Using Battery Level as Metric for Graph Planarization

Jovan Radak, Nathalie Mitton, David Simplot-Ryl

▶ **To cite this version:**

Jovan Radak, Nathalie Mitton, David Simplot-Ryl. Using Battery Level as Metric for Graph Planarization. 10th International Conference on Ad Hoc Networks and Wireless (AdHocNow), Jul 2011, Padderborn, Germany. pp.58-71. inria-00599117

HAL Id: inria-00599117
https://inria.hal.science/inria-00599117

Submitted on 17 Aug 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Battery Level as Metric for Graph Planarization

Jovan Radak, Nathalie Mitton, and David Simplot-Ryl

INRIA Lille - Nord Europe, Univ Lille Nord de France,
USTL, CNRS UMR 8022, LIFL, France
{firstname.lastname}@inria.fr,

Abstract. Topology control in wireless sensor networks is an important issue for scalability and energy efficiency. It is often based on graph reduction performed through the use of Gabriel Graph or Relative Neighborhood Graph. This graph reduction is usually based on geometric values. In this paper we tackle the problem of possible connectivity loss in the reduced graph by applying a battery level based reduction graph. Experiments are conducted to evaluate our proposition. Results are compared with RNG reduction which takes into account only the strength of the received signal (RSSI). Results show that our algorithm maintains network connectivity longer than solutions from the literature and balances the energy consumption over nodes.

Keywords: RNG, topology control, wireless sensor networks, battery-level.

1 Introduction

Wireless sensor networks have been under big scientific interest past few years. They are intended to be deployed in hostile environments (battlefield, forest, etc.). Therefore, it is expected that a large number of cheap simple sensor devices will be randomly scattered over the region of interest. These devices are powered by batteries and have limited processing and memory capabilities. Among numerous challenges faced while designing WSN protocols, maintaining connectivity and maximizing the network lifetime stand out as critical considerations. Important results have been given aiming at generic solutions which will be easily implemented on the variety of platforms used for building wireless sensor networks.

Topology control via per-node transmission power adjustment has been shown to be effective in extending network lifetime and increasing network capacity (due to better spatial reuse of spectrum). Generally, nodes compute a graph reduction on their local view of the network, such as Gabriel Graph (GG) or Relative Neighborhood Graph (RNG) [14]. In the RNG construction, in every triangle *uvw*, the link with the worst value is logically removed in the outcome graph. This graph reduction allows a node to consider only a subset of its neighborhood by keeping connectivity. Then, a node adjusts its range such that the further node in the graph reduction can still be reached.

Most common way to apply graph reduction is by using Euclidean distance as a weight function for reduction. Connection between received signal strength (RSSI) of the sensor nodes and distance between sensors allows usage of RSSI as the indicator of this distance. The distance between the nodes and RSSI are inversely proportional.

In addition, RSSI gives a value on the quality of links. The underlying idea is that longer edges are the most expensive ones so removing them improves energy efficiency. Nevertheless, these methods do not consider the energy level of nodes. And if a node exhausts its battery and fails, it may disconnect the network.

In this paper, we introduce a novel algorithm for topology control which takes into account battery level of each sensor. Battery level is considered as normal or critical if under a threshold. We give appropriate weight to each link in the graph according to the battery levels of its end nodes. We introduce a metric called *power factor* which divides links into three sets:

- reliable links – the ones with normal battery level on both ends (power factor = 0),
- suspicious links with critical battery level on one of the sides (power factor = 1) and
- bad links with critical battery level on both ends (power factor = 2).

By using this metric as main weight in the graph reduction, reliable links are preferred to suspicious and bad links and suspicious links are preferred to bad links. As a result, weak nodes have a limited role in the graph connectivity and have no impact on the network connectivity when they fail. Graph reduction is recomputed periodically since battery level evolves during the time. This allows extending the whole network lifetime. To validate our scheme, we run experiment on real nodes. Experimentation results show that our algorithm maintains network connectivity longer than solutions from literature, balancing connectivity over nodes when needed.

The remaining of this paper is organized as follows. Related works are presented in Section 3. Section 2 presents our model and assumptions. Our main contribution is introduced in Section 4 and evaluated in Section 5. Finally, we conclude and discuss future works in Section 6.

