Springer Annals of telecommunication Journal manuscript No.
(will be inserted by the editor)

Looking for network functionalities' extension by avoiding energy-compromised hotspots in wireless sensor networks

J. Rahme · A. Carneiro Viana · K. Al Agha.

Received: 12th March 2008 / Accepted: date

Abstract The vast literature on the wireless sensor research community contains many valuable proposals for managing energy consumption, the most important factor that determines sensors lifetime. Interesting researches have been facing this requirement by focusing on the extension of the entire network lifetime: either by switching between node states (active, sleep), or by using energy efficient routing. We argue that a better extension of the network lifetime can be obtained if an efficient combination of management mechanisms can be performed at the energy of each single sensor and at the load distribution over the network. Considering these two accuracy levels (i.e., node and network), this paper presents a new approach that uses cost functions to choose energy efficient routes. In particular, by making different energy considerations at a node level, our approach distributes routing load, avoiding thus, energy-compromised hotspots that may cause network disconnections. The proposed cost functions have completely decentralized and adaptive behavior, and take into consideration: the end-to-end energy consumption, the remaining energy of nodes, and the number of transmissions a node can make before its energy depletion. Our simulation results show that, though slightly increasing path lengths from sensor to sink nodes, some proposed cost funtions (1) improves significantly the network lifetime for different neighborhood densities degrees, while (2) preserves network connectivity for a longer period of time.

Keywords Static wireless sensor networks · Load distribution · Energy management

1 Introduction

Context. Self-configuring wireless sensors are revolutionizing the way to integrate computing in our daily environment. This is mainly due the fact that they make possible to gather

Joseph Rahme and Khaldoun Al Agha
LRI, Université Paris-SUD XI – Paris
Tel.: +331-69-15-6617
Fax : +331-69-15-6586
E-mail: joseph.rahme@lri.fr, alagha@lri.fr

Aline Carneiro Viana
ASAP/INRIA Saclay - Ile de France sud, France
E-mail: aline.viana@inria.fr

and to process information in ways not previously possible [7]. Beside this feature, they include data accuracy, flexibility, cost effectiveness, and ease deployment characteristics. As a consequence, sensor-based networks play an important role in the design of applications whose aim is surveillance, data-gathering, or monitoring. It consists in deploying a large number of sensors to execute a determined task in a specified geographic area. The task can be the monitoring of specific events or the tracking of targets within the area of interest. Sensor-based networks have thus, attracted the attention of civil, medical, and military domains, justifying the numerous research in the wireless sensor area.

It is usual to consider application scenarios where sensors are deployed in regions of difficult access, and/or human intervention is not feasible. In this scenarios, self-organization is a particularly important attribute for the autonomy dimension of the network. This requires the network to be able to organize/configure by its own self in order to solve problems such as routing, load balancing, or energy consumption.

Motivation. Despite the recent advances in electronics, numerous constraints are still imposed on sensors devices and especially on their energy. This fact makes the proposal of energy optimization mechanisms an important requirement. In this context, an important question raises: *how energy consumption can be managed in order to increase network lifetime?* This is the topic addressed in the paper.

The vast literature on the wireless sensor research community contains many valuable proposals for managing energy consumption. Recently, interesting researches have been facing this requirement by focusing on the extension of the entire network lifetime. In a global point of view, these researches :

- switch nodes' energy level between sleep and awake states [2,4,6,14,17,18,21] or
- by keeping nodes in the active state, perform power control [1,3,11] or energy-aware routing [5,9,13,19,20,22].

Despite having clearly defined outlines and presented good solutions, those works deal with the network lifetime's extension problem (*1*) by reducing the energy consumption at each single sensor (*i.e.*, at a node accuracy level) **or** (*2*) by assuring a homogeneous load distribution over the network (i.e., at a network accuracy level). Section 5 gives a detailed review of these works.

Contributions. Instead, our approach takes into account both: the overall energy consumption and the load distribution over the network. By considering those two accuracy levels (i.e., at the node and at the network scope), this paper discusses different self-configuring and adaptive approaches that uses cost functions to determine energy efficient routes. By making different energy considerations at a node level, our approaches distribute routing load, avoiding thus, energy-compromised hotspots that may cause network disconnections. In addition, the end-to-end energy consumed when sending a packet is minimized. So, different from the previous approaches, cost functions englobe what is needed to increase network lifetime.

In summary, the contributions of this paper are twofold:

- a set of self-configuring cost functions used to determine energy efficient routes and to optimize energy consumption;
- a method allowing to distribute routing load over the network and to avoid energy-compromised hotspots nodes.

The proposed cost functions have completely decentralized and adaptive behaviors and take into consideration: the end-to-end energy consumption, the remaining energy of nodes, and/or the number of transmissions a node can make before its energy depletion. We perform exhaustive comparison between the proposed cost functions and the Dijkastra shortest path algorithm. In particular, by using two different scenarios and simulation stop criteria, our simulation results show that, though slightly increasing path lengths from sensor to sink nodes, some cost functions (1) improves significantly the network lifetime for different neighborhood densities degrees, while (2) preserves network connectivity for a longer period of time.

Outline. The paper is organized as follows. Section 2 introduces our system model. We present our proposal by introducing the cost functions in Section 3. Performance results are presented in Section 4. In Section 5, we present a review of the main related works by providing a general classification of existent approaches. Finally, Section 6 concludes this paper and discusses future works.

2 System Model

We will target a general application scenario where the n sensor nodes are randomly deployed in a zone of interest of difficult access and/or where human intervention is not feasible. The considered scenario has then, a finite set of n nodes, each uniquely identified. We consider that sensors form at the beginning, a connected network.

Nodes are all *equal*, in the sense that they have the same attributes, i.e. computational, memory, and communication capabilities. We do not consider Byzantine failures, so nodes may only go out of the system when their battery goes off. Regarding energy, a node may only be in the active state. That is all the nodes in the network are active until their depletion and they all have the same energy when they are deployed.

Each node has the same radio communication range r that allows it to communicate by broadcasting messages. Thus, a node i is able to directly communicate wirelessly with a subset of nodes that are located in the transmission range r_i, and no obstacles interfere with the communication – we refer to that subset as the *neighbors* of the node i. We assume *bidirectional* communications: for any nodes i and j, if i can communicate with j, then j can communicate with i. We consider that sensing and communication ranges are equal. No synchronization is required.

Table 1 Parameters summary.

Parameter	Description
E_{TX}	Energy consumed at a packet's transmission by source nodes.
E_{RX}	Energy consumed at a packet's reception by 1-hop neighbors.
E_I	Energy consumed due the interference caused by a 2-hop neighbor transmission.
$E_r(i)$	Remaining energy at the node i.

We consider a simplified MAC layer where neither messages losses, nor collisions, nor duplications are considered.

