Dynamic Localized Broadcast Incremental Power Protocol and Lifetime in Wireless Ad Hoc and Sensor Networks

Julien Champ, Vincent Boudet, Anne-Elisabeth Baert

Dynamic Localized Broadcast Incremental Power Protocol and Lifetime in Wireless Ad Hoc and Sensor Networks

Julien Champ, Anne-Elisabeth Baert, and Vincent Boudet

LIRMM - University of Montpellier II - CNRS, 161 rue Ada,
F-34392 Montpellier Cedex 5, France
{champ,baert,boudet}@lirmm.fr

Abstract. This paper deals with the problem of broadcasting messages and lifetime in wireless ad-hoc and sensor networks. The study is based on the best known localized algorithm, namely *LBIP*, which is based on a centralized one, *BIP*, whose principle consists in constructing a broadcast tree rooted on the source node, taking into account the specificities of wireless networks. Even if *LBIP* has excellent performances regarding energy consumption, it selects for each broadcast the same nodes to retransmit the message; if the source of the broadcast, the base station in a sensor network, is always the same, this will lead to deplete quickly the energy of relay nodes. In this paper, we propose *DLBIP*, a new localized broadcast algorithm based on *LBIP*, which dynamically changes the broadcast tree to balance energy consumption on nodes without any additional messages. We show that proposed strategy can significantly increase the number of broadcasts before the network failure. We provide simulations results that clearly demonstrates the lifetime enhancement due to our optimization.

1 Introduction

As a consequence of recent advancements in miniaturization and wireless communications, a new kind of network has come to the fore: *Wireless Sensor Networks* (WSN). In those networks, nodes can gather information from their environment, such as temperature, gas leak, etc. They can also communicate, thanks to their wireless communication device, with other nodes in their transmission range. WSN are also composed of at least one special node, called the base-station, or sink, the purpose of which is either to centralize collected data from the WSN, send queries in the network, or connect the WSN to other networks. WSN recently attracted a lot of attention because of their wide range of applications. They can be used in a many different fields, monitoring tasks either for the military, or the environment, security, health-care, and habitat automation [1].

When broadcasting, the source node needs to send a message to all the nodes in the network. Many applications need to broadcast messages to the whole network: so as to send a query to all the nodes, to broadcast an information, or to do some route discovery ... The broadcasting task occurs more frequently in such networks. Proposed methods need to be designed for wireless sensor networks: sensor nodes are small objects working thanks to a tiny battery and communicating thanks to their wireless communication device.

J. Wozniak et al. (Eds.): WMNC 2009, IFIP AICT 308, pp. 286–296, 2009.

Due to the limited battery power, these networks are power constrained, and as communication ranges are limited, an important set of nodes needs to retransmit the message in order to cover the whole network. The easiest way to broadcast a message to all sensors in the area is called *Blind Flooding* and it works as follows: each node relay once the message, and if there exists a path between the broadcast source and any node in the network, all nodes will receive the message properly. But this method implies a lot of redundant messages.

We can find in the literature various broadcast algorithms used to save energy consumption in the WSN: sometimes nodes can adjust their transmission power in order to save energy and obtain better results, sometimes it is only possible to reduce the number of retransmitting nodes to achieve a full coverage.

Nevertheless, reducing the energy consumption is always realized for the same purpose: to increase the lifetime of the network. It is not sufficient to analyze the energy consumption for one broadcast; it is more interesting to study the lifetime network after several broadcasts. The notion of network lifetime is not clearly defined for ad-hoc and sensor networks in the literature and is clearly application dependent [2]. This point is discussed in section 2.3.

In this paper, we try to guarantee as long as possible the reception of broadcast messages in the network, *i.e.*, over 90% of the sensors have to receive the broadcast messages. We based our work on the *Localized Broadcast Incremental Power Protocol* (*LBIP*) [3], the best known localized algorithm regarding to energy consumption when transmission range adjustment is possible, and we propose the *Dynamic Localized Broadcast Incremental Power Protocol* (*DLBIP*) a new localized broadcast protocol whose principle is to use dynamic broadcast trees to improve lifetime. We provide simulation results demonstrating its efficiency regarding to lifetime.

