larger will be the coverage area. However, more energy will be spent in this case.
- "Turn On Radio?" – This question is answered using the parameter *turn-on-radio-prob*. This parameter models the communication needed to perform cooperative tasks that happen when nodes working in a sensing activity exchange information in order to get a better knowledge about the observed phenomenon.
- "Receiving?" – This question is answered using the parameter *receiving-prob*. When a node has to turn on its radio to perform cooperation, it will go to state 3 with probability *receiving-prob*, and to state 4 with probability $(1-$*receiving-prob*$)$.

As described above, in this work, we model the transmission of the sensed information to the monitoring node. We simulated this behavior turning on the *routing-bit* of all nodes in the straight line between the central point of the event and the monitoring node. Then, when these nodes turn on their radios, they will go for a short time to state 3 and after that to state 4. This is a simple model that does not deal with some of the problems faced by routing algorithms, such as retransmissions and collisions. However, this simplistic model can give us a lower bound of the amount of energy spent by any routing algorithm. Accordingly, in practice, whatever routing algorithm will spend more energy than the amount spent by the SEDM.

The state transition described above tries to capture the behavior of a sensor node specially in terms of energy consumption. As there are no real large sensor networks implemented already, we have no information about the real energy dissipation of a sensor node. Although, we believe that this model can represent the energy drop in an acceptable way.

In this section, we explained the behavior of sensor nodes. However, we still have to model the events in the sensor networks. In next two sections, we present the event arrival models and the three models used to simulate the behavior of the events.

2.2. Event Arrival Models

In this section, we present the models that describe how events arrive. We consider that an event occurs when a sensor node picks up a signal with power above a predetermined threshold. An event can be static, such as a localized change in temperature or pressure in an environment monitoring application, or it can propagate, such as signals generated by a moving object in a tracking application.

In the SEDM, we simulate two types of arrival models. In the first one, the event arrival is modeled by a Poisson process with parameter λ. This process is appropriate to model events that happen randomly and independently from each other. In this model, the times between successive events are an independent exponential random variable with rate λ. Then, the inter-arrival time of events modeled by a Poisson process is given by: $f(x) = \lambda e^{-\lambda x}$.

In the second model, the event arrival is modeled by a Pareto distribution. This distribution has a heavy-tailed property that implies that small occurrences are extremely common, whereas large instances represent very few occurrences. When a Pareto distribution is used to simulate the inter-arrival time of events, they will happen in bursts. This is because most of the inter-arrival time will be small, meaning that we have lots of events. However, the occurrence of a large inter-arrival time cannot be neglect, and thus it is possible to have long periods of time without any event. Then, using this distribution, we can model events that arrive in bursts. The Pareto distribution is given by the equation: $f(x) = \frac{ak^a}{x^{(a+1)}}$, where a is the shape parameter. A small value of a leads to more bursts in the event arrival. The parameter k represents the initial value and, in our simulations, we consider $k = 1$.

2.3. Event Generation Models

In the previous section, we presented the models that describe the event arrival. In this section, we show the models that explain the behavior of events in the SEDM. When an event arrives, a position (X, Y) is randomly chosen for it, and its behavior is described by one of the following three models:

- Static Event Model: events are static and have a fixed size. This model represents a basic event that is static and has a fixed size. The radius of influence of each event is a random variable uniformly distributed between *event-radius-min* and *event-radius-max*, and all nodes within the circle of influence of an event will be affected by it. This means that the test "Is There any Event?" for these nodes will return true. The duration of each event is uniformly chosen between the values *event-duration-min* and *event-duration-max* seconds.
- Dynamic Event Model: events are dynamic and have a fixed size. Like in the static model, in this case, the radius of influence of each event is a random variable uniformly distributed between *event-radius-min* and *event-radius-max*. However, the duration of each event will not be determined by a random variable like in the static model. In the dynamic model, in each second of simulation, each event will cease with probability *cease-prob*. If the event does not cease, it will move with probability *mobility-prob*, and will stay still with probability $(1-$*mobility-prob*$)$. If the event has to move, it will choose randomly one of the eight neighbor positions to move to. These neighbor positions are defined in relation to the position (X, Y), that represents the central point of the event, as being the possible combinations: $(X \pm 1, Y \pm 1)$, $(X, Y \pm 1)$ and $(X \pm 1, Y)$.
- Static and Increasing Event Model: events are static and have an increasing size. Each event starts with a minimum size and its area of influence gets larger as time goes by. In this model, like in the second one, in each second of simulation, each event will cease with probability *cease-prob*. If the event does not cease, it will increase its area of influence with probability *increase-prob*. When an event increases its area, its radio of influence is incremented by one unit of area.

