Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

ELSEVIER

Computer Communications xx (0000) xxx–xxx

computer communications

www.elsevier.com/locate/comcom

The distinctive design characteristic of a wireless sensor network: the energy map

Raquel A.F. Mini[a,*], Antonio A.F. Loureiro[a], Badri Nath[b]

[a]Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, MG 31270-010, Brazil
[b]Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA

Received 9 January 2004; accepted 9 January 2004

Abstract

The key challenge in the design of a wireless sensor network is maximizing its lifetime. This is a fundamental problem and new protocol engineering principles need to be established in order to achieve this goal. The information about the amount of available energy in each part of the network is called the energy map and can be useful to increase the lifetime of the network. In this paper, we propose using the energy map as a protocol engineering principle for this kind of network. We argue that an energy map can be the basis for the entire design trajectory including all functionalities to be included in a wireless sensor network. Furthermore, we show how to construct an energy map using both probabilistic and statistical prediction-based approaches. Simulation results compare the performance of these approaches with a naive one in which no prediction is used. The experiments performed use an energy dissipation model that we have proposed to simulate the behavior of a sensor node in terms of energy consumption. The results show that prediction-based approaches outperform the naive in a variety of parameters.

Keywords: Energy map; Sensor networks; Prediction

1. Introduction

Wireless sensor networks (WSNs) are composed of low-cost sensor nodes that can communicate with each other in a wireless manner, have limited computing capability and memory and operate with limited battery power. These sensors can produce a measurable response to changes in physical conditions, such as temperature or magnetic field. The main goal of such networks is to perform distributed sensing tasks, particularly for applications like environmental monitoring, smart spaces and medical systems. These networks form a new kind of ad hoc network with a new set of characteristics and challenges.

Unlike conventional wireless ad hoc networks, a wireless sensor network potentially comprises hundreds to thousands of nodes [27]. The sensors often operate in noisy environments and, in order to achieve good sensing resolution, higher densities are required. Therefore, in a sensor network, scalability is a crucial factor. Different from nodes of a traditional ad hoc network, sensor nodes are generally stationary after deployment. Although the nodes are static, these networks still have dynamic network topology. During periods of low activity, the network may enter a dormant state in which many nodes go to sleep to conserve energy. Also, nodes go out of service when the energy of the battery runs out or when a destructive event takes place [20]. Another characteristic of these networks is that sensors 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. In addition, the sensors will be battery powered and it is often very difficult to change or recharge batteries for these nodes. Therefore, in sensor networks, we are interested in prolonging the lifetime of the network and thus the energy conservation is one of the most important aspects to be considered in the design of these networks.

* Corresponding author.
E-mail addresses: raquel@dcc.ufmg.br (R.A.F. Mini), badri@cs.rutgers.edu (B. Nath).

0140-3664/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.comcom.2004.01.004

COMCOM 2465—5/2/2004—21:22—RAJA—93593— MODEL 5

1.1. The energy map

The information about the remaining available energy in each part of the network is called the *energy map* and could aid in prolonging the lifetime of the network. We could represent the energy map of a sensor network as a gray level image, in which light shaded areas represent regions with more remaining energy, and regions short of energy are represented by dark shaded areas. Using the energy map, a user may be able to determine if any part of the network is about to suffer system failures in near future due to depleted energy [32]. The knowledge of low-energy areas can aid in incremental deployment of sensors because additional sensors can be placed selectively on those regions short of resources. The choice of the best location for the monitoring node can be made also based on the energy map. A monitoring node is a special node responsible for collecting information from the sensor nodes. Typically this node is named observer or end user and it is interested in obtaining information from the sensor nodes about the observed phenomenon. We know that nodes near the monitoring node probably will spend more energy because they are used more frequently to relay packets to the monitoring node. Therefore, if we move the monitoring node to areas with more remaining energy, we could prolong the lifetime of the network.

Other possible applications of the energy map are reconfiguration algorithms, query processing, data fusion, etc. In fact, it is difficult to think of an application and/or an algorithm that does not need to use an energy map. Therefore, the energy map is an important information for sensor networks. However, the naive approach, in which each node sends periodically only its available energy to the monitoring node, would spend so much energy due to communications that probably the utility of the energy information will not compensate the amount of energy spent in this process. For that reason, better energy-efficient techniques have to be devised to gather the information about the available energy in each part of a sensor network.

1.2. Protocol engineering for wireless sensor networks

In the protocol area, the term protocol engineering was coined to denote the protocol development cycle [15,21]. This area includes disciplines such as formal methods, software and knowledge-based engineering principles and basically follows the traditional software life cycle. Protocol engineering has been a very active research area during the last two decades where the fundamentals for traditional computer networks were defined and protocol design became a more systematic activity [9,10,13]. However, with the advent of WSN, new protocol engineering principles need to be established.

The key challenge in the design of a WSN is maximizing its lifetime. From the point of view of protocol design, a protocol architecture for these networks should consider a power management plane as depicted in Fig. 1. Protocols for WSNs must be energy-efficient in order to make better use of the limited energy supply of the sensor nodes.

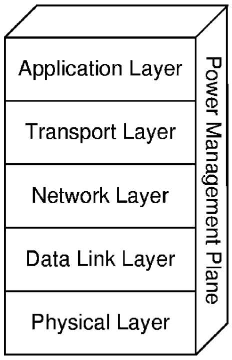


Fig. 1. Protocol architecture for a wireless sensor network with a power management plane.

Most of the protocols proposed for WSNs, which take into account the available energy in a sensor node, use the information available locally when performing a given task. For instance, some protocols [11,29,30] try to reduce the energy consumption in order to be suitable for this new kind of network. Other protocols [7,14,31] use the amount of available energy in the node when they make a decision. In many cases, to look at just the amount of the available energy in a node may either be sufficient or lead to an acceptable solution. Even in these cases, it would be interesting to evaluate whether an energy map could provide a better solution.