This paper is organized as follows: we introduce the network and the energy consumption models, and also definitions of network lifetime in Section 2. Section 3 is dedicated to a brief overview of existing broadcasting algorithms. In section 4, we introduce our protocol *DLBIP*. Section 5 presents simulation results comparing *DLBIP* to *LBIP*. Section 6 concludes this article and presents future directions.

2 Preliminaries

2.1 Network Model

We represent a WSN using the widely adopted Unit Disk Graph Model, denoted UDG. An UDG is defined by $G = (V, E)$ where V is the set of nodes (sensors), and E the set of edges representing available communications. Let R be the maximum communication range for all nodes. There is an edge $e = (u, v) \in E$ if and only if the Euclidean distance between u and v, denoted $d(u, v)$ is less or equal R:

$$E = \{(u, v) \in V^2 | d(u, v) \leq R\} \tag{1}$$

Two nodes that can communicate are considered to be neighbors. The hop-distance between nodes i and j is the minimum number of edges to cross to reach j from i. The k-hop neighborhood for node i, is defined as the set of nodes reachable within at most k-hops of node i. Nodes can adjust their transmission range, so as to consume less energy,

while chosen transmission range r is less or equal maximum transmission range R. We assume that nodes are equipped with omnidirectional antennas: if a node i transmits a message with its transmission range set to x, all its neighbors j with $d(i,j) \leq x$ receive it.

2.2 Energy Consumption Model

In networks where nodes are not able to adjust their transmission range, one easy way to measure the energy consumption of a broadcasting algorithm is to count the number of nodes which retransmit the message. In this paper, we consider networks where sensors can adjust their transmission range to reduce their energy consumption. Thus, we use the most commonly used energy consumption model where energy consumption is given according to the chosen transmission range. If a node broadcasts a message with a transmission range equals to r the energy consumption will be:

$$energy(r) = \begin{cases} r^{\alpha} + c & \text{if } r > 0 \\ 0 & \text{else} \end{cases} \tag{2}$$

The most used values for this model are given by Rodoplu and Meng in [4]. They propose to use it with $\alpha = 4$ representing the signal attenuation and $c = 10^8$ an overhead due to signal processing.

2.3 Lifetime Definition

Many broadcasting algorithms proposed in the literature are designed to reduce global energy consumption. But energy consumption reduction is made to extend network lifetime. It seems insufficient to consider only energy consumption of only one broadcast; it is more suitable to analyze the behaviour of the network after more broadcasts.

Lifetime in WSN is still not very well defined whereas choosing a good lifetime criterion is very important when designing new protocols [2]. A good lifetime metric is needed to analyze exactly protocol's behaviour, and to optimize your protocol regarding requirements. There is no definition of lifetime suitable to all kind of applications in Wireless Sensor Networks. The choice of one or another criterion depends clearly on your application requirements.

We can find in the litterature communication algorithms using the Time To First Failure (TTFF), or number of tasks done before one failure to define network lifetime [5]. But, unless if the failure of only one node is a disastrous state regarding our application requirements, TTFF criterion seems to be insufficient. In many applications, the network is not considered as being faulty if one node runs out of its battery energy, or if one node do not receive a broadcast.

It is important to note, that some nodes may be considered as *more important* than other nodes in WSN. For example, a node can run out of its battery energy and partition the network : but, it is not always the First Failure which partition the network into two components. When our application needs that all sensors have to be alive, TTFF is suitable, else we should analyze the number of broadcasts (or time) until less than X% of nodes receive the message, with X depending on application requirements.

3 Related Work

3.1 Broadcasting without Range Adjustment

The easiest way to reduce energy consumption consists in reducing the number of nodes which retransmit the message to achieve the broadcast; many solutions have been proposed to minimize the number of communications.

We can find clustering algorithms constructing connected dominating sets, providing a backbone for communications, so as to reduce the number of nodes used to retransmit messages. Probabilistic protocols have also been proposed such as [6,7].