Our energy model uses the parameters described at the Table 1. In particular, we consider a 2-hop interference model. When a node i transmits a packet, it consumes an energy E_{TX}

to code and transmit the packet. All the nodes existing 1-hop away from the emitting node i, i.e. $neighbors_i$, receive the packet and decode it. The nodes in $neighbors_i$ that are not the destination, receive the packet, consume E_{RX} energy to decode it, and then, discard the packet. The 2-hop neighbors of the transmitting node receive an non-intelligible signal. This reception makes these nodes to consumes E_I energy.

3 Our proposal

In sensor network, the nodes use batteries as their source of energy. Batteries, however, constitute a limited source of energy. In large-scale sensor networks, nodes are often deployed in hostile environment. If nodes batteries deplete, the possibility of their replacement is almost impossible. Moreover, in case the nodes are accessible, replacing their battery is not always feasible if large networks are considered. In this case, the optimization of nodes' energy consumption is essential to extend network's lifetime. The switch of node state between active and sleep seems an interesting approach but presents a major challenge in decentralized systems like WSNs:
How to determine the duty cycle of nodes and still guarantee connectivity without requiring a global knowledge of the network or a centralized management?

Instead, our proposal considers that nodes are always in the active state. In addition, to optimize the energy consumption in the network, our proposal implements an energy efficient routing that chooses routes based on energy-based weights associated to links. At the following, we briefly describe this routing mechanism and then provide a detailed description of how links are associated to energy-based weights.

3.1 Energy efficient routing

We propose using a *Modified Shortest Path* algorithm to determine energy efficient routes among nodes in the network. Energy efficiency is gotten through energy-based cost functions. In particular, the proposed cost functions assign weights to links between a node and its 1-hop neighbors. Routing is then performed by selecting links that better optimize the locally computed link weights. This optimization will depend on the kind of cost function used. For instance, considering a cost function that only consider the energy consumed by nodes after a transmission, the routing protocol will select the route that minimize the total energy consumed by links between a source and a destination.

The next sections introduce the different cost functions to associate weights to links and detail how energy efficient routing is performed. Each of them considers distinct but dependent nodes' energy-related parameters. Since the links' weights are updated each time a transmission is performed, routing load is adaptively distributed among links that present better energy levels. In addition, energy-compromised hot spots are detected and consequently avoided, before packet transmissions.

3.2 Energy-based cost functions

This section presents and discusses the proposed cost functions, named:

1- $E_{\theta_1}(i)$: considers the amount of energy consumed by a emitting node i and its 1- and 2-hop neighbors, when i performs a packet's transmission.

2- $E_{\theta_2}(i)$: considers the remaining energy of node i and its 1- and 2-hop neighbors.
3- $\omega(i)$: considers the maximal number of transmissions that node i can perform before node i, or one of its 1- or 2-hop neighbors dies.
4- $E_{\theta_4}(i)$: considers the amount of energy consumed due to node i's transmission, while taking into account the remaining energy of intermediate nodes forming the route.

For each proposed cost function, we will describe its characteristics by analyzing its advantages and constraints. It will be easily to observe that, after the description of the 1st cost function $E_{\theta_1}(i)$, each new introduced function in fact proposes new improvements to better distribute routing load. As will be shown in Section 4, the selection of the best cost function will indeed depend on the considered network lifetime assumptions and application requirements.

3.2.1 Regarding energy consumption – 1st cost function:

According to the 2-hop interference model described in the Section 2, when a node transmits, all its 1-hop neighbors consume energy to decode the packet. Therefore, energy consumption for a transmission is proportional to the number of neighbors. Having this in mind, we introduces the cost function E_{θ_1}, which prevents nodes with a lot of neighbors to participate in the routing procedure. This is due the fact that their energy consumption after a transmission, may represent a significant amount for the network lifetime. E_{θ_1} is thus, used to assign weights between a node and its 1-hop neighbors: the weight of the link (k,i) between k and any 1-hop neighbor i is equal to E_{θ_1} of node k. $E_{\theta_1}(k,i)$ is defined as:

$$E_{\theta_1}(k,i) = E_{TX} + \sum_{n_1 \in N_1(k)} E_{RX} + \sum_{n_2 \in N_2(k)} E_I \tag{1}$$

, where

- $N_1(k)$ is the set of 1-hop neighbors of node k;
- $N_2(k)$ is the set of 2-hop neighbors of node k;
- and E_{TX}, E_{RX}, and E_I are described at the Table 1.

In fact, cost function E_{θ_1} calculates the impact a node's transmission will have on the energy of the network, *i.e.,* the amount of energy consumed by the emitting node and its 1- and 2-hop neighbors.

One important point to remark here is that the total energy consumed for routing a packet p from a source to a final destination is additive, representing the amount of energy consumed by the network to route the packet p. Thus, since E_{θ_1} is the energy consumed for a packet's transmission, the whole energy consumed to route the packet to its final destination is the sum of the link weights (E_{θ_1}) forming the route. Therefore, a simple shortest path algorithm using E_{θ_1} as a metric, can easily find an energy efficient route. In particular, the weight of a route between two nodes exchanging packets is the sum of intermediate links' weight forming this route. The route having a minimal sum of weights is then, the selected optimal route given by the modified shortest path algorithm described in Section 5.1.

Moreover, E_{θ_1} enables the shortest path algorithm to avoid nodes in the network that, if used for routing, will waste a lot of energy. Looking at the formula of E_{θ_1}, it is evident that E_{θ_1} gives a high weight for the nodes with a lot of neighbors, which can be seen in the following parts of the formula: $\sum_{n_1 \in N_1(k)} E_{RX}$ and $\sum_{n_2 \in N_2(k)} E_I$. In a dense area, the value of these two parts will be high, which will contribute to the assignment of a high weight to the node k. Hence, nodes with high neighborhood densities will have a low probability of participating to the route determined by the routing protocol.

Despite having the interesting property of minimizing network disconnections, the presented cost function does not consider the remaining energy of nodes. In particular, E_{θ_1} only considers the energy consumed for transmission E_{TX}. This means that a node with few neighbors and a remaining energy that is insufficient for performing one packet's transmission can still be selected as next-hop.

3.2.2 **Regarding remaining energy – 2nd cost function:**

The second cost function E_{θ_2} takes into account the remaining energy of a node and of its 1- and 2-hop neighbors. The cost function E_{θ_2} is as follows:

$$\begin{aligned} E_{\theta_2}(k,i) = \min\{&(E_r(k) - E_{TX}),\\ &min_{n_1 \in N_1(k)}(E_r(n_1) - E_{RX})\\ &min_{n_2 \in N_2(k)}(E_r(n_2) - E_I)\} \end{aligned} \tag{2}$$

, where

- $E_r(k)$ is the remaining energy of node k;
- $E_r(n_1)$, $E_r(n_2)$ are the remaining energy of the 1-hop and 2-hop neighbors affected by the transmission of node k, if k is the selected next-hop;
- E_{TX}, E_{RX}, and E_I are the consumed energy as described at the Table 1.