There are fundamental problems in WNSs, such as routing, that can benefit from having the energy map of the entire network. A routing algorithm can make a better use of the energy reserves if it selectively chooses routes that use nodes with more remaining energy, so that parts of the network with small reserves can be preserved or avoided. The protocol can also form a virtual backbone connecting high energy islands of nodes.

The protocol proposed in Ref. [17] is an example of a routing protocol that could take advantage of the energy map. In that work, it is described the Trajectory Based Forwarding protocol that is a new forwarding algorithm suitable for routing packets along a predefined curve. The idea is to embed the trajectory in each packet and let the intermediate nodes take the forwarding decisions based on their distances from the desired trajectory. If this protocol had the information about the energy map, the trajectory could be planned in order to pass through regions with more remaining energy, thus preserving or avoiding regions of the network with small reserves. Again, the goal here is to make better use the energy reserves to increase the lifetime of the network.

In this work, we propose using the energy map of a WSN as a new protocol engineering principle when designing new protocols for this kind of network. If this is the case, the design of a new protocol for a WSN can specify, given a particular scenario in the network, the best action to be taken to improve its energy efficiency. Therefore, an energy map can be the basis for the entire design trajectory including all functionalities to be included in the WSN. The effectiveness of having an application running in a wireless sensor network will depend on the success in obtaining an energy map. Note that the applicability of this map is not restricted to a particular aspect of the application, but to all activities present in the network since all of them need energy to be carried out.

In this paper, we focus on proposing mechanisms to predict the energy consumption of a sensor node in order to construct the energy map of a wireless sensor network. There are situations in which the node can predict its energy consumption based on its own past history. If a sensor can predict efficiently the amount of energy it will dissipate in the future, it will not be necessary to transmit frequently its available energy. This node can just send one message with its available energy and the parameters of the model that describes its energy dissipation. With this information, the monitoring node can update its local information about the available energy of this node. Clearly the effectiveness of this solution depends on the accuracy with which prediction models can be generated. We analyze the performance of probabilistic and statistical models, and compare them with a naive approach in which no prediction is used. In order to evaluate the approaches to construct the energy map, we need to know how is the energy drop in a sensor node. Thus, we also propose an energy dissipation model that is used to simulate the behavior of a sensor node in terms of energy consumption. Simulation results show that the use of prediction-based models decreases the amount of energy necessary to construct the energy map of WSN.

1.3. Organization of the paper

The rest of this paper is organized in the following way. Section 2 describes the model that we propose to describe the behavior of a sensor node and, thus, to simulate its energy drop. In Section 3, we describe two approaches to construct a prediction-based energy map for a WSN. We evaluate the performance of our approaches in Section 4. In Section 5, we briefly survey the related work and compare with our proposal. Our concluding remarks and directions for our future work are presented in Section 6.

2. Energy dissipation model

In order to build an energy map, we have to know how is the energy dissipation in the sensor nodes. To this end, we use an *energy dissipation model* that tries to describe the energy drop at each sensor node. To our knowledge, there is only the work by Zhao, Govindan, and Estrin [32] that has addressed this problem. 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 other one is the *hotspot dissipation* model, where there are h fixed hotspots uniformly distributed randomly 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 do not take into account the fact that a lack of energy in these networks will influence their behaviors. For example, to conserve energy, some sensors have to sleep during some part of the time. Other problems include the assumption that all nodes working in a sensing event will consume the same amount of energy and that all events have the same radius of influence. In this work, we propose a model that tries to represent more realistically the behavior of a sensor node in terms of its energy dissipation. In the following we describe our energy dissipation model.

The conservation of energy is the paramount issue to be considered in the design of sensor networks. The best way to save energy is to make unused components inactive whenever possible. This can be achieved in a framework in which nodes have different modes of operation with different levels of activation and, thus, different levels of energy consumption and, as soon as possible, they go to a mode that consumes less energy. In sensor networks, the nodes will have to change between different states of activation. Using this idea, we propose a model to describe the behavior of a sensor node and evaluate and simulate its energy dissipation. In this model, each node has four modes of operation: *state 1*, sensing off and radio off; *state 2*, sensing on and radio off; *state 3*, sensing on and radio receiving; *state 4*, sensing on and radio transmitting. These modes represent the simplicity of the hardware found in sensor nodes.

In this model, the following parameters are used: λ, arrival rate of the events; *sleep-time*, time the node will sleep; *sleep-prob*, when a node is not acting in a sensing event it will be in state 1 with probability *sleep-prob*, and in state 2 with probability (1—*sleep-prob*); *event-radius-min* and *event-radius-max*, the radius of each event will be a random variable uniformly distributed between *event-radius-min* and *event-radius-max*; *event-duration-min* and *event-duration-max*, the duration of each event will be a random variable uniformly distributed between *event-duration-min* and *event-duration-max*; *statei-prob*, probability of being in state i during an event; *dist-line*, distance of influence when an information is relayed to the monitoring node.

The behavior of the sensor node can be described by the diagram depicted in Fig. 2. At the beginning of the simulation, each node goes to state 1 with probability *sleep-prob* or to state 2 with (1—*sleep-prob*).

