# Distributed Search of Various Backbones in Wireless Sensor Networks

V. Boudet
LIRMM
University
Montpellier 2
Montpellier,
France
boudet@lirmm.fr

S. Durand
LIRMM
University
Montpellier 3
Montpellier,
France
sdurand@lirmm.fr

L. Gönczy
MIT
BME
Budapest, Hungary
gonczy@mit.bme.
hu

J. Mathieu
IUT Rodez
University
Toulouse 1
Rodez, France
jerome.mathieu@i
ut-rodez.fr

J. Palaysi
LIRMM
University
Montpellier 2
Montpellier,
France
palaysi@lirmm.fr

*Abstract*— **In this paper, we are interested in enhancing lifetime of wireless sensor networks trying to collect data from all the nodes to a "sink"-node for non-safety critical applications. Connected Dominating Sets are used as a basis for routing messages to the sink. We present a simple distributed algorithm, which computes several CDS trying to distribute the consumption of energy over all the nodes of the network. The simulations show a significant improvement of the lifetime of the network.**

***Keywords - sensor network; lifetime; distributed algorithm***

## I. INTRODUCTION

In this paper, we investigate the communication efficiency in wireless sensor networks (WSN), which are consisted of sensor with a limited energy resource (batteries). Each sensor is able to communicate with a few other sensors in its neighborhood within its communication range, which we assume to be roughly the same for all sensors (however, this assumption is not restricting the application of our algorithm). The sensors regularly perform measurements and measured data is collected to a single special node of the network called the sink (this operation is called gathering). Our goal is to maximize the number of gatherings that can be done by the network. For each gathering, the set of nodes used for transmissions have to be connected and to dominate the graph associated to the network (backbone). We also assume that there is no central entity to compute optimal routing for communication, therefore we are interested in developing distributed algorithms for this purpose.

In Sections II and III, we present related work, the model and the main assumptions we made in order to get realistic results. Then a distributed algorithm is proposed in Section IV to compute backbones, which distributes the use of sensors for transmissions to maximize network lifetime. Section V shows experimental results achieved by simulations about the lifetime of the network on grid and random topologies. We conclude and give tracks for future works in Section VII.

## II. RELATED WORK

Because of the critical importance of energy saving in WSN, literacy about this subject is extensive. In order to increase the lifetime of a WNS, one typical way is to use Connected Dominating Sets (CDS) also called *backbones* to route the messages where only the nodes belonging to the backbone use energy to forward messages. The goal is then to minimize the number of nodes of the backbones [1]. Because of the need of robustness and scalability of the solution, most algorithms are operating in a distributed manner with local election of the nodes belonging to the backbone [2]. Since this problem is NP-hard, some authors work to find guarantees on the approximation ratio [3].

Unfortunately, with this strategy, the nodes belonging to the backbone will consume more energy. Thus after a certain period of time, the network will be disconnected while the other node may still have a lot of energy. In order to increase the lifetime one can compute several backbones, trying to find a set of CDS such that the maximum number of CDS a node belongs to is minimized. Such a distributed algorithm is proposed in [4]. Nevertheless, this model does not take into account the real consumption of energy of the nodes, which depends (among others) on the number of received messages i.e. on the degree of the node. In [5], the authors dynamically construct backbones taking into account the remaining energy of each node. In our case, all data have to be gathered to a fixed sink. We thus only have to compute a directed in-tree rooted at the sink, and the sink may initiate this computation (the algorithm is not localized but only distributed). This specification allows us to need only a small number of messages to compute a backbone.

## III. THE MODEL

A WSN can be modeled by a graph $G=(V, E)$ where $V$ is the set of sensors and an edge $e=(v, w) \in E$ if $v$ and $w$ can communicate. We suppose that if $v$ can communicate with $w$, the contrary is also (G is not directed). Results on WSN are highly dependant on several parameters of the network (density, model of energy consumption, measurement frequency, etc.). We thus have to make several assumptions in order to specify the global framework of our work.