When an event ends, data has to be propagated to the monitoring node. As described earlier, we simulate this behavior making all nodes in the straight line between the point (X, Y)

and the monitoring node go for a short time to state 3 and after to state 4. Thus, in the SEDM, the data is sent to the monitoring node only at the end of each event.

3. Ideal Energy Dissipation Model

In order to evaluate the SEDM, we compare it with an *Ideal Energy Dissipation Model.* In this model, we consider that all nodes have a global knowledge about the network. For example, even if a node is in mode 1, it knows if there is an event for it or if there is any other node trying to communicate with it. In these situations, this node can wake up immediately, without the overhead of waking up frequently to see if there is any task to be done. Therefore, in the Ideal model, all nodes know exactly the behavior in the network and they can do their best to perform the desired task. This implies that this model represents the lower bound in terms of energy consumption to the type of wireless sensor networks we are considering in this work.

In Figure 2, we have the state transition of the Ideal Energy Dissipation Model. In this model, at the beginning of simulation, all nodes go to state 1. The tests in this diagram are the same of the SEDM model. We can see that the tests "Routing?" and "Is There Any Event?" are reachable from any one of the four states, meaning that all nodes have a global knowledge about the network without the overhead of turning frequently the sensing and the radio on.

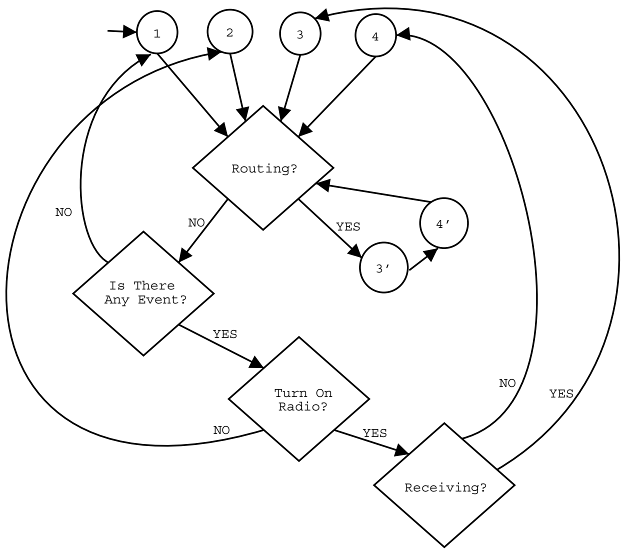


Figure 2: Diagram of the Ideal Energy Dissipation Model.

The main difference between the Ideal and the SEDM is the fact that all nodes, in the Ideal model, can do their best because they have a global knowledge about the network. It is important to point out that the Ideal model does not represent the behavior of a real wireless sensor network. In these networks, there is no way to tell a node that there is an event for a node if the node has both its sensing and radio off. The behavior of a real sensor network is best represented by the SEDM. Our goal, when presenting the Ideal model, is to have a way of comparing the SEDM with a lower bound in terms of energy consumption. In next section, we present the comparison between these two models.