In the Neighbor Elimination Scheme (*NES*) [1][8], when node *i* receives the message, it monitors if its neighbors have received the message, until a timeout. If all nodes seems to be covered, node *i* does not rebroadcast r the message; else node *i* needs to retransmit the message. This leads to a significant elimination of redundant messages. It is important to note that *NES* can often be used as an additional mechanism over another protocol. In the Multipoint Relay protocol (*MPR*) proposed in [9], 2-hop neighborhood knowledge is needed. The source node selects a subset of its 1-hop neighbors to relay the message in order to cover all its 2-hop neighbors, and sends the message, including its choice in the packet so as to propagate them to its 1-hop neighbors. Finding such a minimal set of nodes is a NP-complete problem, however an interesting greedy heuristic is proposed in [9].

3.2 Broadcasting with Range Adjustment

When nodes are able to adjust their transmission range so as reduce their energy consumption, it is not sufficient to reduce the number of nodes retransmitting messages.

RBOP, LBOP and TR-LBOP: Broadcast Oriented Protocols. *RBOP*, *LBOP* and then *TR-LBOP* [10] have been proposed to improve the efficiency of already existing protocols, taking into account the possibility to adjust transmission range. Their general principle is to use *NES* on a subset of their neighbors defined with respectively the Relative Neighborhood Graph (RNG), the Local Minimum Spanning Tree (LMST), and LMST computed with a target radius. These protocols, specially *TR-LBOP*, offer goods performances regarding to energy consumption, but suffers of latency due to the use of *NES*.

BIP: Broadcast Incremental Power Protocol. In [11], authors proposed a centralized algorithm, named Broadcast Incremental Power protocol (*BIP*), allowing transmission range adjustment, and providing interesting results regarding to energy consumption. The main idea consists in proposing a variant of Prim's minimum spanning tree algorithm, taking into account the wireless multicast advantage: when a node sends a message using its maximal transmission range, all nodes inside its transmission range (its neighbors) receive the message. The principle of *BIP* is the following:

- Initially, the broadcast tree is empty, and the source node is marked.
- All nodes start with their transmission range set to 0.

[1] Also known as Wait & See protocol.

– At each step and until all nodes are covered, *BIP* selects the pair (i, j), with i a marked node and j an unmarked node, and such that the *additional power* needed to reach j is minimum; then i sets its transmission range so as to reach j.

The *additional power* is defined as being the cost for i to reach j, minus the cost of its already selected transmission range.

LBIP: Localized Broadcast Incremental Power Protocol. The Localized *BIP* (*LBIP*) protocol, in [3], is the localized version of *BIP*. Its main principle is the following; the source node applies the *BIP* algorithm in its k-hop neighborhood, and forwards its instructions with the message, as with *MPR* protocol. When node i receives the message, if there is instructions inside the packet i applies the *BIP* algorithm on its k-hop neighborhood using previously received instructions. If there is no instruction in the packet, this means that i neighborhood is already entirely covered, and thus i can drop the packet.

Simulation results show that *LBIP* outperforms other distributed protocols regarding to energy consumption, and is only slightly more energy consumming than centralized *BIP* protocol. Authors shows that the *NES* could be additionally used in order to guarantee a complete coverage. Else, without *NES*, conflicting decisions lead to reach nearly 98% of the nodes in the network (which is in most cases sufficient). They also show that using *LBIP* with $k = 2$ is the best compromise, providing excellent results while required knowledge is not too important.

Lifetime oriented centralized algorithms. In [12], authors consider the problem of broadcasting the maximum number of messages until a node do not receive a broadcast. They propose interesting heuristics but proposed algorithms are centralized and they consider a different source for each broadcast. Another interesting centralized work can be found in [13].

4 Dynamic Localized Broadcast Incremental Power Protocol

4.1 Principle

We consider here a static wireless ad hoc or sensor network, where always the same node (the base station in a WSN) broadcasts a message to all nodes. Our purpose is to maximize the number of broadcast until the reachability goes below than 90%. The reachability is defined as being the number of receiving nodes divided by the total number of nodes.

The general principle of our optimization consists in balancing energy consumption between nodes, by changing relay nodes according to remaining energy. The main part of the protocol is the same used in *LBIP*: each sensor computes its broadcast tree on its k-hop neighborhood, and according to received instructions.