The function E_{θ_2} is thus used to calculate links' weights between a node k, which has the packet to transmit, and its 1-hop neighbors. Thus, for a link (k,i) where i can represent any 1-hop neighbor of k, the weight is equal to $E_{\theta_2}(k,i)$. By considering the remaining energy of node k after its transmission, *i.e.,* $(E_r(k) - E_{TX})$, node k avoids selecting a next-hop node with a minimum remaining energy to participate in the routing of a packet. By consequence, only links with the highest weights (*i.e.,* nodes with highest level of remaining energy) will compose the determined route. The others parts of the cost function described in (2) gives the minimum remaining energy at $1-$ and $2-hop$ neighbors after a transmission, for each possible next-hop.

In summary, the use of E_{θ_2} to assign a weight for a link between two nodes, allows us to find a route that uses nodes with a high level of remaining energy. This insures an homogeneous consumption of nodes energy, preventing the case where some nodes deplete their batteries before others. Nevertheless, E_{θ_2} does not consider neither the overall energy consumption of the route selected by the algorithm nor the number of hops. In this way, it may result in *longer routes*, which by consequence, makes more nodes in the network to consume energy in the routing procedure.

Since the remaining energy is not an additive metric, routes that maximizes the sum of the weights resulted from E_{θ_2} can not be considered at the energy efficient route's computation. Therefore, a *shortest-widest routing algorithm* (widest in term of remaining energy) is considered.

The shortest widest algorithm chooses among all the routes between a source and a destination, the one where the minimum remaining energies of intermediate nodes is maximal. More specifically, the weight of a route is the minimum weight among intermediate links connecting the source to the destination and the shortest-widest route is the route with the maximum weight. If multiple routes have the same maximum weight, the shortest-widest

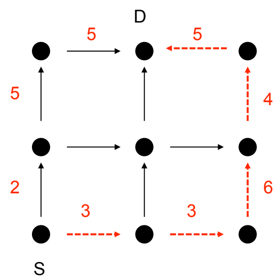

Fig. 1 Example of the selected shortest widest route, represented by the dashed links.

routing algorithm chooses the one with the minimum number of hops. In this way, the algorithm tries to minimize the number of hops from the source to the destination and still keeps a maximal gain in remaining energy.

Despite this, the number of hops is not considered at the weight computation of E_{θ_2}, which still results in long routes being determined by the shortest-widest routing algorithm, even if the nodes' remaining energy is decreased. As an example, consider the Figure 1. In this figure, the weight of each link is calculated using E_{θ_2}. The dashed links constitute the route determined by the shortest-widest routing algorithm. In the example, this route has the maximum minimum weight among the two considered routes between the source and the destination. In particular, the route represented by weights $\{2,5,5\}$ in the example, has 2 as minimum weight, and the route represented by weights $\{3,3,6,4,5\}$ has 3 as minimum weight, which represents the maximum minimum weight among these routes.

The next section introduces a third cost function that tries to solve the problem of long routes.

3.2.3 Regarding number of transmissions – 3rd cost function:

Despite to also consider the remaining energy of nodes, the third cost function $\omega(i)$ uses a strategy different from the previous functions. $\omega(i)$ calculates the weight of a link (i,j) between node i and any 1-hop neighbor j, as following:

$$\begin{aligned}\omega(k,i) = \min\{&\tfrac{E_r(k)}{E_{TX}},\\ &min_{n_1\in N_1(k)}\tfrac{E_r(n_1)}{E_{RX}}\\ &min_{n_2\in N_2(k)}\tfrac{E_r(n_2)}{E_I}\}\end{aligned} \tag{3}$$

, where $E_r(x)$, E_{TX}, E_{RX}, and E_I represent the energy level as previously explained for the Equation 2.

$\omega(k,i)$ uses the ratio $\frac{E_r(k)}{E_{TX}}$ to determine the remaining energy level of node k. Thus, besides indicating the energy of the node, the ratio $\frac{E_r(k)}{E_{TX}}$ also represents the maximal number of transmissions that the node can perform. For example, a ratio equal to n means that the node remaining energy is $n \times E_{TX}$ and that it can still transmit n packets before having its battery off. In the same way, the ratio $\frac{E_r(n_1)}{E_{RX}}$ indicates the number of packets a node can receive before its depletion. And finally, the ratio $\frac{E_r(i)}{E_I}$ determines the number of non-intelligible packets a node can receive before depleting all its energy.

Combining these three ratios and computing their minimum will result in the $\omega(k,i)$ metric. In particular, this metric describes the minimum number of transmissions node k can execute before it or a node in its 1- or 2-hop neighborhood looses all their energy.

Finally, as described in Section 3.2.2, to find an energy efficient route between a source and a destination, we use the modified shortest-widest route algorithm with $\omega(k,i)$ as the cost function. Thus, the weight of a route between two nodes is the minimum weight among intermediate nodes forming this route, being the resulted shortest widest route the route with that maximum weight.

The use of $\omega(k,i)$ allows an homogeneous load distribution over the network by avoiding nodes with low remaining energy. Nevertheless, like the previous cost function E_{θ_2}, $\omega(k,i)$ will result in routes with a significant remaining energy without to consider their consumed energy for routing packets from the source to the destination.

3.2.4 Regarding consumed and residual energy – 4th cost function:

As discussed in Sections 3.2.2 and 3.2.3, the nodes' energy consumption, as well as their remaining energy are important parameters to be considered in route selection. This section introduces thus, the cost function E_{θ_4} that considers both of these parameters.

The first cost function E_{θ_1} reduces the end-to-end energy when used in packets routing and thus increases network's lifetime. But, this method cannot prevent the participation of nodes with low remaining energy in the routing process. Hence, intermediate nodes with low remaining energy but that generate less energy consumption can be selected to participate in the routing procedure.

The source and the destination of a flow are the only nodes that even though they don't have high residual energy, must participate in the routing process when searching for an energy efficient route. So, the method that takes into account the residual energy of nodes in the routing process must operate on intermediate nodes.

Another intrinsic characteristic of $E_{\theta_1}(k,i)$ is the assignment of the same weight for links between a node k and any of its 1-hop neighbors, since the energy consumption model is the same.

By considering all these factors, the proposed cost function E_{θ_4} uses a weighting function to differentiate between neighbors with low and high residual energies. This weighting function have the following form:

$$E_{\theta_4}(k,i) = c_f \times \frac{E_{\theta_1}(k,i)}{E_r(i)}, \forall i \in N_1(k) \tag{4}$$

, where c_f is a weighting factor, corresponding here to the total remaining energy of nodes, and $E_r(i)$ is the remaining energy of the k's 1-hop neighbor i.

This approach permits to distinguish between two nodes with different energy remaining and the same neighborhood density (same E_{θ_1}): the node with greater energy remaining will participate in the routing of a packet. In addition, it allows to avoid the nodes with high neighbors density and low remaining energy. The packet may then be routed through links that $min\{E_{\theta_4}\}$.