4. Simulation Results

In order to illustrate the behavior of the energy dissipation models, we implemented the SEDM and the Ideal model in the ns-2 simulator [ns2, 2002]. Firstly, in Section 4.1, we show their basic

Parameter	Value	Parameter	Value
Number of Nodes	100	**Dynamic Event Model**	
turn-on-radio-prob	0.4	*event-radius-min*	5 m
receiving-prob	0.7	*event-radius-max*	10 m
Static Event Model		*cease-prob*	0.2
event-duration-min	5 sec	*mobility-prob*	0.5
event-duration-max	50 sec	**Static and Increasing Event Model**	
event-radius-min	5 m	*cease-prob*	0.2
event-radius-max	50 m	*increase-prob*	0.8

Table 1: Default values used in the simulations

operation to have a clear idea of the differences between these models. Secondly, in Section 4.2, we show the amount of residual energy at the end of simulation when using these two models. Finally, in Section 4.3, we analyze the performance of the SEDM in terms of non-detected events for a variety of situations.

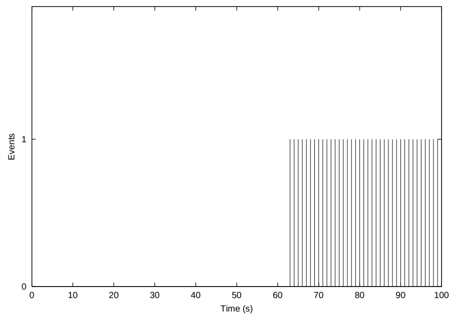
The numerical values chosen for the base case of our simulations can be seen in Table 1. Unless specified otherwise, these values are used as the parameters throughout the remainder of this work. Moreover, in all simulations, the nodes are deployed in a sensor field of size 50×50 m^2, each node has a communication radio of 15 m, and an initial energy of 100 J. Besides, the monitoring node is positioned at the middle of the field at the position $(25, 25)$, all nodes are immobile and can communicate with other nodes within their communication range.

4.1. Basic Operation

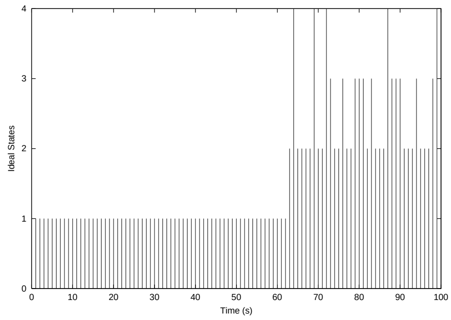
The goal of this section is to illustrate the basic operation of the SEDM and the Ideal model. To this end, we plot, for the sake of illustration, for one specific node, the interval of time in which there is an event in its area, its states (operation mode) and its energy drop when using the SEDM and the Ideal model during a simulation of 100 seconds. The arrival of the events is modeled by a Poisson process with λ = 0.04 and 0.1. Besides, in all simulations of this section, we used *sleep-prob* = 0.8 and *sleep-time* = 10 s.

In Figure 3-a, we can see that there are events for this specific node in the interval from 63 to 99 seconds. When the event starts, in the Ideal model, the node wakes up immediately. Also, using the SEDM, at time 63, the node was in state 2, then this node could see the occurrence of this event immediately. Here, it is important to point out that the amount of energy spent by the two models when the node is working in a sensing event is probabilistically the same. The main difference between these two models, in terms of energy consumption, happens when there is no event and the SEDM has the cost of waking up the nodes frequently to see if there is any event for them.

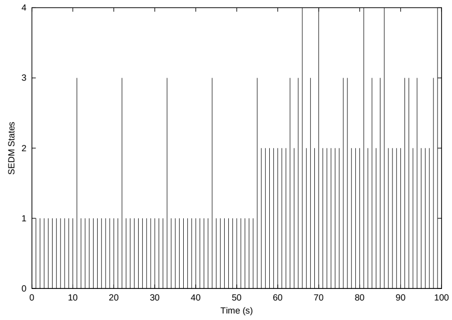
Figure 4 shows the results when we change the value of λ to 0.1, and more events are generated in the area of this node. In this simulation, we can see the main difference between the two models. When the event starts, at time 24 seconds, this node, using the SEDM, was in state 1, and thus it did not see the occurrence of this event. It only notices this event at time 33 seconds when it wakes up. In a situation like this, the Ideal model spends more energy than the SEDM. It is important to point out that using the Ideal model, all nodes inside the area of influence of the event work on that event during all its duration, and thus all events are detected by all nodes inside its area of influence. Using the SEDM, if all nodes inside the area of influence of an event keep sleeping during its duration, this event will be lost. In Section 4.3, we analyze this aspect of the SEDM, evaluating the number of events that are lost.