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

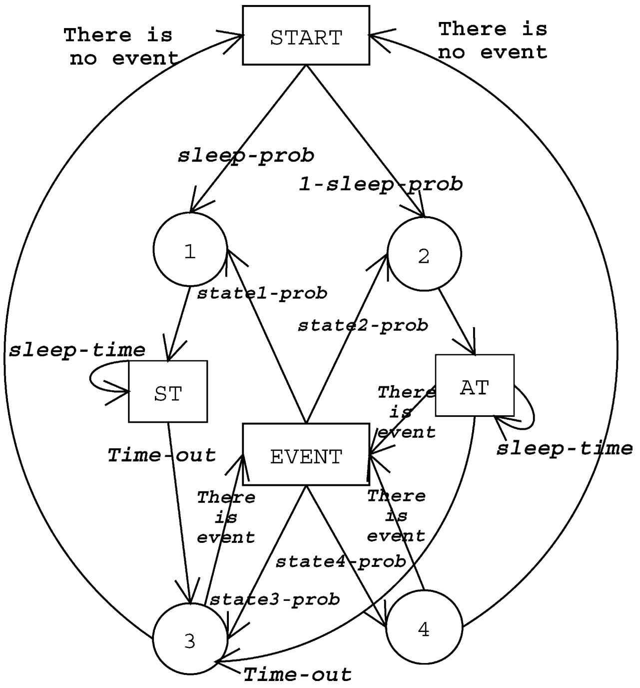


Fig. 2. Diagram of the state transition model: 1, 2, 3, and 4 represent the modes of operation of each node; ST and AT are synchronous and asynchronous timers respectively.

event. After *sleep-time* seconds, the node wakes up and goes to state 3 to check whether there is any event for it or there is any node trying to communicate with it. If there is an event, the node will go to states 1, 2, 3 or 4, with probabilities *state1-prob*, *state2-prob*, *state3-prob* and *state4-prob*, respectively. If there is no event, the node will go to state 1 with probability *sleep-prob* and to state 2 with (1—*sleep-prob*).

If a node goes to state 2, it will be in this state for *sleep-time* seconds, but unlike in state 1, a node that is in state 2 can check the occurrence of an event since the sensing is on. If an event occurs during *sleep-time* seconds, the node will go to states 1, 2, 3 or 4, with probabilities *state1-prob*, *state2-prob*, *state3-prob* and *state4-prob*, respectively. After passing *sleep-time* seconds and no event happens, the node goes to state 3 to check whether there is a node trying to communicate with it and, again, it will go to state 1 with probability *sleep-prob* and to state 2 with (1—*sleep-prob*).

In this model, the events are simulated by a Poisson process with parameter λ. Therefore, the number of events at each second of simulation is described by the random variable:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}. \quad (1)$$

When an event occurs, a position (X, Y) is randomly chosen for it. 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 when these nodes realize that there is an event for them (the nodes have to be in states 2, 3 or 4), they will go to states 1, 2, 3 or 4, with probabilities *state1-prob*, *state2-prob*, *state3-prob* and *state4-prob*, respectively. The duration of each event is uniformly chosen between *event-duration-min* and *event-duration-max* seconds. After that time, the data has to be propagated to the monitoring node. We simulate this behavior making all nodes distant *dist-line* from the straight line between the point (X, Y) and the monitoring node go, for a short time, to state 3 and, after that, to state 4.

The state transition just described tries to capture the behavior of a sensor node, specially in terms of energy consumption.

3. Prediction-based energy map

As described earlier, the knowledge of the available energy reserves at each part of the network is an important information for sensor networks. A natural way of thinking about the energy map construction is one in which periodically each node sends to the monitoring node its available energy. We call this the *naive* approach. As the sensor networks may have lots of nodes with limited resources, the amount of energy spent in the naive approach will be prohibitive. For that reason, better energy-efficient techniques have to be designed to gather the information about the available energy at each part of a sensor network.

In this section, we discuss the possibilities of constructing the energy map using prediction-based approaches. Basically, each node sends to the monitoring node the parameters of the model that describes its energy drop and the monitoring node uses this information to update locally the information about the available energy at each node. The motivation that guided us to this strategy is that if a node is able to predict the amount of energy it will spend, it can send this information to the monitoring node and no more energy information will be sent during the period that the model can describe satisfactorily the energy dissipation. Then, if a node can efficiently predict the amount of energy it will dissipate in the future time, we can save energy in the process of constructing the energy map of a sensor network.

In order to predict the dissipated energy, we studied two models. In Section 3.1, we describe a probabilistic model based on Markov chains, and, in Section 3.2, we present a statistical model in which the energy level is represented by a time series and the Autoregressive Integrated Moving Average (ARIMA) model is used to make the predictions.

3.1. Probabilistic model

In this section, we claim that each sensor node can be modeled by a Markov chain In this case, the modes of operation of a node are represented by the states of a Markov chain and the random variables represent the probability of staying at each state in a certain time. Then, if each sensor node has M modes of operation, each node will be modeled by a Markov chain with M states.

Using this model, at each node, we have a sequence of random variables $X_0, X_1, X_2, \ldots$ that represents its states during the time. Then, if $X_n = i$, we say that the sensor node is in mode of operation i at time-step[1] n. In addition, given that at each time the node is in state i, there is some fixed probability, P_{ij}, that the next state will be j. This probability can be represented by: $P_{ij} = P\{X_{m+1} = j | X_m = i\}$. We can also define the n-step transition probability, $P_{ij}^{(n)}$, that a node currently in state i will be in state j after n additional transitions [25]:

$$P_{ij}^{(n)} = \sum_{k=1}^{M} P_{ik}^{(r)} P_{kj}^{(n-r)},$$

for any value of $0 < r < n$.