As described in Section 4, simulation results using functions E_{θ_1}, E_{θ_2}, and ω show the number of hops is directly related to the energy consumed by routes. Thus, combining E_{θ_4} with a constraint based on the number of hops seems very crucial to further extend network's lifetime. In this way, besides of considering the route that minimizes E_{θ_4}, we plan to also reduce the number of hops of the selected route. This condition on the number of hops allows to reduce the data delivery delay in the case where two interesting routes with the same weight cost are found. Our problem will then have the following form:

$$Problem : \begin{cases} min\ E_{\theta_4} \\ s.t.\ min\ (\#hop) \end{cases} \tag{5}$$

It is expected a route selection that minimizes the energy consumption, preserves the nodes' residual energy, and still limit the number of hops. By preserving the residual energy of nodes in routes, that approach also allows to avoid shortest paths in terms of number of hops, and by consequence, the generation of hot spots.

4 Performance evaluation

The section describes the experiments we have conducted to assess both the performance and the accuracy of the proposed approaches.

4.1 Experimental design and metrics

We have performed experiments by simulation in order to better evaluate the proposed cost functions. In our experiments, we used a homemade C++ simulator. Note that, as we are mostly interested in the energy consumption and connectivity evaluation, our simulator deliberately does not model all the details of a realistic MAC protocol. Instead, we considered a simplified MAC layer where neither messages losses, nor collisions, nor duplications are considered[1].

Nevertheless, as described in Section 2, besides to consider the energy consumed for message transmission and reception, our simulator also consider the energy consumed by a node due to interferences, in particular, the energy consumed by the 2-hop neighbors of an emitting node. For every experiment, the network is composed of 20, 50, 70, 100, 200, 300 and 400 static nodes randomly distributed over a square area of 100 meters on a side. For each network size and in order to guarantee connectivity, we consider the range r of nodes as given by Gupta and Kumar in [8]. In particular, for n nodes uniformly placed at random in a rectangular area $[a,b]$ and $\pi r^2 \geq ab\frac{\log n + c(n)}{n}$, the resulting network graph is asymptotically connected with probability one, if and only if $c(n) \rightarrow \infty$ as $n \rightarrow \infty$. We have used $c(n) = \frac{3}{\pi}$ in our simulations.

Our simulator, is a discrete time-based engine in which the network lifetime is considered as a series of rounds. A round represents the arrival of an event in the network which is implemented by the routing of the packet generated by a source to a destination. Thus, the network lifetime is described here by the number of routes determined during the total simulation time. We consider that at any time, only one event can occur in the network (i.e., no parallel transmission is performed).

We consider that all nodes have the same initial remaining energy estimated to 5000 unities (u). The power consumption for each node state is shown in Table 2. The model of interference is the one described in Section 2. The energy of the nodes in the network is updated after the routing of each packet.

[1] We believe that, even if these issues may cause longer convergence time, they will not affect the correct execution of the proposed algorithms. A detailed study of their impact on our approaches constitutes a future work.

Table 2 Energy consumption

Node state	Energy consumption
Transmission	$1.3u$
Reception	$0.9u$
Interference	$0.4u$

In order to better evaluate the characteristics of each proposed cost function, we have considered two criteria to compute the network lifetime, and by consequence, two simulation stop conditions:

- *1st stop condition:* when the first node in the network depletes its energy [10] ;
- *2nd stop condition:* when the 1-hop neighbors of all the sinks are depleted.

In the 1st condition, two communication scenarios are considered. In the *communication scenario 1*, sources and destinations are randomly chosen among sensor nodes in the network, being no special node selected as sink. In the *communication scenario 2*, sources send packets to determined sinks in the network. For this, we consider that the network area is divided into two zones, where one randomly positioned sink exists per zone. Nodes route packets to the closet sink, and if no more routes exist to it, the second sink is then considered as destination. For both communication scenario, the simulation is stopped when the first node in the network dies. Network lifetime is then determined based on the number of determined routes (*i*) for arbitrarily selected pairs of sources and destinations (in scenario 1), and (*ii*) for arbitrarily selected sources and their corresponding sinks.

Otherwise, in the 2nd simulation stop condition, only the communication scenario where sources send packets to one of the sinks positioned in the network, is considered (see previous description of the communication scenario 2). The simulation is stopped when all the 1-hop neighbors of both sinks are depleted.

The results obtained for each one of the simulation stop conditions and the respectively communication scenarios, are described in the following sections.

4.2 When the 1st node dies

4.2.1 Communication scenario 1

Figure 2 compares the different cost functions with Dijkstra's algorithm. The figure shows for different network sizes, the number of rounds a network can support before the depletion of the first node. At every round, only one event can happen, and parallel transmissions are not considered. Therefore, the network lifetime can be considered as the number of routes successfully determined before the depletion of a node in the network. We vary the number of nodes in the network, which also imposes different nodes densities.

The results show that for all node densities, the shortest path algorithm gives low performance compared to the results obtained with E_{θ_1}. This is expected because E_{θ_1} finds the route that consumes the minimum energy in contradiction with Dijkstra. Since Dijkstra's algorithm uses the minimal number of hops to attend a destination, it tends to put the major load on the nodes situated at the center of the network. Consequently, this depletes the energy of nodes located in dense regions, violating the homogeneous distribution of the energy

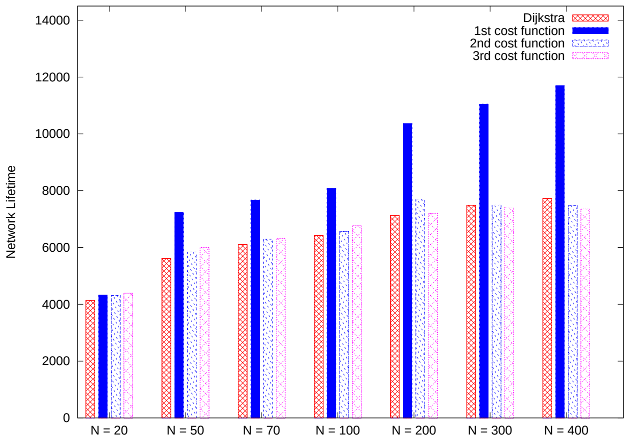

Fig. 2 Network lifetime with different cost functions and node densities, when communication scenario 1 and the 1st simulation stop condition are used.

consumption and increasing the probability of network partitioning. Instead, E_{θ_1} increases the lifetime of the network by:

- avoiding nodes that generate a high energy consumption in the network (nodes with large number of neighbors);
- minimizing the sum of the energy used to route the packet from source to destination.

For low node density ($N = 20, 50, 70, 100, 200$), it can also be observed that E_{θ_2} and $\omega(i)$ slightly increase the lifetime of the network when compared to Dijkstra. In addition, for a very low node density (*i.e.,* for $N = 20$), E_{θ_2} and $\omega(i)$ surpass E_{θ_1}.