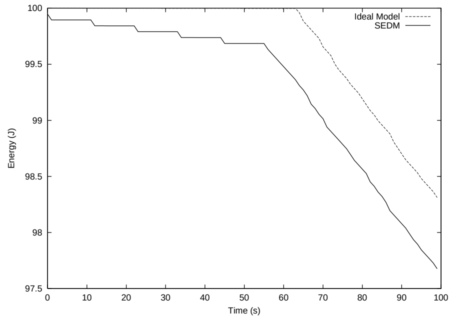
(a) Generation of events.



(b) State transition in a sensor node using the Ideal model.



(c) State transition in a sensor node using the SEDM.



(d) Energy drop in a sensor node using the SEDM and the Ideal Model.

Figure 3: Comparison between the SEDM and the Ideal model for λ = 0.04.

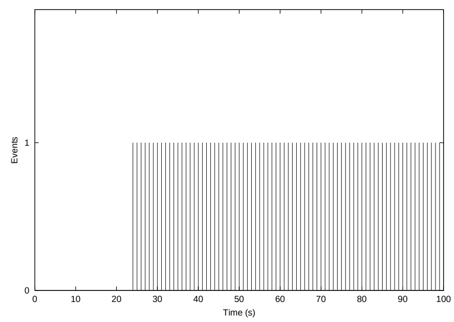
4.2. Final Energy Map

In this section, we analyze the final energy map at the end of simulation using the SEDM and the Ideal model. To this end, we run both models during 500 and 3000 seconds of simulation, and plot the amount of remaining energy in each part of the network after this time has elapsed. In all simulations, we use the following values: *sleep-time* = 10 seconds, *sleep-prob* = 0.8 and λ = 0.04. The results are depicted in Figures 5 and 6. In both maps, the figure on the left side shows the zoom on the energy map, while the one on the right side shows all maps in the same scale. We can see that the SEDM spends more energy than the Ideal model in all simulations. Besides, the longer the simulation time, the greater the difference between the amount of energy spent by the two models.

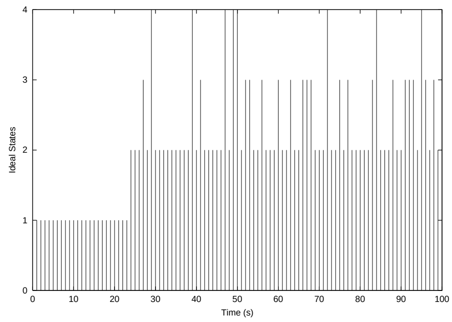
4.3. Non-detected Events

Our next goal is to analyze the performance of the SEDM in terms of events lost. We consider that an event is lost when no node in its area of influence senses its occurrence. Therefore, if one or more nodes sense the event, we consider that this event was detected. It is important to point out that this kind of analysis is plausible only for the SEDM, since in the Ideal model no event is lost.

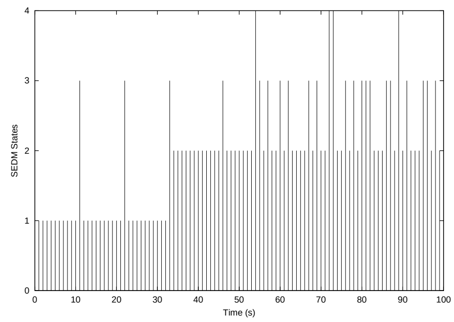
To study the number of non-detected events, we run the SEDM using both types of arrival models (Poisson and Pareto), and for all three events generation model (Static Event Model, Dynamic Event Model and Static and Increasing Event Model). In order to make a fair comparison between both arrival models, we generated 67 events in both of them. The values that generate this amount of events are λ = 0.07 and a = 0.75. In Figure 7-a, we show the instant of time that