With the knowledge of the probabilities $P_{ij}^{(n)}$ for all nodes and the value of X_0 (initial state of each node), it is possible to use them to predict the energy drop of a sensor node. The first step to make this prediction is to calculate for how many time-steps a node will be in a state s in the next T time-steps. If the node is in state $i (X_0 = i)$, the number of time-steps a node will stay in the state s can be calculated by: $\sum_{t=1}^{T} P_{is}^{(t)}$. Also, if E_s is the amount of energy dissipated by a node that remains one time-step in state s, and the node is currently in state i, then the expected amount of energy spent in the next T times, E^T, is:

$$E^T = \sum_{s=1}^{M} \left(\sum_{t=1}^{T} P_{is}^{(t)} \right) \times E_s. \tag{2}$$

Using the value E^T, each node can calculate its energy dissipation rate (ΔE) for the next T time-steps. Each node then sends its available energy and its ΔE to the monitoring node, which can estimate the dissipated energy at each node by decreasing the value ΔE, periodically, from the amount of remaining energy of each node. The better the estimation the node can do, the fewer the number of messages necessary to obtain the energy information and, thus, the fewer the amount of energy spent in the process of getting the energy map.

3.2. Statistical model

In this section, we present the statistical model used to forecast the available energy in the sensor nodes. In this model, we represent the energy drop of a sensor node as a time series. A time series is a set of observations x_t, each one being recorded at a specific time t [3]. A discrete-time series is one in which the set T_0 of times at which observations are made is a discrete set. Continuous-time series are obtained when observations are recorded continuously over some time interval. There are two main goals of time series analysis [28]: identifying the nature of the phenomenon represented by the sequence of observations, and forecasting (predicting future values of the time series variable). In this work, we are interested in using the time series analysis to forecast future values of the available energy in a sensor node. We will use the discrete-time series in such a way that each node will verify its energy level in a discrete time interval.

We can observe that the time series which represents the energy drop of a sensor node has a clear decreasing trend.[2] In this work, we suppose that there is no replacement in the battery and no seasonality.[3] The decreasing trend will also imply in a decreasing mean and then the energy level will also be a nonstationary time series.[4]

In this work, we will use the ARIMA model to predict future values of the time series. The ARIMA model was proposed by Box and Jenkins [2] and they consist of a systematic methodology for identifying and estimating models that could incorporate both autoregressive and moving average approaches. This makes the ARIMA model a powerful and general class of models [18]. The 'integrated' part of the model is because of the differencing step necessary to make the series stationary.

The first step in developing an ARIMA model is to determine if the series is stationary. When the original series is not stationary, we need to difference it to achieve stationarity. Given the series Z_t, the differenced series is a new series $X_t = Z_t - Z_{t-1}$. The differenced data contain one less point than the original one. Although one can difference the data more than once, a small number of differences is usually sufficient to obtain a stationary time series [18]. The number of differencing applied in the original series is represented by the parameter d.

The next step in the construction of the ARIMA model is to identify the AR terms. An autoregressive model is simply a linear regression of the current value against one or more prior values of the series. The value of p is called the order of the AR model. Then, an autoregressive model of order p can be summarized by: $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + Z_t$, where X_t is the time series, $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive model parameters, and Z_t represents normally distributed random errors.

After defining the differencing and the autoregressive parameters, we have to identify the MA terms. A moving average model is essentially a linear regression of the current value of the series against the random shocks of one or more prior values of the series [18]. The random shocks at each point are assumed to come from the same distribution, typically a normal distribution, with constant location and scale. The distinction in this model is that these random

[1] A time-step is a small amount of time. We suppose that all state transitions occur at the beginning of any time-step.

[2] Trend refers to a gradual, long-term movement in the data.

[3] Seasonality refers to periodic fluctuations that are generally related to weather factors or to human-made factors such as holidays and vacations.

[4] A stationary time series is one whose statistical properties, such as mean, variance, and autocorrelation, are all constant over time.

shocks are propagated to future values of the time series. A moving average model of order q is represented by: $X_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \cdots + \theta_q Z_{t-q}$, where X_t is the time series, $\theta_1, \theta_2, \cdots, \theta_q$ are the moving average model parameters and the Z_t are random shocks to the series.

In order to use the ARIMA model we have to identify the values of p (order of the autoregressive model), d (number of differencing required to achieve stationarity), q (order of the moving average model) and the coefficients of the autoregressive and moving average models. Thus, a time series T_t can be represented by an ARIMA (p, d, q) model if, after differencing this series d times, we find a stationary time series X_t, such that for every t: $X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q}$.

When using equation above, we can predict the value of the time series in time t using the previous values and some random variables that represent the errors in the series. In general, the estimation of these parameters is not a trivial task. In Refs. [18,28], the authors describe some techniques to help in the process of parameter identification.

4. Simulation results

In order to analyze the performance of the proposed schemes, we implemented the prediction-based energy maps in the ns-2 simulator [19]. The approaches implemented were: the Markov, in which each node sends periodically to the monitoring node its available energy and its predicted energy consumption rate; and the ARIMA, in which each node sends to the monitoring node its available energy and the parameters of this model. These approaches are compared with the naive one in which each node sends periodically to the monitoring node only its available energy.

In our simulations, we use the energy dissipation model, presented in Section 2, to describe the behavior of sensor nodes and, thus, to simulate their energy dissipation. Therefore, each node has four modes of operation: state 1 (sensing off, radio off), state 2 (sensing on, radio off), state 3 (sensing on, radio receiving) and state 4 (sensing on, radio transmitting). The values of power consumption for each state were calculated based on information presented in Ref. [8]: state 1: 25.5 μW, state 2: 38.72 mW, state 3: 52.2 mW and state 4: 74.7 mW. These values will be used throughout all simulations.

In the Markov model, each node sends its available energy and its energy dissipation rate to the monitoring node. To obtain its energy dissipation rate, each node locally calculates its own probabilities, $P_{ij}^{(n)}$. In this case, P_{ij} will be the number of times the node was in state i and went to state j divided by the total number of time-steps the node was in state i. With these probabilities, each node uses Eq. (2) to find its energy dissipation rate. If each node can predict efficiently its energy dissipation rate, this approach can save energy compared with the naive, because no more energy information packet has to be sent while the energy dissipation rate describes satisfactorily the energy drop in this node.