Since E_{θ_2} and $\omega(i)$ use the remaining energy in the calculation of links' weights, the network load is distributed over nodes with high remaining energies. This only increases the lifetime in low dense networks because the extend in routes length is not significant. For high node density (*i.e.,* for $N = 100, 200, 300, 400$) the route length increases dramatically (as shown in Figure 4.2.1), which impacts the energy consumption in the network and decreases the network lifetime.

Figures 4.2.1 and 4 respectively show the average number of hops and the average consumed energy per route for different nodes densities. The figure shows high values for E_{θ_2} and $\omega(i)$ when the network size increases. More specifically, in Figure 4.2.1, for $N = 400$, the number of hops for E_{θ_2} and $\omega(i)$ is very high compared to Dijkstra. This consequently explains the high consumed energy showed in Figure 4 and the network lifetime decreasing showed in Figure 2. It is however, interesting to observe the strong load balancing characteristics of these cost functions. Considering again the 400-node network case, E_{θ_2} and $\omega(i)$ result in almost the same network lifetime compared to Dijkstra, even using routes with the double of the Dijkstra size and consuming the double of the Dijkstra consumed energy.

Since E_{θ_1} gives the best performance compared to the other cost functions, we only provide comparison between E_{θ_4} and E_{θ_1}.

In Figure 5, we notice that for N =20 and 50, the network lifetime using the first and the fourth cost function is approximately the same. This result is expected since the number of

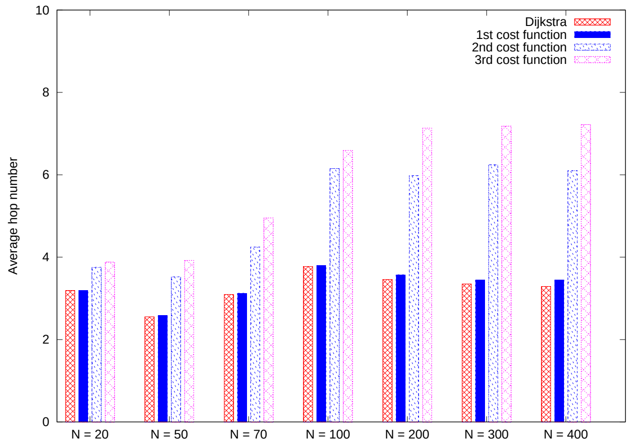

Fig. 3 Average hop number per route for different cost functions and node densities, when communication scenario 1 and the 1st simulation stop condition are used.

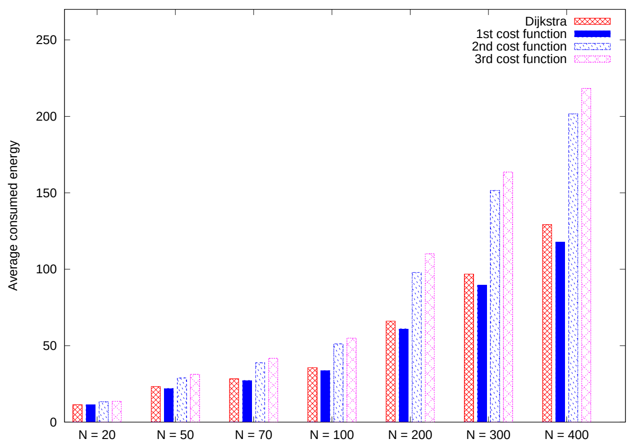

Fig. 4 Average energy consumed per route for different cost functions and node densities, when communication scenario 1 and the 1st simulation stop condition are used.

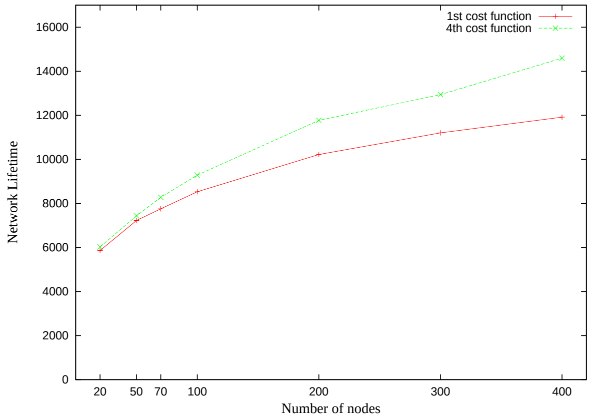

Fig. 5 E_{θ_1} and E_{θ_4}'s network lifetime comparison, when communication scenario 1 and the 1st simulation stop condition are used.

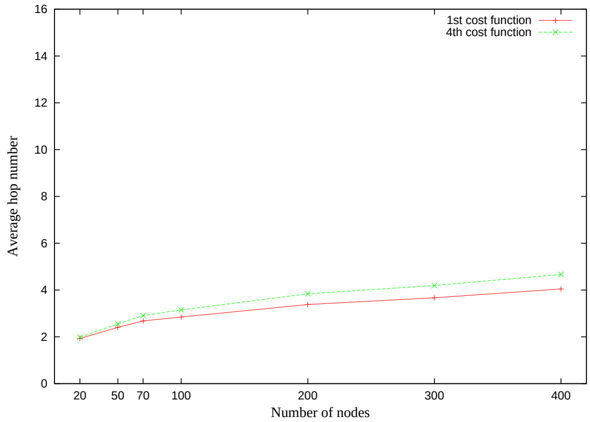

Fig. 6 E_{θ_1} and E_{θ_4}'s average hop by route comparison, when communication scenario 1 and the 1st simulation stop condition are used.

nodes is limited and the routing algorithm do not have many alternative nodes to be use. For this reason, the average hop by route shown in Figure 6, and the average energy consumed per route, shown in Figure 7, are also similar. This shows that the two cost functions use almost the same routes.

Nevertheless, we can clearly see that E_{θ_4} outperforms the first cost function for dense networks, since the weighting part of the function $\frac{c_f}{E_r(i)}$ forces the routing to choose nodes with higher remaining energies.

In fact, E_{θ_4} uses the same routes as E_{θ_1} but avoids the nodes with low remaining energy. In dense networks, this will slightly increase the average number of hops per route (see Figure 6) resulting in an increase in the energy consumed by route (see Figure 7). In summary, with a little increase in energy consumed per route, the fourth cost function dramatically increases the network lifetime. This makes E_{θ_4} the best cost function to be implemented when the 1st simulation stop condition and the communication scenario 1 are considered. We will see in the following that the same is valid for the 2nd communication scenario too.