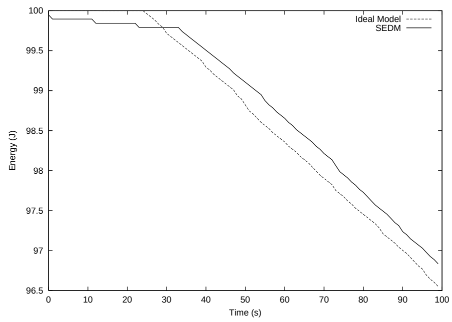
(a) Generation of events.



(b) State transition in a sensor node using the Ideal model.

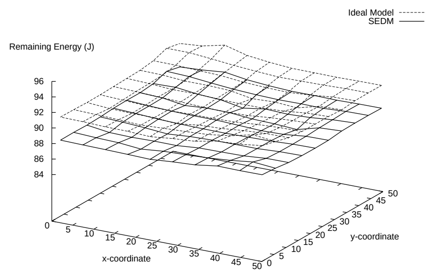


(c) State transition in a sensor node using the SEDM.

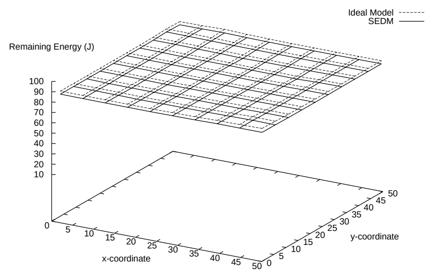


(d) Energy drop in a sensor node using the SEDM and the Ideal Model.

Figure 4: Comparison between the SEDM and the Ideal model for λ = 0.1.



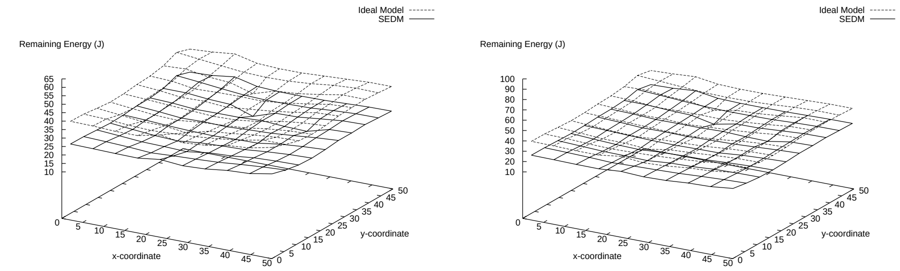
(a) Residual energy.



(b) Residual energy in a uniform scale.

Figure 5: Final energy map for 500 seconds of simulation, λ = 0.04.

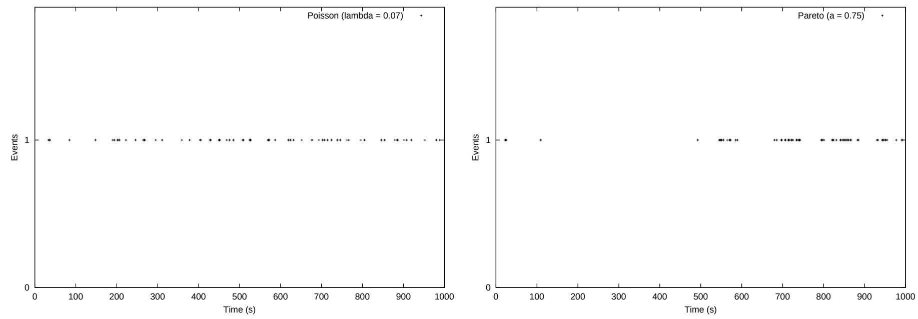
each event happens using the Poisson process and, in Figure 7-b, we have the initial time of events using the Pareto distribution. In these figures, we can see the burst characteristic of the Pareto distribution in opposite to the uniformly distributed characteristic of the Poisson arrival model. The results of the next simulations were obtained as an average of 33 runs. In all runs, we kept constant all characteristics of the events (radius of influence, duration, etc.), except their positions. Then, in all runs, we have the same events arriving at the same time in different positions of the sensing fields. This leads to a fair comparison between the models. Also, the results have a 95%



(a) Residual energy. (b) Residual energy in a uniform scale.