We suppose that the frequency of the measurements is small enough so that there is enough time to collect data from all sensors to the sink without new measurements being sent. If we compute a routing to collect the data, we then can *aggregate* them at each node. So to do a gathering, we just have to find a directed in-tree rooted at the sink. We also assume that the size of the data is small enough so that even

 ISBN: 978-1-61208-144-1 

after several aggregations the size of the messages sent will be less than one packet. Thus for each gathering, the emission cost will be the same for all sensor. This is our unit to measure the energy consumption. Note that this assumption is just done to fix the conditions of the experiments: our algorithm still works if the size of the messages (and thus the energy needed to send them) is not constant and the sensors know their remaining energy. The cost for a reception cost will not be supposed to be equal to zero. Although this cost is not often taken into account especially in theoretical models, this seems to be a reasonable assumption regarding for example [6] or [5]. Both emission and reception costs depend on the sensors type and on the transmission range but they usually have the same order of magnitude.

The energy needed for measurement depends on the kind of measurement performed and the device used to do so. Taking into account this energy in the model would make it highly dependent on the application. In order to avoid this dependency, we will suppose that auxiliary batteries provide the energy needed to make the measurements and thus the cost for measurement is null. One can remark that this cost would not be difficult to take into account in our model and since for each gathering, each node will make exactly one emission, we just need to increase the cost of emission of this value for the messages containing data.

## IV. DISTRIBUTED SEARCH OF VARIOUS CDSS (DSVB)

### A. Main Algorithm

The principle of our algorithm is that each node will choose a father in the backbone the first node from which he received an "invitation" message. Since all the nodes but the sink send their invitation after receiving one and waiting a certain time, this ensures that we built a directed in-tree rooted at the sink. More formally, for each search of a backbone $b$, the algorithm works as follows:

- The sink $s$ sends invitation <INV> with its id and $b$'s id (broadcast to all of its neighbors)
- For all other nodes $v$ do
  - Father ← sender's id in the first <INV> received
  - Send acceptation <ACC> (with $v$'s id, Father's id and $b$'s id)
  - Chose a delay $w$
  - Wait $w$
  - Send <INV> (with $v$'s id and $b$'s id)
  - If (number of received <ACC> = number of neighbors) and ($v$ is a father in none of the received <ACC>) then $v \notin b$
  - If, in a received <ACC>, (Father's id = $v$'s id) then $v \in b$

For each node, computing a backbone with this algorithm needs only to send two messages and so a number of receptions equal to twice the degree of the node.

### B. Computing the Delay

Fine-tuning of the algorithm is done by the computation of the delay each sensor has to wait before sending an invitation. The influence of this delay obviously depends on the time needed for one node to run the algorithm (if the algorithm needs 100 µs to be run, adding a delay of 2 µs will not have much influence). Let $t$ be an upper bound of the time a node needs to run the algorithm (receive and process a message, compute the delay and send a message). Our unit to measure the energy of a node will be the amount $e$ of energy needed to send or receive a message. Let $e \cdot E_i$ be the initial energy of the node (thus $E_i$ is the number of messages a node can send), $e \cdot E_r$ its remaining energy after a certain duration of use. This remaining energy may by known by the sensor or evaluated considering the numbers of emissions and receptions already done (in this second case, the battery model is supposed to be linear). Let $n_c$ be the number of CDS already used and $n_b$ the number of CDS a node already belongs to. A "penalty" is computed for each node. It has to increase when:

- The proportion of CDS a node belongs to increases in order to discriminate the nodes even at the beginning of the process. This parameter is especially important for regular graphs.
- The remaining energy of a node decreases.

The penalty is thus computed using formula (1)

$$p = t \times \left( f\left( n_c / n_b ; E_r \right) \right) \quad (1)$$

where the function $f$ is an increasing function of its parameters. In order to differentiate nodes that would have the same penalty, each delay is randomly chosen in a range $[0, exp*p]$ where $exp$ is an expansion factor that increases the differences of penalty ensuring that the delay is not too long. Note that if the delay is constant, then our algorithm is formally identical to a Breadth First Search (BFS).