In our protocol, we change the weight metrics to compute dynamic broadcast trees, *i.e.*, which can change at each broadcast. Weights used to compute the broadcast tree are not only computed according to the energy consumption of the communication but also regarding to remaining amount of energy. Let B_i be the remaining amount of energy on

node i. If we denoted $C(i \to j)$ the weight for a communication from i to j, our new weight is computed as follows:

$$C'(i \to j) = \frac{C(i \to j)}{B_i} \tag{3}$$

Using this new weight to compute the broadcast tree, nodes with lower remaining energy are less selected to retransmit the broadcast, or are asked to communicate on a shorter distance. This leads to a better balance of energy consumption between nodes. This solution extend network lifetime : the broadcast tree, contrary to *LBIP*, is now not unique, it changes according to remaining energy. As our optimization is based on a dynamic broadcast tree, and is based on *LBIP*, we name it *DLBIP* for Dynamic *LBIP*.

Fig. 1 illustrates our dynamic algorithm. On the left we can see the unique broadcast tree computed by *LBIP* ; this is also at least the first broadcast tree computed by *DLBIP* (if initially $B_i = B_j$ for every node i and j). On the right we can see one of the other broadcast trees computed by *DLBIP*.

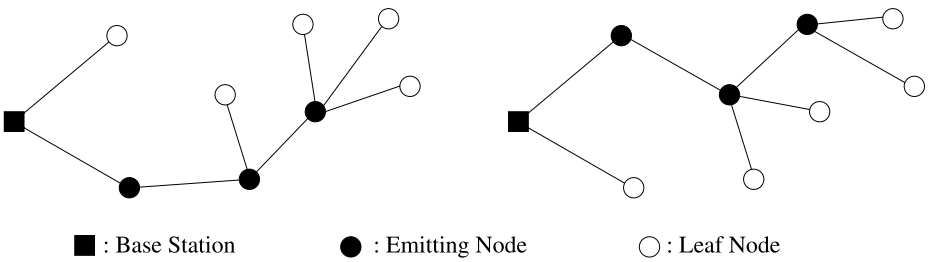

Fig. 1. A small network

In the following example, we compare the weightings used to compute broadcast trees, with *LBIP* and *DLBIP*. We consider three nodes i, j and k such that $d(i,j) < d(i,k)$. Node i wants to communicate the message to j and k.

With *LBIP* algorithm, the choice is made by the following inequality:

$$C(i \to k) < C(i \to j) + C(j \to k) \tag{4}$$

If previous inequality is true, i sets its transmission range to communicate with both nodes ; else i sets its transmission range to communicate with j and asks it to retransmit the message to k.

As described before, with *DLBIP* remaining energy on nodes i and j change previous inequality:

$$\begin{aligned} & C(i \to k)/B_i < C(i \to j)/B_i + C(j \to k)/B_j \\ \Leftrightarrow \quad & C(i \to k) < C(i \to j) + C(j \to k) \times B_i/B_j \end{aligned} \tag{5}$$

The choice is related both to communication costs and remaining energy. If j remaining energy is still high while i remaining energy has decreased, j is more easily chosen as a

relay node. On the other hand, if $B_i > B_j$ node i will probably send the message to both nodes.

Remark: In this paper, we have decided to consider only remaining energy of sending nodes to compute the dynamic broadcast trees. But other ways to weight edges can be used to balance energy consumption in the network: for example, critical nodes whose failure lead to partition the network can be detected. We can use this to limit their energy consumption, so as to partition the network as late as possible.

4.2 Energy Updates

However, so as to use *DLBIP*, each sensor needs to know or assess remaining energy of each its k-hop neighbors. We propose here a method which can be divided in two parts:

Approximate calculations
Each sensor tries to estimate remaining energy on nodes in its neighborhood according to its knowledge:

- Each time a sensor computes a broadcast tree, covering its neighbors, it can assess energy consumption of its neighbors. As the broadcast tree is computed according to received instructions, and as sensors which will receive messages will obey to included instructions, estimations are close to real energy consumption.
- For a given broadcast, when sensor i receives a message without instruction for i, the message is not retransmitted. However, when a node receives the packet for the first time, it can read instructions for all its k-hop neighbors, providing a good way to update the estimation of remaining energy of its neighbors.