In the implementation of the ARIMA model, we have to identify the parameters p, d, q and to estimate the coefficients of the AR and MA models. The first step in fitting an ARIMA model is the determination of the order of differencing needed to stationarize the series (parameter d). Normally, the correct number of differencing is the lowest order of differencing that yields a time series which fluctuates around a well-defined mean value and whose autocorrelation function plot decays fairly rapidly to zero, either from above or below [16]. If the series still exhibits a long-term trend, i.e. a lack of tendency to return to its mean value, or if its autocorrelations are positive out to a high number of lags, it needs a higher order of differencing. In general, the optimal order of differencing is often the one at which the standard deviation is lowest [16]. In addition, if the lag 1 autocorrelation is -0.5 or more negative, the series may be over-differenced. In our simulation, we chose the smallest value of d that produces the lowest standard deviation in such a way that the lag 1 autocorrelation is not more negative than -0.5. The number of AR and MA terms was found using the autocorrelation and partial autocorrelation functions. The lag at which the partial autocorrelation function cuts off indicates the number of AR terms, and the number of MA terms is determined by the lag at which the autocorrelation function cuts off. The values of the coefficients of the AR and MA models were calculated based on a conditional sum-of-squares and maximum likelihood (minimize CSSML) method implemented in Ref. [23].

In all simulations we use the parameter *threshold* that determines the accuracy required or the maximum error acceptable in the energy map. If we define a threshold of 3%, a node will send another energy information to the monitoring node only when the error between the energy value predicted by the monitoring node and the correct value is greater than 3%. Each node can locally determine this error by just keeping the parameters of the last prediction sent to the monitoring node. Then, adjusting the value of the threshold, we can control the precision at which the energy maps are constructed.

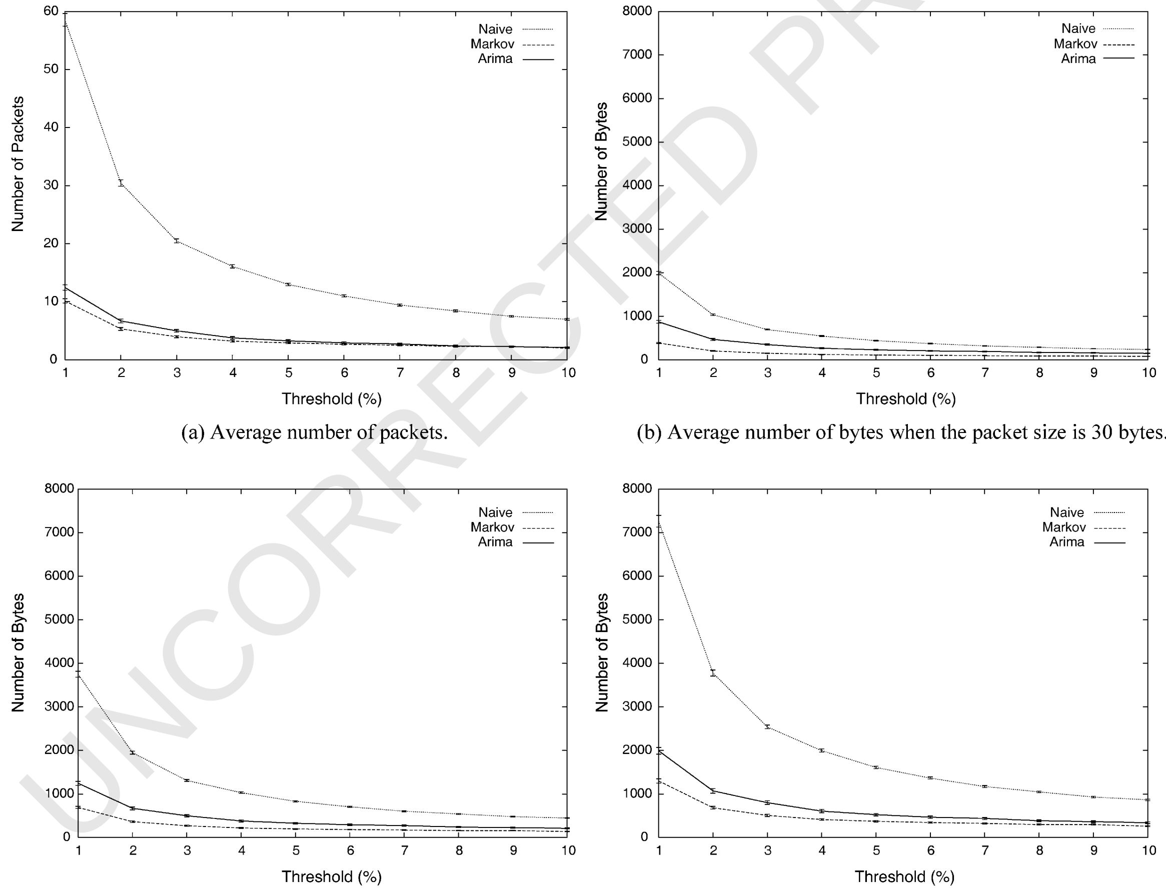
The numerical values chosen for the base case of our simulations can be seen in Table 1. Unless specified otherwise, these values are used as the parameters throughout the remainder of this work. Moreover, in all simulations, the monitoring node is positioned at the middle of the field at position (50, 50), and all nodes are immobile and can communicate with other nodes within their communication range.