## V. MAIN RESULTS

In order to validate our approach and be as near as possible of the functioning of a real network, we use WSNET simulator. Despite NS-2 is more often use in the literacy, WSNET is reported to have a more realistic model for transmission [8]. We simulate the effective communications between the nodes with an autonomous functioning of the nodes. The sink is supposed to have an unlimited energy. In our simulations, $t \approx 2$ µs. Those values are chosen considering the devices Micaz of MEMSIC on which we plain to make experiments on in further works.

For the delay, $f$ is chosen according to equation (2).

$$f\left( n_c / n_b ; E_r \right) = \left( \frac{\max(n_c ; 1)}{\max(n_b ; 1)} \times \frac{E_i}{1 + E_r} \right)^k . \quad (2)$$

The value of $f(n_c/n_b;E_r)$ is lower or equal to 1, but usually very smaller than 1. In order to know the highest penalty when a backbone is computed, each node has to transmit the maximal current penalty it knows (from its penalty and its children) when returning a data. The highest penalty $p_{max}$ is

 ISBN: 978-1-61208-144-1 

then included in the invitation message. The expansion factor is given by $exp = c / p_{max}$ where $c$ is a coefficient depending on the size of the network. The latency to built a backbone is at most $|V|*exp$. In our experiments, $c$ is set so that the latency is lower or equal to 10 ms.

### A. Networks

Two kinds of networks are used to do the simulations.

#### 1) Grids

In many potential applications, sensors are not randomly spread and the network has a "grid-like" shape (deployment in fields, cities, building, containers on a boat). Furthermore, grids have interesting properties for our studies since they both have low density (which make the computation of disjoint CDS difficult) and relatively high connectivity (which helps to avoid degenerated cases that may occur because of the presence of isthmus or lowly connected parts).

We made simulations on two kinds of grids. The first one $G_R(p, q)$ is the usual $p \times q$ grid. In the second one $G_{R\sqrt{2}}(p, q)$, each node cannot only communicate with its 4 neighbors but is also connected on the diagonal and thus has 8 neighbors. The sink is always the center of the grid. The main part of the simulations are made using $p = q = 11$.

#### 2) Random networks

The networks are unit disk graphs. For those network, $|V|$=100 and the density (average number of neighbors) is 10. We choose the 100 first connected networks generated.

### B. Results

#### 1) Setting the parameters

In order to see how the different parts of the delay influence the construction of the CDSs, we try several combinations of the parameters. The best value for $k$ (the exponent in formula 2) is 6.

In TABLE I. we present the number of gatherings that can be collected to the sink using different parameters for the delay computation and for different graphs. In the first line, the penalty is randomly chosen between 0 and 1. In the second line, $E_r$ is set to $E_i$-1. In the third line, $n_c$ is set to $n_b$ and in the last line, both parameters are taken into account. The values in the 2 first columns are means for 16 runs on the same graph. The values in the last column are means for 16 random graphs with various density and connectivity 2 or 3. A new CDS is computed every 116 gatherings.

TABLE I. NUMBER OF GATHERINGS COLLECTED AT THE SINK ($EI$=8000)

| Penalty | Network | | |
|---|---|---|---|
| | $G_R(11, 11)$ | $G_{R\sqrt{2}}(11, 11)$ | *Random graphs* |
| *Random* | 1578 | 911 | 593 |
| *Frequency* | 1849 | 1358 | 949 |
| *Energy* | 2059 | 1615 | 1182 |
| *Both* | 2215 | 1735 | 1159 |

These results show the interest in using both parameters to compute the delay on regular graphs. For random graphs, taking into account the frequency does not seems to improve the lifetime, but the loss is negligible so both parameters will be used for the next simulations on random networks.

#### 2) Frequency of CDS computation

Figure 1. presents how the initial energy and the frequency of the CDS computations influence the ratio of gatherings achieved (number of gatherings / intial energy). We give average values obtained considering 16 runs on $G_{R\sqrt{2}}(11, 11)$.