Accurate updates: When sensor i computes the broadcast tree needed to cover its k-hop neighbors, and includes instructions in the message, it can also include a field with its remaining energy B_i. This message is sent to its neighbor, which can accurately update their estimation of remaining energy on node i. These updates can be done until the message reaches node outside i neighborhood, then the field can be removed.

Remark: to avoid an excessive packet size increase, it is not needed to include remaining energy for each broadcast, this can be done once in a while.

5 Performances Evaluation

5.1 Simulation Parameters

In order to evaluate performances of *DLBIP*, we present simulation results in this section using WSNET simulator [14]. We compared our protocol to *LBIP*, because our protocol is based on it and as it outperforms other localized protocols regarding to energy consumption. We have chosen nearly the same parameters used for the evaluation of *LBIP* in [3]. As said in Section 2 we used the Unit Disk Graph to model available communications. We consider a static network, composed of 500 nodes randomly deployed using an uniform distribution inside a square area. The size of the area is computed according to chosen network density. The maximum transmission range is fixed to 250.

All nodes have initially the same amount of energy, except the base station which can transmit as many messages as needed. The energy consumption model is the one presented in section 2 equation 2 with $\alpha = 4$ and $c = 10^8$. As in [3], *LBIP* and *DLBIP* have been implemented with an ideal MAC layer: two nodes can transmit a message at the same moment, without collisions. *LBIP* and *DLBIP* compute their broadcast tree within their 2-hop neighborhood as simulations in [3] shows that $k = 2$ seems to be the best compromise.

To resolve conflicting decisions in *LBIP* protocol, the authors proposed in [3], to use the Neighbor Elimination Scheme to reach a total coverage. Without the additional *NES* over *LBIP*, the coverage is still enough high for most applications. As our algorithm is based on *LBIP*, the same conflicting decisions may appear, and can also be solved using the *NES*. Thus, so as to measure precisely the impact of our proposition we have decided to analyze simulation results without the use of the *NES* in *LBIP* and in *DLBIP*.

We have decided to consider that the network is faulty when less than 90% of nodes receive the broadcast, so we study the number of broadcasting tasks that can be done until reachability goes below 90%. Simulation results are similar, whatever the initial energy on sensor is; that's why we do not compare results according to the initial amount of energy.

5.2 Simulation Results

In Fig. 2 we give for *LBIP* and *DLBIP* the percentage of receiving nodes, regarding to the number of broadcasts done. With *LBIP* protocol, the chosen broadcast tree is always the same, thus always the same node are selected as relay nodes. In density 20 networks, this obviously lead to a achieve less than 90% of covered nodes after 210 broadcasting tasks. Contrary to *LBIP*, *DLBIP* uses a dynamic broadcast tree, relay nodes and transmission ranges are selected according to both communication costs and remaining energy: this leads to a better lifetime of the network. Indeed, 600 broadcast are achieved before less than 90% nodes are covered. This means an increase of nearly 185% until 10% of nodes are not covered in density 20 networks.

The percentage of transmitting nodes for density 20 networks, given in Fig. 3, is obviously initially the same for *LBIP* and *DLBIP*, but as the number of realized broadcast