In order to analyze the performance of the approaches in situations where it is necessary an energy map with very low error (small threshold) and also when we can tolerate a greater error (big threshold), we changed the value of the parameter threshold. We ran the naive, Markov and ARIMA

Table 1
Default values used in the simulations

Parameter	Value
λ	0.5
Sleep-time	10 s
Sleep-prob	0.7
Event-radius-min	10 m
Event-radius-max	30 m
Event-duration-min	10 s
Event-duration-max	50 s
Dist-line	20 m
State 1-prob	0.01
State 2-prob	0.2
State 3-prob	0.45
State 4-prob	0.34
Threshold	3%
Initial energy	100 J
Communication range	20 m
Time-steps	1 s

algorithms for 200 nodes in a 100×100 m^2 field in which the average degree of each node is 22.7. Fig. 3a shows the average number of energy information packets that each node had to send to the monitoring node, during a simulation of 1000 s, to construct an energy map with an error no greater than the corresponding threshold. These results correspond to an average of these values and a 95% confidence interval. We can see that the Markov approach is better than the other two for all values of threshold. But its performance is very close to the ARIMA model, meaning that both approaches have similar power of prediction for all values of threshold. However, the graph of Fig. 3a is not a fair way of comparing the three approaches because when a node, running the naive algorithm, has to send an energy information packet, the size of the extra information required is only 4 bytes (its available energy). In the Markov algorithm, the overhead is of 8 bytes (its available energy and its current power consumption) and in the ARIMA model the overhead is about 40 bytes (with the parameters p, d, q and the coefficients of the AR and MA models). In order to perform a fair comparison between the three approaches, we have to analyze the average number of bytes that each node has to send when running the naive, Markov and ARIMA algorithms. Thus, the metric used to define energy efficiency will be the number of bytes transmitted. Fig. 3b compares the average number of bytes that each node had to send to the monitoring node if the normal packet size of a sensor network is 30 bytes. In this situation, each time a node has to send its energy information, it will have to send 34 bytes (30 bytes of



(a) Average number of packets.

(b) Average number of bytes when the packet size is 30 bytes.

(c) Average number of bytes when the packet size is 60 bytes.

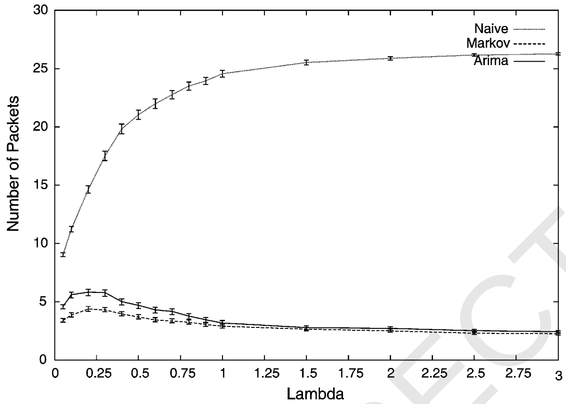
(d) Average number of bytes when the packet size is 120 bytes.

Fig. 3. Comparison between the three approaches when we change the value of the threshold.

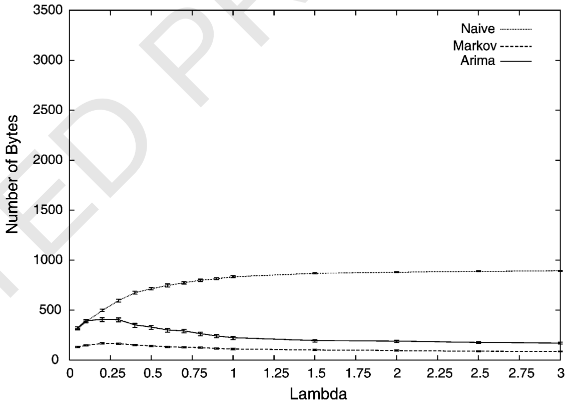
the normal packet plus 4 bytes of the naive overhead) in the naive algorithm, 38 in the Markov and 70 bytes in the ARIMA. We can see that when we compare the number of bytes rather than the number of packets, the performance of the ARIMA is closer to the naive, and the Markov is still the best of the three. Fig. 3c and d show what happens when the normal size of a packet is 60 and 120 bytes, respectively. As the normal packet size increases, the naive becomes even worse because, in these situations, the overhead of the large amount of information required by the ARIMA has a smaller impact in the total number of bytes sent. Thus, for all values of threshold analyzed, the Markov model was more energy-efficient than the other two models, and for sensor networks whose size of the packet is small, the performance of the ARIMA is very close to the naive approach.

Next we altered the value of the parameter λ in order to study the behavior of each approach when the number of events increases. We executed the three approaches using the same scenario described above, during a simulation of 1000 s. Fig. 4a shows the average number of packets when we increase the number of events in the network. In these simulations, the threshold was fixed in 3%. We can see that the power of making prediction of the Markov model is very similar to the ARIMA, but still better for all values of λ. Also, as the network becomes more active, the difference between the number of packets required by the naive and by the prediction-based approaches is getting larger. Nevertheless, as described above, to do a fair comparison, we have to analyze the number of bytes transmitted by each approach. These results are shown in Fig. 4b–d. We can see that the Markov approach is still better than the other two for all values of packet size, and also that when the packet size increases, the difference between the number of bytes transmitted by the prediction-based approaches and the naive one increases. One interesting fact is that the prediction approaches have a better behavior when the number of events is very small or big. The worst case of these approaches happens for medium values of λ. This means that the fact of having more events does not make the problem of prediction more difficult. The more difficult situations for the prediction approaches are when there is a medium number of events. On the other hand, in the naive approach, as more events happen, more energy will be spent by a node and more often it will have to send energy information packets to the monitoring node. Then, the prediction approaches scale well when the number of events increases or, the power of making prediction does not decrease when the activity of the network increases.