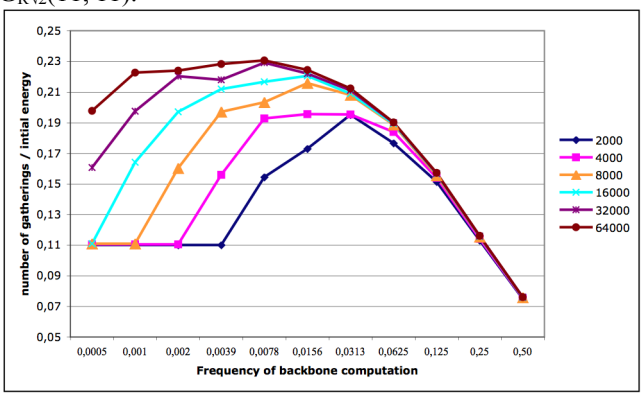


Figure 1. Ratio of gatherings collected at the sink regarding the frequency of backbone computation for different initial energy

As expected, when the number of gatherings per backbone is higher than the number of gathering collected (low frequency), the ratio is constant and corresponds to the use of a single backbone. On the contrary, computing many backbones has a cost, which becomes excessive when the frequency is too high.

#### 3) Efficiency of DSVB

In order to know how our algorithm performs, we compute lower and upper bounds for the number of gathering that may be achieved.

The number of gathering made using a single CDS (for example, found by a BFS) $C$ gives a lower bound. In this case, for a node $v$ in the CDS, the energy used for each gathering is $(d(v)+1).e$ (where $d(v)$ is the degree of $v$ in $G$): this node receives messages from all its neighbors (cost $d(v).e$) and transmits its aggregated data to its father (cost $e$). Using this strategy, the maximum number of gatherings allowed is $\min_{v \in C}(E_v(v)/(d(v)+1))$. We thus have to find a CDS $C$ such that $\max_{v \in C}(d(v))$ is minimum.

To compute an upper bound for the number of gatherings we extend the idea proposed in [7] to our model. For each vertex-cut $S$ that disconnect $G$ (for convenience, we will suppose that $S$ does not contain any leaf of $G$), we need, for each gathering, to use at least one of its vertices. The energy used by a node $v$ for a gathering is $(d(v)+1).e$ if $v$ is in a CDS and $e$ else. So if $x_i$ is the number of CDSs a node $i$ belong to, for each node $i$ we must have $(d(i)+1)x_i + \Sigma_{j \cdot i} x_j \leq E_i(i)$. The number $z$ of gathering that can be supported by the set $S$ is then: $z = \Sigma_i x_i$. To obtain an upper bound, we can solve the relaxation of this linear program. If every constraint is tight $(x_i = (E_i(i) - z)/d(i)$ it leads to:

$$z = \sum_{i \in S} \frac{E_i}{d(i)\left(1 + \sum_{j \in S} \frac{1}{d(j)}\right)}. \qquad (3)$$

Since it is possible to find a solution of the dual having the same value (set $y_l = 1/(d(i)(1+\Sigma_j(1/d(j)))$ for all $l$ in $S$), this solution is optimal.

Finding the set $S$ of vertices disconnecting $G$ such that (3) is minimized gives an upper bound for the maximum number of gatherings. For example, for a network $G_R(p, q)$, a set $S$ achieving the minimum number of gatherings is composed of the two neighbors of a corner.

In order to evaluate the efficiency of our algorithm, TABLE II. shows the number of gatherings collected using DSVB compared to a lower and an upper bound (the conditions are the same as for TABLE I.

TABLE II. EFFICIENCY OF DSVB

| Network | Number of gatherings achieved | | |
|---|---|---|---|
| | *Lower Bound* | *DSVB* | *Upper bound* |
| $G_R(11, 11)$ | 1600 | 2215 | 2666 |
| $G_{R\text{-}2}(11, 11)$ | 888 | 1735 | 2754 |
| Random graphs | 554 | 1182 | 1763 |

Although the actual result using a single CDS would be the lower bound, the upper bound may be overestimated, which suggests a better performance of our algorithm.

