Avoiding energy-compromised hotspots in resource-limited wireless networks

Joseph Rahmé[1], Aline Carneiro Viana[2], Khaldoun Al Agha[1]

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, the proposed scheme (1) improves significantly the network lifetime for different neighborhood densities degrees, while (2) preserves network connectivity for a longer period of time.

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 and to process information in ways not previously possible [7]. Beside this feature, they include data accuracy, flexibility, cost effectiveness, and ease

[1] LRI, Université Paris-SUD XI – Paris, France. Contact email: {rahme, alagha}@lri.fr
[2] ASAP, INRIA Saclay - Ile de France sud, France

of 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 [4, 2, 6, 15, 11] or
- by keeping nodes in the active state, perform power control [9, 3, 1] or energy-aware routing [10, 13, 8, 5, 14, 16].

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 2 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 presents a new approach that uses cost functions to determine energy efficient routes. By making different energy considerations at a node level, our approach distributes 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 group what is needed to increase network lifetime.

In summary, the contributions of this paper are twofold:

- an intelligent method allowing to (*i*) determine energy efficient paths between nodes in the network, (*ii*) distribute routing load over the network, and (*iii*) avoid energy-compromised hotspots nodes;

- a set of self-configuring cost functions used to determine energy efficient routes and to optimize energy consumption.

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 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, the proposed scheme (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. In Section 2, we present a review of the main related works by providing a general classification of existent approaches. After introducing our system model in Section 3, we present our proposal by introducing the cost functions in Section 4. Performance results are presented in Section 5. Finally, Section 6 concludes this paper and discusses future works.

2 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.

2.1 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 this energy.

In [10], 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 4.2.1), and does not take into account the remaining energy of nodes (described in Section 4.2.2).

Authors in [13, 5], propose a reactive and multi-routing protocol that uses the remaining energy in the node to improve network lifetime. In [13], 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, being a route considered shorter than another if it contains a node with minimal 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 [16], 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 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.

2.2 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 [9], 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. The Hitch-Hiking mechanism is used for that. 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 energy. 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.

2.3 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, one set is active and all the other sets are in the sleep mode. 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 a good knowledge of the overall network. In a similar way, authors in [11] 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 [12] every node detecting that two of his 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.

In [15] 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 embedded in every node which is unfeasible in large scale networks.

3 System Model

We will target a general application scenario where the n sensor nodes are randomly deployed in a zone of interest difficult to 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 begging, 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.

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.

4 Our proposal

In sensor network, the nodes use batteries with limited energy as their 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. To change nodes states between active and sleep seems interesting but presents a major challenge in decentralized systems like WSNs, in other words: *how to determine the duty cycle of nodes and still guarantee connectivity without requiring a global knowledge of the network?*

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-related weight 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-related weight.

4.1 Energy efficient routing

The proposed routing algorithm is in fact, a modified shortest path algorithm, being energy efficiency gotten through *energy-based cost functions*. The values given by these functions represent the weight of the link between a node and his 1-hop neighbors. Thus, once weight of links are computed, routing is performed by following the routes that minimizes the total energy consumed to send a packet from the source to the destination.

The next sections introduce three different cost functions to associate weights to links. 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 distributed among links that present better energy levels. In addition, energy-compromised hot spots are detected and consequently avoided, before packet transmissions.

4.2 Energy-based cost functions

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

- $E_{\theta_1}(i)$: 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.
- $E_{\theta_2}(i)$: considers the remaining energy of node i and its 1- and 2-hop neighbors.
- $\omega(i)$: considers the maximal number of transmissions that node i can perform before node i, or one of its 1- or 2-hop neighbors dies.

4.2.1 Considering consumed energy – 1st cost function:

When a node transmits, all its 1-hop neighbors will 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 cost function E_{θ_1} which avoids the participation of nodes with a lot of neighbors in the routing process. This is due the fact that their energy consumption after a transmission, may represent a significant amount for the network lifetime. thus, This E_{θ_1} is used to assign weights between a node and his 1-hop neighbors: the weight of the link (i, j) between i and any 1-hop neighbor j is equal to E_{θ_1} of node i. E_{θ_1} is thus, defined as:

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

, where

- $N_1(i)$ is the set of 1-hop neighbors of node i;
- $N_2(i)$ is the set of 2-hop neighbors of node i;
- 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 his 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. 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 optimal route given by the modified shortest path algorithm.