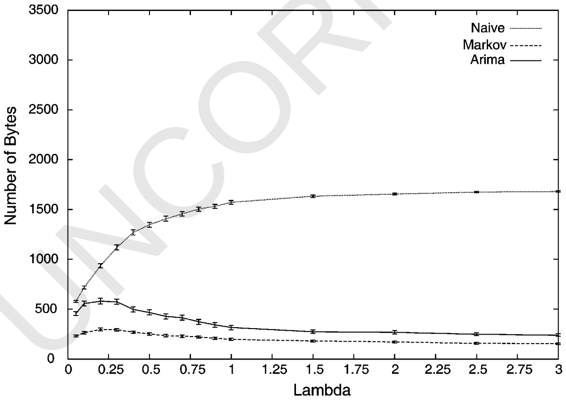
Due to the nondeterministic characteristic of the sensor networks, it is better to perform predictions that are simple



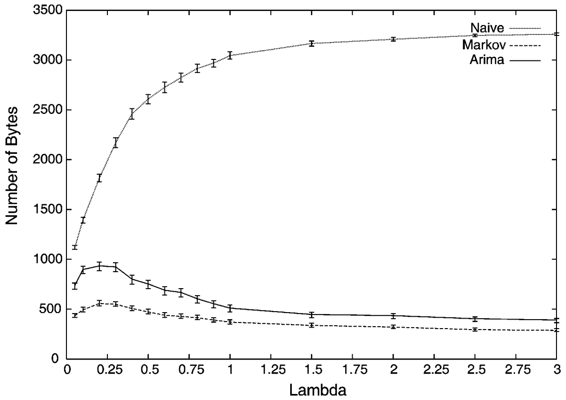
(a) Average number of packets.



(b) Average number of bytes when the packet size is 30 bytes.



(c) Average number of bytes when the packet size is 60 bytes.



(d) Average number of bytes when the packet size is 120 bytes.

Fig. 4. Comparison between the three approaches when we change the value of the parameter λ.

both in terms of the computation required to find the parameters of the prediction model and, mainly, in terms of the number of parameters that have to be sent to the monitoring node. This feature becomes clear when we compare the two prediction techniques. Even though both present similar capacity of making prediction, the Markov approach is better because, in this model, only one parameter describes the energy dissipation in a sensor node, and thus, only the available energy and the current dissipation rate have to be sent to the monitoring node. Thus, in the construction of prediction-based energy maps, it is better to use simple models instead of sophisticated predictions that demand a lot of communication between the sensors and the monitoring node.

5. Related work

5.1. Prediction techniques

The use of prediction techniques is very common in many research areas such as meteorology [4], stock market [5] and biology [6]. In computer networks, prediction algorithms have been used to predict network traffic [26]. The ability to predict traffic patterns within a network is one of the fundamental requirements of network management.

Wireless sensor nodes tend to have very restrict hardware resources. Thus, a prediction algorithm for these networks is required to be simple. This simplicity implies that the processing time in estimating the future energy consumption rate and the number of parameters that have to be sent to the monitoring node cannot pose a heavy burden on the sensor node. Another characteristic that we pursue when choosing a prediction algorithm, is that all computation is done locally. Each node should make its own prediction based only on its past behavior and no communication between neighboring nodes is required.

Our goal when choosing the Markov chain is to have a very simple prediction algorithm based on states, like the energy dissipation model presented in this work. The main idea is that the transitions between states will happen in the future in the same way they happened in the past. As example, if in 30% of the time when a node was in operation mode 1, it went to operation mode 2, it means that when this node will be in state 1, it will go to state 2 with probability 0.3. This prediction technique has two main advantages to WSN:

- The computation of the prediction is simple and it is done locally, since each node computes its power consumption only keeping track of its past state transitions.
- It is suitable for WSNs since, in these networks, the node has to turn off the parts that are not been used to save energy. Thus, nodes can be modeled by states of operation.

We chose the ARIMA model in order to have a more sophisticated technique to be contrasted with the Markov chain. Our goal was to compare a technique to make predictions based on time series with the very simple one based on states. The results showed that the ARIMA is not suitable for WSN due to its complexity in terms of the number of communications.

5.2. Wireless sensor networks

In Refs. [1,12,22,24] the authors explore issues related to the design of sensors to be as energy-efficient as possible. In particular, the WINS [1,22] and PicoRadio [24] projects are seeking ways to integrate sensing, signal processing, and radio elements onto a single integrated circuit. The SmartDust project [12] aims to design millimeter-scale sensing and communicating nodes.

The energy efficiency is the primary concern in designing good media access control (MAC) protocols for aWSN. Another important attribute is scalability with respect to network size, node density and topology. A good MAC protocol should easily accommodate such network changes [29]. In addition, some energy-aware routing schemes have been proposed for WSNs. Directed diffusion [11] is a new paradigm for communication between sensor nodes. In this paradigm, the data is named using attribute-value pairs and data aggregation techniques are used to dynamically select the best path for the packets. This enables diffusion to achieve energy savings. Sensor Protocols for Information via Negotiation (SPIN) [7,14] is a family of adaptive protocols that efficiently disseminate information among sensors in an energy-constrained WSN.

5.3. Energy map generation

The work proposed in Ref. [32] obtains the energy map of sensor networks by using an aggregation based approach. A sensor node only needs to report its local energy information when there is a significant energy level drop compared to the last time the node reported it. Energy information of neighbor nodes with similar available energy are aggregated in order to decrease the number of packets in the network. In Ref. [32], each node sends to the monitoring node only its available energy, whereas in our work each node sends also the parameters of a model that tries to predict the energy consumption in the near future. Thus, in our approach, each node sends to the monitoring node its available energy and also the parameters of the model chosen to represent its energy drop. With these parameters, the monitoring node can update locally its information about the current available energy at each node, decreasing the number of energy information packets in the network.