*4) "real" lifetime*

Lifetime of WSN is not a well-defined notion [8]. Especially in the case of networks where the measures made by the sensors are redundant, one may accept a reasonable ratio of loss of messages (i.e., lifetime is not equal with the guaranteed message transfer). This logically influences the lifetime of the network. We have seen that in usual grid networks, the main (theoretical) problem occurs for the corners since they are only connected to the rest of the network by 2 nodes of degree 3. Nevertheless, if we accept a loss rate of 4% those nodes are not any more limiting as soon as the grid has more than 100 nodes. Figure 2. shows how the number of transmitted measurements (i.e. connected nodes) decreases regarding the number of gathering for two representatives runs on $G_R(11, 11)$.

We fix a threshold of 90% such that a new backbone is computed either if 100 gatherings are done or if less than 90% of data are collected. For 16 runs, the first failure occurs in mean at the gathering number 2215, whereas we achieve 2341 gathering collecting more than 90% of data. For random networks, the lifetime increases of 10% with a threshold of 95% and 13% with a threshold of 90%.

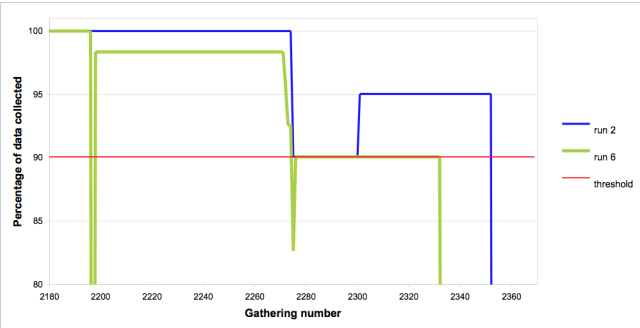


Figure 2. Percentage of data effectively collected regarding the number of gatherings on $G_R(11, 11)$

## VI. CONCLUSION

In this paper, we presented a distributed algorithm to collect data in a Wireless Sensor Network. Easy to implement, it computes several Connected Dominating Sets sharing out the use of the sensors for the transmissions. Simulations have shown a significant improve of the network's lifetime. One of the next steps is to validate our approach on real sensors. In several applications, measures are redundant and some algorithms exist to optimize the energy used for measuring. In a future work, we will try to combine our technique with an efficient k-coverage of the network. We aim to save energy globally for both measuring and communicating.

## ACKNOWLEDGMENT

This work was supported in part by the Hungarian–French Intergovernmental S&T Cooperation Program under the "Dependable Services in Sensor Networks" (F-6/2008) project.

## REFERENCES

[1] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, pp. 374–387, Apr. 1998.

[2] C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominated Sets with Multipoint Relays," *Adhoc & Sensor Wireless Networks*, vol. 1, n°. 1, pp. 27-39.

[3] K. Islam, S. G. Akl, and H. Meijer, "A Constant Factor Localized Algorithm for Computing Connected Dominating Sets in Wireless Sensor Networks," in *International Conference on Parallel and Distributed Systems*, pp. 559-566, 2008.

[4] K. Islam, S. Akl, and H. Meijer, "Distributed Generation of a Family of Connected Dominating Sets in Wireless Sensor Networks," in *Distributed Computing in Sensor Systems*, vol. 5516, B. Krishnamachari, S. Suri, W. Heinzelman, and U. Mitra, Ed. Springer Berlin / Heidelberg, 2009, pp. 343-355.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, pp. 481–494, Sep. 2002.

[6] T. Val, A. Van Den Bossche, and E. Campo, "Technologie ZigBee / 802.15.4 - Protocoles, topologies et domaines d'application," *Techniques de l'ingénieur*, vol. 7508, May. 2008.

[7] E. Ben Hamida, G. Chelius, and J. M. Gorce, "Impact of the Physical Layer Modeling on the Accuracy and Scalability of Wireless Network Simulation," *SIMULATION*, vol. 85, n°. 9, pp. 574 -588, 2009.

[8] J. Champ, C. Saad, and A. Baert, "Lifetime in Wireless Sensor Networks," presented at the CISIS : International Conference on Complex, Intelligent, and Software Intensive Systems, Fukuoka, Japan, 2009.

 ISBN: 978-1-61208-144-1 