Moreover, E_{θ_1} enables the shortest path algorithm to avoid nodes that, if used for routing, will waste a lot of energy in the network. 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 part of the formula: $\sum_{n_1 \in N_1(i)} E_{RX}$ and $\sum_{n_2 \in N_2(i)} E_I$. In a dense area, the weight of these two factors will be high and the routing protocol *will not* route through nodes with high neighbors' density. In this situation, the routing is biased toward using nodes deployed at the borders because they possess a minimal number of neighbors which reduces energy consumption in the network.

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 a remaining energy that is insufficient for performing one packet's transmission can still be selected as next-hop.

To deal with this problem, we use the *remaining energy* of the 1-hop neighbors and E_{θ_1} of the source node to calculate the weight of a link. Thus, our new approach to assign weights for links is the following: the weight of a link (i, j) is equal to $c_f \times \frac{E_{\theta_1}(i)}{E_r(j)}$ where c_f is a weighting factor and $E_r(j)$ is the remaining energy of the 1-hop neighbor j. This approach permits the routing protocol to distinguish between two neighbors of i having different energy remaining: the neighbor with greater energy remaining will form the link with i. The new approach let the routing protocol avoids the nodes with low neighbors density (*i.e. small* E_{θ_1})and remaining energy.

Next section presents the second cost function E_{θ_2}, which explicitly considers the reaming energy in the cost function.

4.2.2 Considering 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:

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

, where

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

As for the 1st cost function, the function E_{θ_2} is used to calculate links' weights between a node and his 1-hop neighbors, thus, for a link (i, j) where j can represent any 1-hop neighbor of *i*, the weight is equal to $E_{\theta_2}(i)$. By considering the emitting node's remaining energy after a transmission, *i.e.,* $(E_r(i) - E_{TX})$, we avoid the case where a node with a minimum remaining energy participates 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 factors of the cost function (2) gives the minimum remaining energy at $1-$ and $2-hop$ neighbors after a transmission.

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 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. As a consequence, it may result in longer routes, consuming by consequence, a high level of energy for the routing of a packet.

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 route algorithm (widest in term of remaining energy) is used.

The optimal route between two nodes is the route where the minimum remaining energy among intermediate nodes is maximal. To find the optimal route, a modified shortest-widest route algorithm is used. 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 route 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}. This imposes long routes to the shortest-widest route algorithm. Longer routes result in more transmissions in the network which increase the energy consumption and consume the remaining energy of nodes.

The next section introduces a third cost function that tries to solve this problem.

4.2.3 Considering 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 two nodes as following:

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

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

$\omega(i)$ uses the ratio $\frac{E_r(i)}{E_{TX}}$ to determine the remaining energy level of a node. Thus, besides indicating the energy of the node, the ratio $\frac{E_r(i)}{E_{TX}}$ 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 his energy.

Combining these three ratios and computing their minimum will give the $\omega(i)$ metric. The final result of this cost function is then, to give the minimum number of transmissions a node can execute before it or a node in its 1- or 2-hop neighborhood looses all their energy.

The weight of a route between two nodes is thus, the minimum weight among intermediate nodes forming this route. As described in Section 4.2.2, to find an energy efficient route between a source and a destination, we use the modified shortest-widest route algorithm with $\omega(i)$ as the cost function. The use of $\omega(i)$ allows an homogeneous load distribution over the network by avoiding nodes with low remaining energy.

Nevertheless, $\omega(i)$ does not consider the energy consumed to route a packet. It only insures an homogeneous energy consumption in order to prevent the depletion of some nodes' batteries before others. Thus, like the previous cost function E_{θ_2}, $\omega(i)$ will give routes with a significant remaining energy but will not take into consideration the energy consumption for routing packets from the source to the destination.

5 Performance evaluation

We have performed some experiments by simulation in order to better evaluate the proposed cost functions. In our experiments, we use a homemade C++ simulator. Our simulator takes into consideration 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 fixed nodes randomly distributed over a square area of 100 meters on a side. The detection range of a node is 20 meters and all nodes possess the same initial power. We consider that at any time only one event can occur in the network. The model of interference is the one described in Section 3.