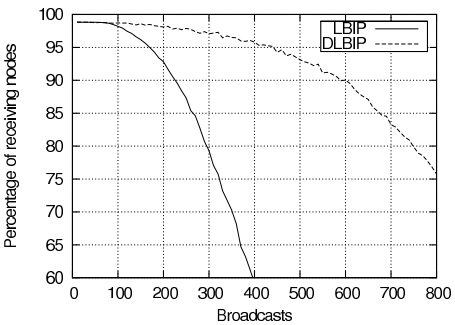

Fig. 2. Reachability in density 20 networks

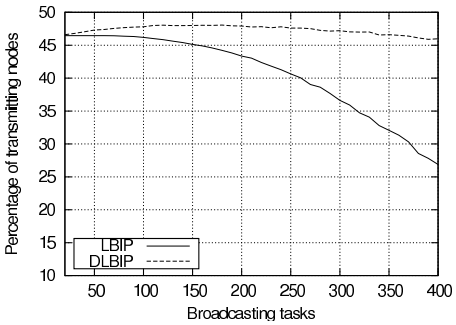

Fig. 3. Percentage of transmitting nodes

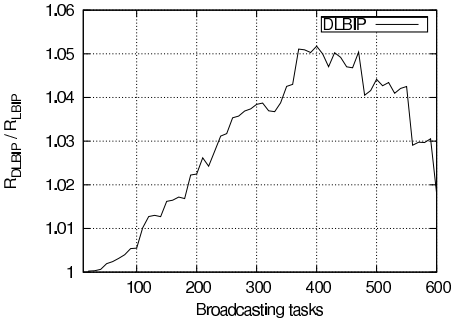

Fig. 4. Ratio R_{DLBIP}/R_{LBIP}

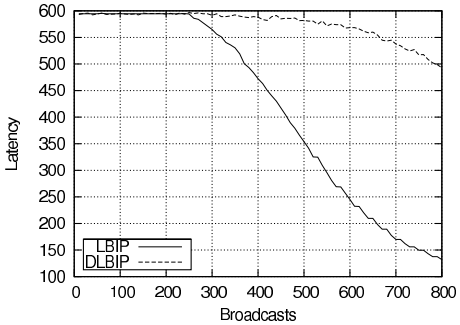

Fig. 5. Broadcasts latency

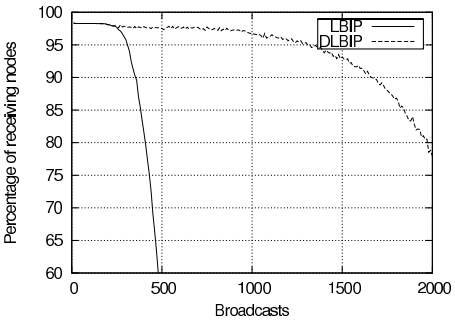

Fig. 6. Reachability in density 40 networks

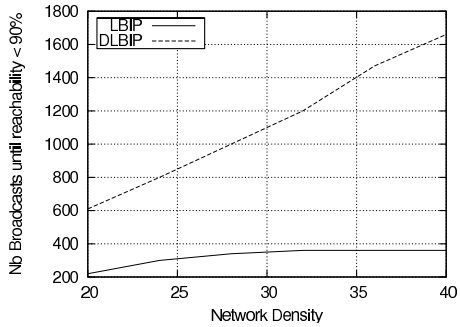

Fig. 7. Lifetime regarding to density

increases, *DLBIP* use a few more emitters. This can be explained: if node A was used to cover some nodes, when its remaining energy decreases too much, there is two possible alternatives. Either another node increases its transmission range to cover its neighbors, or two other nodes communicate the message, leading to an higher number of emitters.

Let R_{LBIP} be the average energy consumed by a node, *i.e.* the ratio obtained by dividing the energy consumption for a broadcast using *LBIP* protocol with the *number of nodes*. As the broadcast tree computed thanks to *LBIP* is static, we compute this ratio only for the first broadcast. Let R_{DLBIP} be the average energy consumed by a node, *i.e.* the ratio obtained by dividing the energy consumption for a broadcast using *DLBIP* protocol with the *number of covered nodes*. This time, the broadcast tree is dynamic, thus we analyze the evolution of ratio R_{DLBIP}/R_{LBIP} regarding to realized broadcasts in Fig. 4. We can see that the average energy consumption by node, needed by *DLBIP* is slightly higher (less than 10%). This increase is due to the weighting of communication costs with remaining energy, and is needed to balance energy consumption so as to improve lifetime.

Fig 5 give us the latency in arbitrary units of the broadcasting tasks regarding to the number of broadcast realized. The latency is computed as being the elapsed time for all sensors to receive a broadcast. There is no significant increase of the latency after

several broadcast when using *DLBIP* compared to *LBIP*. The latency is only linked to the number of remaining alive nodes and to the broadcast tree depth which is nearly the same for both protocols.