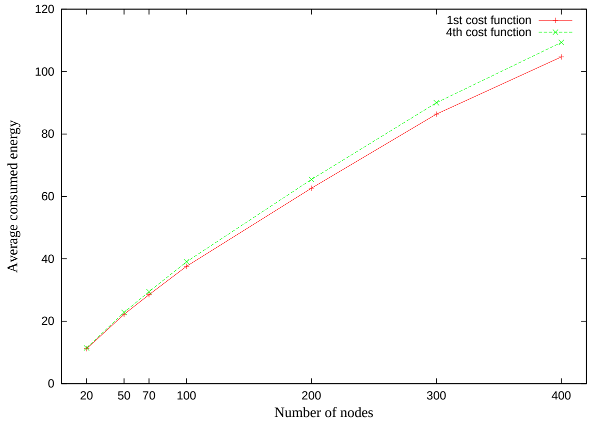

Fig. 7 E_{θ_1} and E_{θ_4}'s average consumed energy per route comparison, when communication scenario 1 and the 1st simulation stop condition are used.

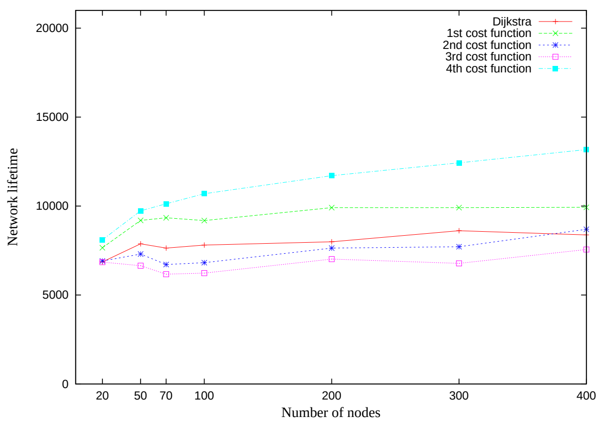

Fig. 8 Network lifetime as a function of the network size, when communication scenario 2 and the 1st simulation stop condition are used.

4.2.2 Communication scenario 2

We now consider scenario where the network area is divided into two zones, being one sink randomly positioned in each zone. The simulation is stopped when the first node depletes in the network.

Figure 8 represents the lifetime of the network using the different cost functions. Since we consider that route can be determined in parallel, network lifetime can be represented as the number of routes before the depletion of the first node in the network. We observe that the fourth cost function outperforms all the other cost functions for all network sizes. Since E_{θ_4} uses the same cost function as E_{θ_1}, it reduces the end to end energy consumed in addition to avoiding the nodes with low residual energy. Again, this is mainly due to the weighting part of the function $\frac{c_f}{E_r(i)}$, which increases the weight of a given link (i,j) if the 1-hop neighbor j of i is energy limited. As a result, links with high weights will be omitted from the calculated route.

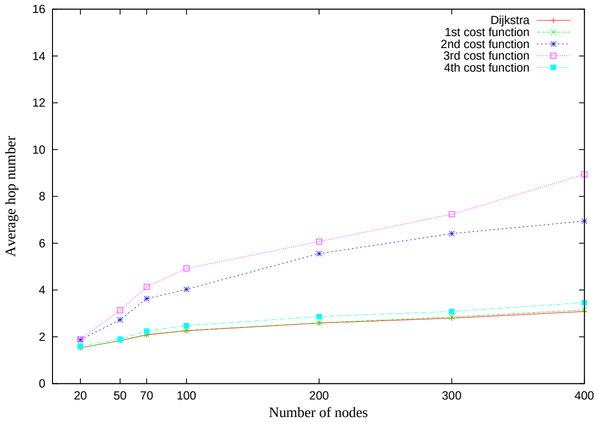

Fig. 9 Average hop number per route as a function of the network size, when communication scenario 2 and the 1st simulation stop condition are used.

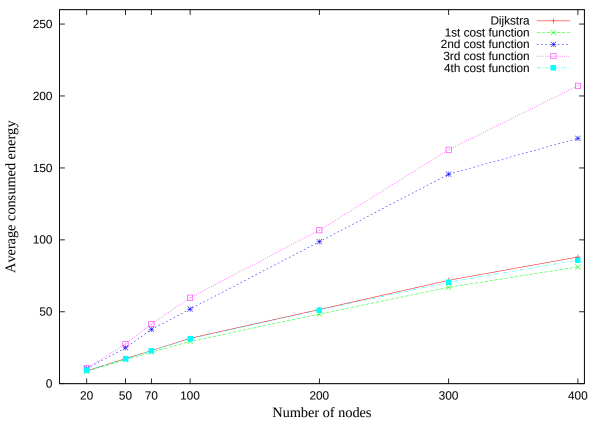

Fig. 10 Average consumed energy per route as a function of the network size, when communication scenario 2 and the 1st simulation stop condition are used.

Figure 9 shows the average hop number per route as a function of the network size. As previously observed in communication scenario 1 (see Figures 6 and 7), E_{θ_4} increases slightly the number of hops compared to E_{θ_1}, also resulting in a little increase in the average energy consumed per route (see Figure 10).

We have, however, verified that contrarily to Dijkstra where the first node to died is in the center of the network (consequently, in a dense region), for those two cost functions, the first depleted node is closer to the border or is on the border of the topology. This is an interesting property to be considered, specially in cases where network lifetime is considered as *the maximum network operational time before the first disconnection with any sink node happens* (as defined for the 2nd simulation stop condition).

4.3 When the 1-hop neighbors of all the sinks dies

In this section, we consider the second scenario where the network possesses two sinks and nodes' collected information is sent to their closest sink. We simulate the case where all nodes have information to send in a given period: all the nodes send packets in a sequential way in this period. If there is no route from a node to its closest sink because all the sink's 1-hop neighbors are depleted, the sensor routes the packet to the other sink. The simulation is stopped when the 1-hop neighbors of both sinks are depleted which means that the network is partitioned.

Figure 11 shows the lifetime of the network using the four cost functions and Dijkstra. We can notice that Dijkstra algorithm increases network lifetime compared to the fourth cost function. This result can be explained as follows. It is easy to understand that the neighbors of the sink are the most used nodes because they always participate in relaying the packets to the sink. As a result, they are the first nodes to deplete in the network. Since E_{θ_4} avoids nodes with a lot of neighbors and low remaining energies, when the sink's neighbors begins to deplete, the length of the routes increases because the routed packets have to bypass the nodes with limited energy to reach the sink. This depletion can be seen in Figure 12 between round 11778 (first node dies) and round 13337 (30 nodes are depleted), which generates a semi-partition of the network. After the corresponding peak between round 11778 and round 13337 in Figure 13, the number of hops becomes stable again, since another partition of the network close to the sink can still send packets.

If we compare Dijkstra algorithm to E_{θ_4}, the first node to die happens earlier as can be noticed in Figure 12 because Dijkstra do not use any energy metric. But, even if Dijkstra algorithm uses the shortest path, when the major number of sink neighbors are depleted (as seen in Figure 12, between round 12000 and 15000), this immediately increases the number of hops per route (as observed in Figure 13), resulting in more energy consumption.