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. 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. A single event is generated per round of simulation over the network and no wireless routing protocol is implemented. To evaluate the connectivity of the network, we choose arbitrarily a source and a destination and try to find a route between these two nodes. The energy of the nodes in the network is updated after the routing of each packet. The simulation stops when the first node in the network depletes its energy.

Table 2 Energy consumption

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

Figure 1 compares the different cost functions with Dijkstra's algorithm. The figure shows for different network sizes, the number of round a network can support before the depletion of the first node. We vary the number of nodes in the network between $20, 50, 70, 100, 200, 300, 400$, which represents 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 which minimizes the number of hops. 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 in the center of the network. Consequently, this depletes the energy of nodes located in dense regions, violating the homogeneous distribution of the energy consumption and increasing the probability of network partition. Instead, E_{θ_1} increases the lifetime of the network by:

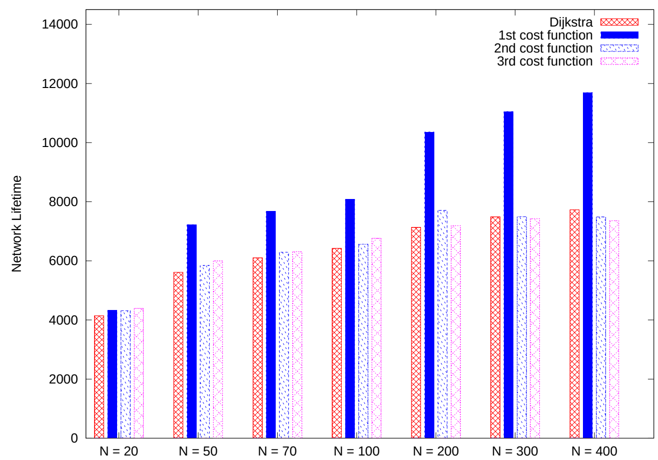

Fig. 1 Network lifetime with different cost functions and node densities.

- avoiding nodes that make the network consumes a lot of energy (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 2), which impacts the energy consumption in the network and decreases its lifetime.

Figures 2 and 3 show the average number of hops and the average consumed energy per route for different nodes densities. The figures show high values $(hop, energyconsumed)$for E_{θ_2} and $\omega(i)$ when the network size increases. This explains why they have a lifetime close to Dijkstra, as shows Figure 1.

More specifically, in Figure 2, for $N = 400$, the number of hops for E_{θ_2} and $\omega(i)$ is very high compared to Dijkstra, which by consequence, explains the high consumed energy showed in Figure 3 and decreases the lifetime of the network. Therefore, it can be concluded that these two cost functions are better adapted for networks with a small number of nodes. We have, however, verified that contrarily to Dijkstra where the first node to die is in the center of the network (consequently, in a dense region), for these two cost functions, the first node is closer to the border

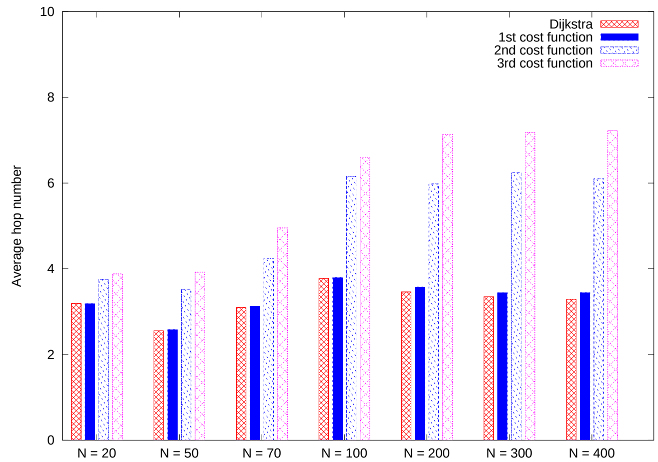

Fig. 2 Average hop number per route for different cost functions and node densities.