We provide in Fig. 6 reachability, *i.e.* the percentage of receiving nodes, in a network with a higher density than in Fig. 2; this is a network with density equals to 40. We can note that our algorithm is scalable: indeed, in density 40 networks *DLBIP* perform more than 350% more broadcast than *LBIP* until reachability goes below 90%. This is confirmed by Fig 7, which gives the average number of broadcasts done before the reachability goes below 90%. We can note that when the density is between 30 and 40, *LBIP* achieve less than 400 broadcasts, while *DLBIP* goes on increasing the number of broadcasting tasks. The higher the density is, the more choices *DLBIP* has to balance energy consumption.

6 Conclusion and Future Work

In this paper, we deal with optimizing the number of broadcasts until 10% of sensors failed receiving the message. We argue that only measuring the energy consumption of one broadcast is not sufficient to make a good performance analysis of the lifetime of a sensor network. Thus we propose a new protocol, *DLBIP*, which is a new dynamic broadcast algorithm based on the *LBIP* algorithm.

Actually, in the case of a static wireless ad hoc or sensor network, where always the same node wants to broadcast a message to all nodes, computing the energy consumption of one broadcast is not sufficient. Our method consists in taking into account the remaining energy of nodes in the network, to obtain in time a dynamic broadcasting tree which optimizes the lifetime of network. The main property of *DLBIP* is that it does not need additional messages to work. Simulations show the efficiency of the protocol in term of lifetime comparatively to *LBIP*, even if *DLBIP* is slightly more energy consuming.

Further research should address other ways to balance energy consumption, such as detecting critical nodes, or trying other weightings to compute *DLBIP* broadcast trees, to see the most efficient trade-off between energy consumption and load balance. Other simulations made with different broadcast sources for each broadcasting task should also be done to analyze the impact of such a protocol when the source is different for each broadcast. As for *LBIP* protocol, it should be interesting to analyze our protocol using an energy consumption model which considers reception costs and also with more realistic physical layer.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Champ, J., Baert, A.E., Saad, C.: Lifetime in wireless sensor networks. In: CISIS: Conference on Complex, Intelligent and Software Intensive Systems, Fukuoka, Japan (2009)
3. Ingelrest, F., Simplot-Ryl, D.: Localized broadcast incremental power protocol for wireless ad hoc networks. Wirel. Netw. 14(3), 309–319 (2008)

4. Rodoplu, V., Meng, T.: Minimum energy mobile wireless networks. In: Proc. of IEEE International Conference on Communications, ICC (1998)
5. Chang, J.H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: INFOCOM, pp. 22–31 (2000)
6. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. Wirel. Netw. 8(2/3), 153–167 (2002)
7. Cartigny, J., Simplot, D.: Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In: Telecommunication Systems, pp. 189–204 (2003)
8. Peng, W., Lu, X.C.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, Piscataway, NJ, USA, pp. 129–130. IEEE Press, Los Alamitos (2000)
9. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In: HICSS 2002: the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), vol. 9 (2002)
10. Ingelrest, F., Simplot-Ryl, D., Stojmenovic, I.: Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. IEEE Trans. Parallel Distrib. Syst. 17(6), 536–547 (2006)
11. Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: Energy-efficient broadcast and multicast trees in wireless networks. Mob. Netw. Appl. 7(6), 481–492 (2002)
12. Park, J., Sahni, S.: Maximum lifetime broadcasting in wireless networks. IEEE Trans. Comput. 54(9), 1081–1090 (2005)
13. Elkin, M., Lando, Y., Nutov, Z., Segal, M., Shpungin, H.: Novel algorithms for the network lifetime problem in wireless settings. In: Coudert, D., Simplot-Ryl, D., Stojmenovic, I. (eds.) ADHOC-NOW 2008. LNCS, vol. 5198, pp. 425–438. Springer, Heidelberg (2008)
14. Fraboulet, A., Chelius, G., Fleury, E.: Worldsens: development and prototyping tools for application specific wireless sensors networks. In: IPSN 2007: Proceedings of the 6th international conference on Information processing in sensor networks, pp. 176–185. ACM, New York (2007)