To continue the analysis of the Figure 11, we can conclude from Figure 12 that the first cost function distribute the charge better that the three others, since the curve is on the convex hull of the others. Moreover, Figure 13 shows that this cost function do not increase the average number of hops per route. Since E_{θ_1} only considers the end-to-end energy consumed, it will avoid the nodes with a lot of neighbors without increasing the length of the routes when the neighbors of the sink begins to deplete as the fourth cost function does.

5 Related Work

This section discusses the works in the literature related to the energy management in wireless sensor networks. Moreover, at the following sub-sections, we provide a general classification of these works into three different categories. These categories are the following: energy efficient routing, power control, and the management of nodes activity by state switching.

5.1 Energy efficient routing

We discuss here the works that, in order to increase the network lifetime, proposes to perform routing by considering the energy consumed by nodes in the network. In particular, they intend to determine paths that optimize that energy.

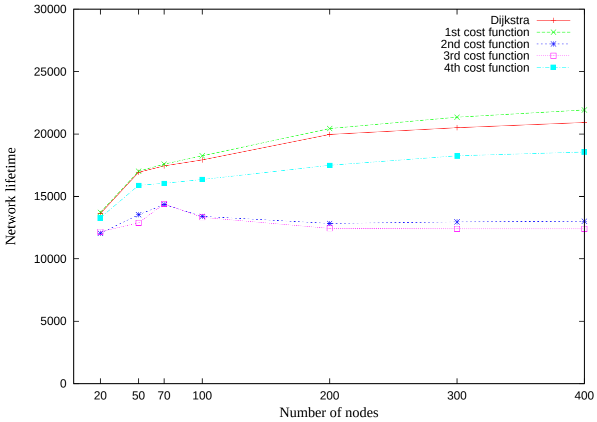

Fig. 11 Network lifetime as a function of the network size, when communication scenario 2 and the 2nd simulation stop condition are used.

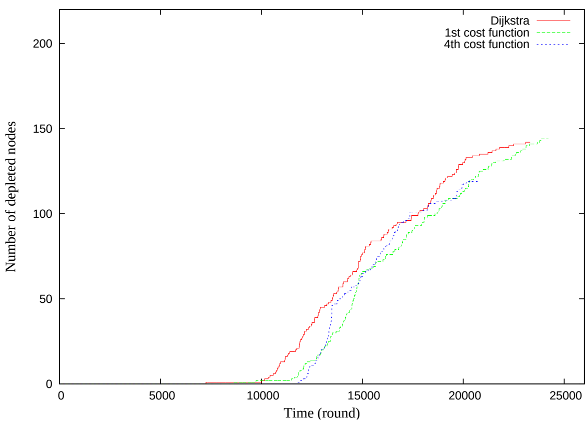

Fig. 12 Number of depleted nodes as a function of the simulation time. The first depleted node approximately arrives at the 11000 simulation round.

In [13], Kwon *et al.* propose a routing protocol to find a route that minimizes the energy consumption of a flow. They thus, calculate for each link in the network, the increment ΔE in energy dissipation resulting from the routing of a flow. A route between two nodes is calculated using a shortest path algorithm with the increment ΔE as the weight of the links. This proposal, however, does not guarantee an end-to-end energy optimization, as one of our cost functions do (presented in Section 3.2.1), and does not take into account the remaining energy of nodes (described in Section 3.2.2).

Authors in [19,5], propose a reactive and multi-routing protocol that uses the remaining energy in the node to improve network lifetime. In [19], routes are selected using a *cost* that depends on the remaining energy of intermediate nodes. The probability of using a route for a flow is inversely proportional to its cost. Thus, contrarily to our approach, authors do not take into account the energy dissipated by interferences, which makes it not realistic. In [5], each node constructs a vector containing the remaining energy of every intermediate node. A route is then considered shorter than another if it contains a node with minimal

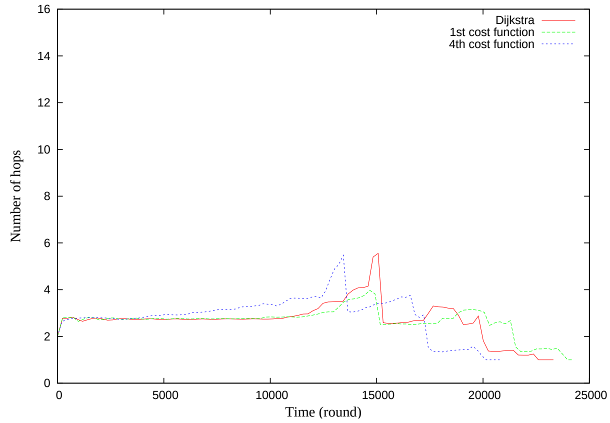

Fig. 13 Number of hops per route as a function of the simulation time.

remaining energy. An energy efficient route is the longest route that avoids using nodes with low energy. This method requires, however, a centralized management in order to be properly implemented, which is not always feasible in wireless sensor networks.

In [22], authors introduce a query-based protocol that searches for the route with nodes having maximal remaining energy. Therefore, a source node sends a route request with an energy threshold, all intermediate nodes with higher energy reply then, to this request. If no route is found, the threshold is decreased and the same procedure is repeated until a route is found. This protocol presents a problem when the threshold is not properly chosen, which consequently generates multiple flooding.

5.2 Power control

Some approaches deal with the problem of increasing the network lifetime by changing each node transmission power. They then, look for the decrease of the consumed energy in data transmission, while assuring network connectivity.

In [11], authors show that reducing the transmission power of nodes will not necessarily minimize the energy consumption, since it will increase the number of hops. They proved that at a certain radius range, the energy consumed for communication is minimal. Nevertheless, the optimal radius for global diffusion differs from the optimal radius for point to point communications. Changing the radius for each communication type makes this solution difficult to implement.

In [3] the paper presents an algorithm to obtain a strongly connected topology by adjusting the transmission power of every node in the network. Being this solution NP-hard, authors have proposed a distributed solution that used the Hitch-Hiking mechanism. This approach enables every node to locally choose its transmission power by using the available information about its 1- and 2-hop neighbors.

In [1] the authors use a closed loop for power control. For this, the destination node embeds in each answer (CTS for RTS or Acks for DATA) the reception power and the minimal threshold required for a good reception. The source receiving the response can then adjust its transmission power. This approach presents a problem when the MAC layer does not receive a response for a CTS or Ack due to an interference: the transmission power of the source will be incremented without any real need.

5.3 Management of nodes activity by state switching

The approaches in this category propose to alternate the activity level of nodes into sleep or awake modes.

In [2], the approach divides the network into disjoint set of sensors such that every set covers all the monitored targets. These sets are activated successively such that at any instant, only one set is active and all sensors of the others sets are in the sleep mode. The network lifetime is then proportional to the number of disjoint sets. In this way, the goal is to maximize this number of sets, which constitutes a NP-hard problem. A heuristic was then proposed as a solution. Nevertheless, although to significantly improve network's lifetime, it requires a centralized management.