Figure 6: Final energy map for 3000 seconds of simulation, λ = 0.04.

confidence level.



(a) Poisson process with λ = 0.07. (b) Pareto distribution with a = 0.75.

Figure 7: Instant of time in which the 67 events were generated.

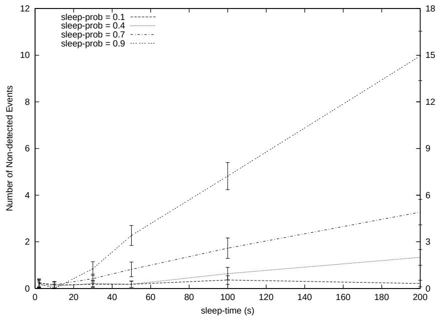
In the next sections, we illustrate the number of non-detected events when using the three models of event generation and the two models of event arrival. In Section 4.3.1, we show the number of non-detected events when we have a static event with a static area of influence. In Section 4.3.2, we show the number of non-detected events when we have a moving event with a fixed area of influence. Finally, Section 4.3.3 shows the results for static events with an increasing area of influence.

4.3.1. Static Event Model

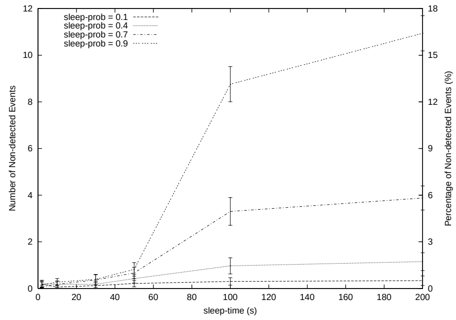
In this section, we changed the values of *sleep-prob* and *sleep-time* in order to examine the behavior of the SEDM in terms events lost. In Figure 8-a, we show the number of non-detected events when using the Poisson arrival model. The left axis of these figures show the number of non-detected events whereas, in the right one, this value is showed as a percentage value in relation to the total amount of generated events. In Figure 8-c, we have the total amount of energy spent by all nodes in the network for the simulation showed in Figure 8-a. We can see that as the value of *sleep-prob* gets larger, more events are lost, and less energy is spent by the network.

In Figure 8-b, we have the number of non-detected events when using the Pareto distribution to model the arrival of events, and, in Figure 8-d, we show the total amount of energy spent by all nodes in the network. We can observe that the total amount of energy spent by the two arrival models is basically the same.

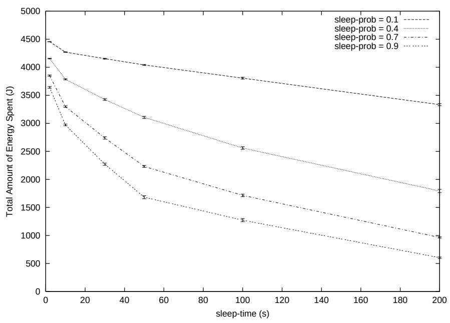
Analyzing the behavior of the SEDM, we can say that, in general, the burst arrival is more difficult to detect than the Poisson process. This is true because, when events are distributed uniformly in time, the probability of having a node with its sensing on to detect an event is higher. For that reason, in almost all simulations, the number of lost events using the Pareto distribution is higher than when using the Poisson process. However, in the simulations showed in this work, this difference is small due to the small time interval analyzed.