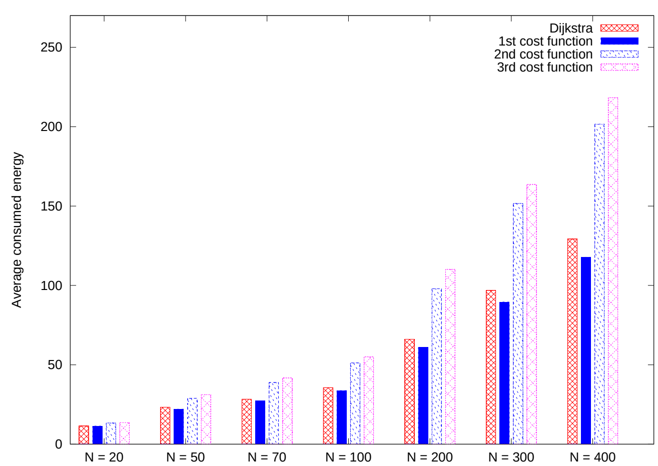

Fig. 3 Average energy consumed per route for different cost functions and node densities.

or on the border of the network. This is an interesting property to be considered, specially in cases where network lifetime is considered as *the maximum operational time of the network before the first disconnection with any sink node happens*.

All these analysis lead us to the following conclusion: E_{θ_1} gives good results but does not explicitly take into account the remaining energy of the nodes. Adding the remaining energy of the node in the cost function avoids nodes depletion. Nevertheless, cost function E_{θ_2} gives poor performance in terms of network lifetime compared to the shortest path algorithm. Moreover, since the cost functions E_{θ_2} and $\omega(i)$ use the shortest-widest algorithm to calculate the route between a source and a destination, it results in long routes that dramatically increase the energy consumption.

The results given by the cost function $\omega(i)$ prove that despite preventing the depletion of nodes, the network lifetime is increased slightly and only for certain node densities compared to the network using Dijkstra's algorithm. $\omega(i)$ takes into consideration:

- The remaining energy of a node by using the ratio of the residual energy and the one needed for a transmission
- The remaining energy of a 1-hop neighbor by using the ratio of the energy of 1-hop neighbors divided by the energy needed for a good reception.
- The remaining energy of a 2-hop neighbor by using the ratio of the energy of 2-hop neighbors divided by the energy needed for decoding a non-intelligible signal

, and chooses the minimum among them. This efficiently prevents node depletion.

Nevertheless, the results in Figure 3 indicate that a limit must be introduced on the amount of energy consumed when choosing a route. We can notice this for $N = 400$, where the energy consumption per route is very high when compared to Dijkstra's algorithm. The function $\omega(i)$ presents the same disadvantage as the 2nd cost function because they both use the shortest-widest algorithm to find a route. In order to choose the route with the maximum of the minimum $\omega(i)$ among all paths, this algorithm tends to choose longer routes consuming more energy.

By benefiting from the results given by our simulations, we can see that insuring a minimal number of hops per route is crucial for extending the lifetime of the network. Therefore, we will intend to calculate the route between the source and the destination using multiple constraints. *A good improvement would be to combine E_{θ_1} with the minimization of the number of hops in the calculation of routes.*

6 Conclusion and future work

This paper presented an approach to distribute the routing loads 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. Three cost functions were presented and evaluated by simulations. Simulation results helped us to better understand the behavior of each proposed function, and by consequence, to find future directions.

Our future work is based on the results of our simulations. The first cost function reduces the energy consumption considerably and increases network lifetime. Therefore, we will use this cost function combined with another constraint on the number of hops that seems very crucial to extend network lifetime. In particular, we will use the cost function E_{θ_1} to calculate the weight of the link of a route, while minimizing at the same time the number of hops of this route. This will preserve at the same time the residual energy without using long route that consumes a lot of energy.

We will also implement a method using three constraints. Thus, in future work, we intend to solve the following problem: to minimize the energy consumption to route a packet from the source to the destination while at the same time (1) to reduce the number of hops between the source and the destination and (2) to maximize the residual energy of intermediate nodes.

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. 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)
9. 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)
10. Kwon, S., Shroff, N.B.: Energy-efficient interference-based routing for multi-hop wireless networks. IEEE INFOCOM 06, Barcelona, Spain (2006)
11. Merrer, E.L., V. Gramoli, A.C.V., Bertier, M., Kermarre, A.M.: Energy aware self-organizing density management in wireless sensor networks. In: ACM MobiShare. Los Angeles, CA (2006)
12. 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)

13. 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)
14. Shresta, N.: Reception awarness for energy conservation in ad hoc networks. PhD, Macquarie University Sydney, Australia (2006)
15. 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)
16. Zhang, B., Mouftah, H.: Energy-aware on-demand routing protocols for wireless ad hoc networks. Wireless Networks **12 Issue 4** (2006)