In [4] the authors use a localized method to switch the nodes state between active and sleep. For this, the proposed method chooses a dominant set of nodes that are not energy constrained to stay active and all the other nodes are in the sleep state. This set must keep the network connected and the surveillance zone covered. The periodic execution of the algorithm makes the dominant set dynamic and avoids that certain nodes loose their energy early. This methods requires however, a good knowledge of the overall network. In a similar way, authors in [14] propose to switch nodes energy state into sleep, forwarding, or sensing-only. The proposed method relies on a distributed probing approach and on the redundancy resolution of sensors for getting energy optimizations. Contrarily to [4], this method does not require any global network knowledge, but, for some particular cases, it fails to guarantee network connectivity.

In [15] every node detecting that two of its neighbors cannot communicate using an active node, becomes active. The duration of the active state is subject to the remaining energy of the node and the number of nodes it can connect together. This rule permits the node to switch between the active and the sleep state and optimizes the energy consumption. This method requires nodes to change their neighborhood lists in order to correctly activate nodes. Decisions are then based on 2-hop neighborhood knowledge.

In [21] the network is divided into virtual grids using node positions given by a GPS. All nodes in a grid are equivalent in terms of routing and packet forwarding. A node in the active or discovery state becomes inactive when it determines that another node in the same grid can do the routing. The lifetime of the network is optimized by activating one node in each grid. The choice of this node is based on its remaining energy. This method requires a GPS imbedded in every node which is unfeasible in large scale networks.

In [17, 18], authors presents the TITAN management technique. TITAN provides a on-demand topology management for determining nodes that will participate of a forwarding backbone composed of active nodes. Nodes that are not required for forwarding are then allowed to stay in power-saved mode. TITAN, however, is designed to work with on-demand routing protocols, like DSR [12] or AODV [16]. Nodes in the power-save mode receive Route Requests messages (RREQs) but defer their forwarding. Although to avoid any explicit forwarding backbone maintenance, TITAN has no load balancing, being routing load not distributed among nodes.

6 Conclusion and future work

This paper presented an approach to distribute routing load in the network, avoiding thus, the use of energy-compromised hotspots that may cause network disconnections. A modified shortest path algorithm is proposed, where energy efficiency is gotten through *energy-based cost functions* that assigns energy-related weights to links in the network.

Four cost functions were presented and evaluated by simulations. Simulation analysis lead us to the following conclusion: E_{θ_1} gives good results for any communication scenario or simulation stop condition, but does not explicitly take into account the remaining energy of nodes. Adding the remaining energy of the node in the cost function avoids nodes depletion. This was considered in the definition of cost functions E_{θ_2} and $\omega(i)$. These cost functions have strongly load balancing properties, but result in long routes that dramatically increase the energy consumption. In particular, in order to choose the route with the maximum of the minimum E_{θ_2} (or $\omega(i)$) among all routes, this algorithm tends to choose longer routes consuming more energy.

By benefiting from results obtained from the analysis of E_{θ_1}, E_{θ_2}, and $\omega(i)$, we could verify that insuring a minimal of hops per route is crucial for extending the lifetime of the network. The cost function E_{θ_4} was then proposed to optimize both energy consumption and number of hops. The results given by this cost function in Section 4.2, show that despite lightly increasing the number of hops and the consumed energy per route, the network lifetime is highly extended compared to E_{θ_1}. In the 2nd simulation stop condition, E_{θ_4} generates good network lifetime results when all nodes are still alive. In addition, the 1st node to be depleted with E_{θ_4} arrives after the one with E_{θ_1}, which clearly shows the E_{θ_4} lifetime extension capability. Longer routes are however, generated by E_{θ_4} after the 1st node depletion, which impacts the network lifetime and the consumed energy per route.

Future work includes considering more realistic MAC layer models and explores the proposed cost function in well known wireless network routing protocols.

References

1. Agarwal, S., Krishnamurthy, S., Katz, R., Dao, S.: Distributed power control in ad-hoc wireless networks. Proceedings of PIMRC (2001)
2. Cardei, M., Du, D.: Improving wireless sensor network lifetime through power aware organization. ACM Journal of Wireless Networks (2005)
3. Cardei, M., Wu, J., Yang, S.: Topology control in ad hoc wireless networks with hitch-hiking. The First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON04 (2004)
4. Carle, J., Simplot-Ryl, D.: Energy-efficient area monitoring for sensor networks. Computer **37, no.2**, 40–46 (2004)
5. Chang, J., Tassiulas, L.: Energy conserving routing in wireless ad-hoc nertworks. IEEE INFOCOM 2000, Tel Aviv, Israel (2000)
6. Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networs. Wireless networks **Vol.8 Issue 5** (2002)
7. Culler, D., Estrin, D., Srivastava, M.: Overview of sensor networks. IEEE Computer Society pp. 41–49 (2004)
8. Gupta, P., Kumar, P.: Critical power for asymptotic connectivity in wireless networks. In: Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G.Yin, and Q. Zhang (Eds.). Birkhauser, Boston (1998)
9. Hassanein, H., Luo, J.: Reliable energy aware routing in wireless sensor networks. Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems DSSNS (2006)
10. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS). Maui, Hawaii (2000)

11. Ingelrest, F., Simplot-Ryl, D., Stojmenovic, I.: Optimal transmission radius for energy efficient broadcasting protocols in ad hoc networks. IEEE Transactions on Parallel and Distributed Systems (2006)
12. Johnson, D., Maltz, D., Broch, J.: DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, chap. 5, pp. 139–172. Addison-Wesley (2001)
13. Kwon, S., Shroff, N.B.: Energy-efficient interference-based routing for multi-hop wireless networks. IEEE INFOCOM 06, Barcelona, Spain (2006)
14. Merrer, E.L., Gramoli, V., Viana, A.C., Bertier, M., Kermarre, A.M.: Energy aware self-organizing density management in wireless sensor networks. In: ACM MobiShare. Los Angeles, CA (2006)
15. Mirza, D., Owrang, M., Shrugers, C.: Energy-efficient wakeup scheduling for maximizing lifetime of ieee 802.15.4 networks. International Conference on Wireless Internet (WICON'05), Budapest, Hungary (2005)
16. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing. RFC 3561 (2003)
17. Sengul, C., Kravets, R.: Conserving energy with on-demand topology management. In: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (2005)
18. Sengul, C., Kravets, R.: Titan: on-demand topology management in ad hoc networks. Mobile Computing and Communications Review **9**(1), 77–82 (2005)
19. Shah, R., Rabaey, J.: Energy aware routing for low energy ad hoc sensor networks. Proceedings of IEEE Wireless Communications and Networking conference (WCNC) **1**, 17–21 (2002)
20. Shresta, N.: Reception awarness for energy conservation in ad hoc networks. PhD, Macquarie University Sydney, Australia (2006)
21. Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. Proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy (2001)
22. Zhang, B., Mouftah, H.: Energy-aware on-demand routing protocols for wireless ad hoc networks. Wireless Networks **12 Issue 4** (2006)