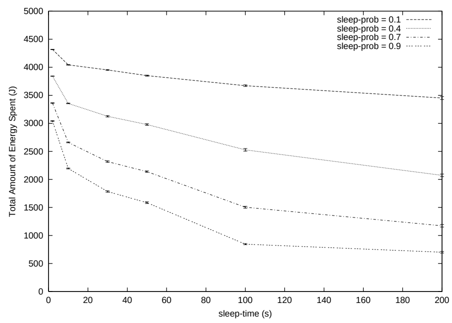
(a) Percentage of non-detected events using the Poisson process.



(b) Percentage of non-detected events using the Pareto distribution.



(c) Total amount of energy spent using the Poisson process.

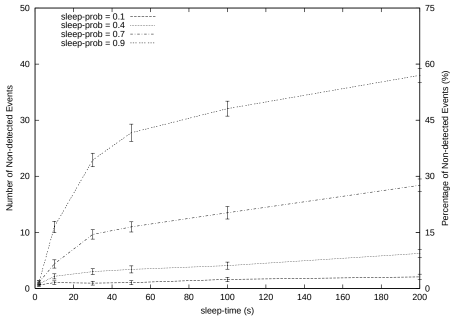


(d) Total amount of energy spent using the Pareto distribution.

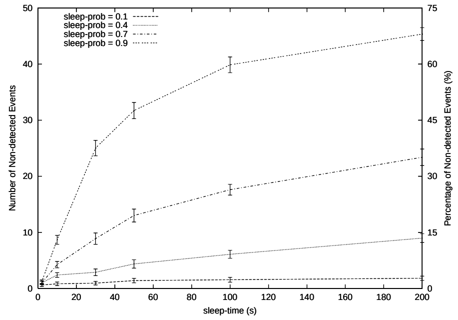
Figure 8: Non-detected events in the Static Events Model.

4.3.2. Dynamic Event Model

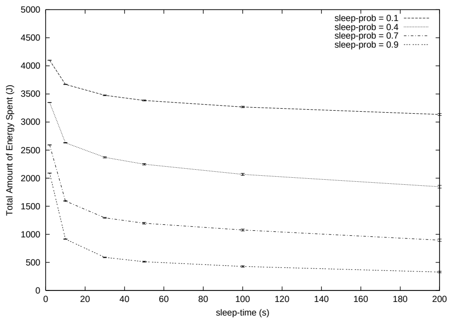
In this section, we consider that events are dynamic and have a fixed size. We can see in Figure 9 the number of non-detected events using Poisson and Pareto, and the respective amount of energy spent. It is possible to observe that, in this model, there are more events lost than in the model presented in the last section. This can be explained by the default parameters of Table 1. The radius of influence of the static model is uniformly distributed between 5 and 50 meters, whereas the radius of the dynamic model is uniformly distributed between 5 and 10 meters. Thus, in general, events of the static model are bigger than events of the dynamic model. Besides, the fact that events are static also makes them easier to be detected than the one of the dynamic model. However, it is important to point out that this result depends on the parameters of the event generation model.



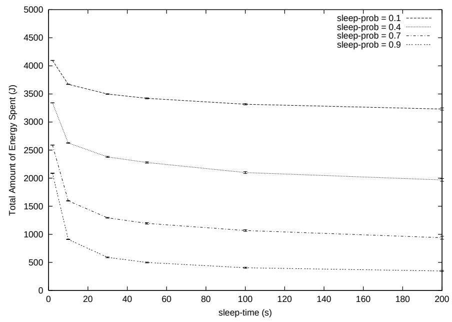
(a) Percentage of non-detected events using the Poisson process.



(b) Percentage of non-detected events using the Pareto distribution.



(c) Total amount of energy spent using the Poisson process.



(d) Total amount of energy spent using the Pareto distribution.

Figure 9: Non-detected events in the Dynamic Event Model.