6. Conclusion and future work

In this work, we have studied the problem of constructing the energy map for WSNs. We analyzed two prediction-based energy maps based on probabilistic and statistical models. In the prediction-based energy maps, each node tries to estimate the amount of energy it will spend in the near future and it sends this information, along with its available energy, to the monitoring node. Using the energy dissipation model proposed in this paper, simulations were conducted in order to compare the performance of the two prediction-based approaches with a naive one, in which only the available energy is sent to the monitoring node. Simulation results indicate that the prediction-based approaches are more energy-efficient than the naive model, and also that these approaches are more scalable with respect to the number of sensing events.

As discussed here, prediction-based techniques are a good approach to construct the energy map for WSNs. We intend to extend this work by examining and evaluating other prediction models for obtaining the energy map.

In Ref. [9], Holzmann points out that protocol design is still much of an art, but more and more we should strive for applying and defining well-established principles and practices. This paper discussed the importance of building an energy map for a WSN since it can be applied to the design of different aspects of this kind of network. Furthermore, it presented how an energy map can be obtained in an efficient way.

Acknowledgements

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. The authors would like to thank the comments and suggestions of Prof. Narayan Mandayam and the members of the DATAMAN group of Rutgers University.

References

[1] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, H. Marcy, Wireless integrated network sensors: low power systems on a chip, In: European Solid State Circuits Conference, The Hague, The Netherlands, October, 1998.

[2] G.E.P. Box, G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1976.

[3] P.J. Brockwell, R.A. Davis, Introduction to Time Series and Forecasting, second ed., Springer, New York, 2002.

[4] R.A. Calvo, H.D. Navone, H.A. Ceccatto, Southern Hemisphere Paleoand Neoclimates: Key Sites, Methods, Data and Models, Chapter Neural Network Analysis of Time Series: Applications to Climatic Data, Springer, Berlin, 2000.

[5] X. Ge. Pattern Matching in Financial Time Series Data. http://www.datalab.uci.edu/people/xge/chart/index.html.

[6] G. Pollastri, P. Baldi, P. Fariselli, R. Casadio, Improved prediction of the number of residue contacts in proteins by recurrent neural networks, Bioinformatics 17 (2001) S234–S242.

[7] W.R. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, In: Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, WA USA, 1999, pp. 174–185.

[8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System Architecture Directions for Networked Sensors, In: Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems, November, 2000.

[9] G.J. Holzmann, Design and Validation of Computer Protocols, Prentice-Hall Software Series, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[10] G.J. Holzmann, Protocol Design: Redefining the State of the Art, IEEE Software 9 (1) (1992) 17–22.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, In: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 2000, pp. 56–67.

[12] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next Century Challenges: Mobile Networking for Smart Dust, In: Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking, Seattle, 1999, pp. 271–278.

[13] P.W. King, Formalization of protocol engineering concepts, IEEE Transactions on Computers 40 (4) (1991) 387–403.

[14] J. Kulik, W.R. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, In: Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking, Seattle, Seattle, WA, August, 1999.

[15] M.T. Liu, Protocol Engineering, in: M.C. Yovits (Ed.), Advances in Computers, vol. 29, Academic Press, San Diego, CA, USA, 1989, pp. 79–195.

[16] Robert F. Nau. Fuqua school of business—Forecasting, 2003, http://www.duke.edu/~rnau/411out03.html.

[17] D. Niculescu, B. Nath, Trajectory-Based Forwarding and its Applications, In: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, San Diego, CA, September, 2003.

[18] NIST. NIST/SEMATECH, e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook, 2002.

[19] ns2. The network simulator. http://www.isi.edu/nsnam/ns/index.html, 2002.

[20] S. Park, A. Savvides, M.B. Srivastava, SensorSim: a simulation framework for sensor networks, In: Proceedings of the Third ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Boston, MA USA, 2000, pp. 104–111.

[21] T.F. Piatkowski, An engineering discipline for distributed protocol systems, in: D. Rayner, R.W.S. Hale (Eds.), Proceedings of IFIP Workshop on Protocol Testing—Towards Proof?, Specification and Validation, vol. 1, Teddington, Middlesex, UK, 1981, pp. 27–29.

[22] G.J. Pottie, W.J. Kaiser, Wireless Integrated Network Sensors, In: Communications of the ACM 43 (2000) 551–558.

[23] R-Project. The R-Project for Statistical Computing, 2002, http://www.r-project.org/.

[24] J.M. Rabaey, M.J. Ammer, J.L. da Silva Jr., D. Patel, S. Roundy, PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking, IEEE Computer 33 (7) (2000) July.

[25] S. Ross, A. First, Course in Probability, Fifth ed., Prentice Hall, Englewood Cliffs, NJ, 1998.

[26] A. Sang, S.q. Li, A Predictability Analysis of Network Traffic, In: INFOCOM, 2000, pp. 342–351.

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

[28] StatSoft. Inc, Electronic Statistics Textbook, StatSoft, Tulsa, OK, 2002, http://www.statsoft.com/textbook/stathome.html.

[29] A. Woo, D.E. Culler, A. Transmission, Control Scheme for Media Access in Sensor Networks, In: The Seventh Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001, pp. 221–235.

[30] W. Ye, J. Heidemann, D. Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, In: Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies, June, 2002.

[31] Y. Yu, R. Govindan, D. Estrin. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department Technical Report, 2001.

[32] Y.J. Zhao, R. Govindan, D. Estrin, Residual Energy Scans for Monitoring Wireless Sensor Networks, In: IEEE Wireless Communications and Networking Conference, March, 2002.