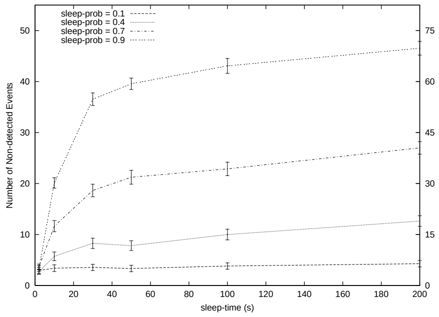
4.3.3. Static and Increasing Event Model

The static and increasing event model is examined in this section. Figure 10 shows the results for this model, which are very similar to the ones of the dynamic model. In addition, as in the previous simulations, in terms of events lost, the number of non-detected events using the Pareto distribution is slightly higher than when using the Poisson process.

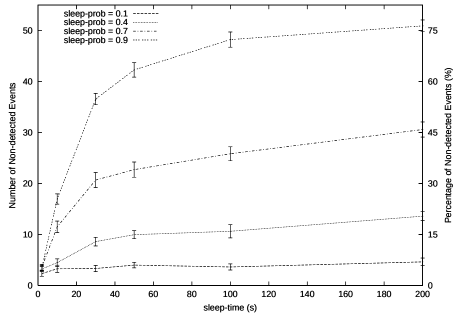
5. Conclusions

In this paper, we proposed a State-based Energy Dissipation Model (SEDM) that tries to represent more realistically the behavior of a sensor network in terms of energy dissipation. Based on this model, more realistic results can be obtained when evaluating solutions to wireless sensor networks that take into account energy aspects of the sensor devices.

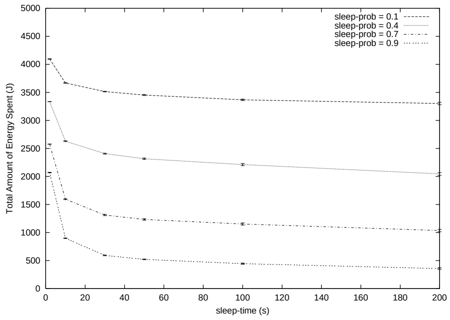
Wireless sensor networks are application-specific in a way that they will be designed specially to the sensing task at hand. Because of that, any model for these networks must be adjustable to work in various applications. The SEDM presents this characteristic since it can easily be adapted in a variety of situations. As example, if it is necessary to analyze other sensor nodes, we have just to change the power consumption of operation modes. Furthermore, other models that represent the event behavior can easily be added in the SEDM.



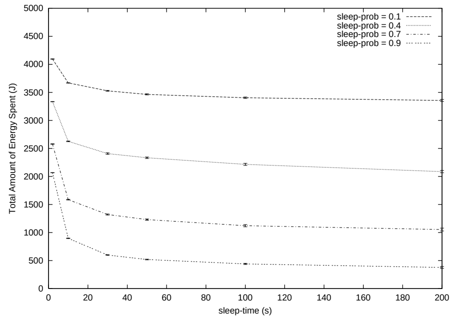
(a) Percentage of non-detected events using the Poisson process.



(b) Percentage of non-detected events using the Pareto distribution.



(c) Total amount of energy spent using the Poisson process.



(d) Total amount of energy spent using the Pareto distribution.

Figure 10: Non-detected events in the Static and Increasing Event Model.

Acknowledgment

This work has been partially supported by CNPq, Brazil, under process number 55.2111/2002-3, DARPA under contract number N-666001-00-1-8953, and a grant from CISCO systems.

References

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. (2000). System architecture directions for networked sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*.

ns2 (2002). The network simulator. http://www.isi.edu/nsnam/ns/index.html.

Pottie, G. and Kaiser, W. (2000). Wireless integrated network sensors. In *Communications of the ACM*, volume 43, pages 551–8.

Sinha, A. and Chandrakasan, A. (2001). Dynamic power management in wireless sensor networks. *IEEE Design & Test of Computers*, 18(2).

Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. (2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7:16–27.

Zhao, Y. J., Govindan, R., and Estrin, D. (2002). Residual energy scans for monitoring wireless sensor networks. In *IEEE Wilress Communications and Networking Conference (WCNC'02)*, Orlando, FL, USA.