2 Preliminaries

2.1 Model

We consider a sensor network where nodes are randomly scattered, aware of their geographical location and able to tune their transmission range between 0 and R ($R > 0$). The network is modeled as a graph $G = (V, E)$ where V is the set of sensors and E is the set of edges. $uv \in E$ if and only if there exists a radio link between sensors u and v, *i.e.* they are in communication range of each other. We denote by $N(u)$ the set of physical neighbors of node u, *i.e.* the set of nodes v such that $uv \in E$. Let $\delta(u) = |N(u)|$ be the cardinality of $N(u)$, also called the degree of node u. We also define $N_{RNG}(u) \subset N(u)$ the set of RNG neighbors of node u. Every node is aware of its battery level, denoted as $BL(u)$ for node u. We consider that every node u has a unique identifier. We denote the identifier of node u as $ID(u)$.

2.2 Assumptions

We assume that a node u is aware of every edge within its neighborhood, *i.e.* node u knows every edge vw such that $v \neq w$ and such that $v \in \{u \cup N(u)\} \wedge \{w \in N(u)\}$. This

can be achieved by two ways: either nodes are aware of their positions and of the one of their neighbors (nodes broadcast their position in Hello messages) or nodes are aware of their 2-neighborhood (nodes broadcast their neighbor list in Hello messages). If nodes are aware of their positions, they can easily compute the edge length. If they are not, we assume that node u can estimate the distance between itself and its neighbor v by the use of the RSSI. The use of the RSSI has an additional feature. Indeed, even if it is not always inversely proportional to the distance, it gives an indication on the link quality. Thus if the link is short and it has a low RSSI, that means that the quality is poor and equivalent to a very distant node.

3 Related Work

Graph reductions are often used for topology control in wireless sensor networks. Almost every topology control is based around some graph reduction strategy with the aim to keep certain parts and properties of the graph and optimize communication between the nodes and energy used. Indeed, the idea is to perform a graph reduction and then adjust the transmission power of every node in such a way that they can reach the further neighbor in the resulting graph. Some of the graph reductions are based on the geometric properties of the graphs and propose topology control using certain geometric structures like minimum spanning trees [9], Delaunay triangulation [3] or Gabriel graph and some propose usage of cluster based-algorithms, like connected dominating sets. A major drawback of techniques such as minimum spanning trees and Delaunay triangulation is the lack of localized properties of the algorithm.

Relative neighborhood graph was first presented in [14]. The principle is as follows. In every triangle, the link with the greatest weight is removed. RNG is widely used and on most of utilizations the weight considered is the distance [1,12]. So, the longest edge in each triangle is removed. RNG has the great advantage to be computed locally (every node only needs to know its neighbors and links between them) and it is not limited to the usage of geometric properties of the graph. Other variants use the RSSI [4] or the expected transmission count (ETX) [8, 15] as a metric. While it can be used as a notion of distance RSSI still cannot be considered as precise and reliable enough to be solely used. RSSI and ETX are mainly used as quantitative values of link quality. RSSI as a possible metric for RNG graph reduction is much discussed and while some authors state that it is under appreciated [11] others are advocating against it stating its unreliability as the major problem.

In [5], authors apply a primary filter before performing a RSSI-based RNG. Indeed, a node first observes its neighborhood and removes every neighbor for which the RSSI is to low or not stable enough. Topology control is built by using the RSSI as the weight function. Algorithm runs in three steps: *(i)* neighborhood discovery – in which each node is sending Hello messages in order to find out its neighbors and the quality of the connection with them (value of the RSSI which is implicitly sent as the part of the Hello message), *(ii)* filtering possible neighbors and removing the ones with RSSI under a given threshold level – in this way unreliable links are removed before applying topology control *(iii)* RNG reduction – RNG algorithm is applied to all nodes, it is run in a distributed way, meaning that each node u runs RNG algorithm on its neighborhood,

removing node $v \in N(u)$ from the list of its RNG neighbors if there exists a node w such that:

$$\exists w \in \{N(u) \cap N(v)\} \mid ((RSSI(u,v) < RSSI(u,w)) \wedge (RSSI(u,v) < RSSI(v,w)))$$

Problems of energy consumption and energy conserving are also one of the main interests in area of wireless sensor networks [2]. Great number of proposed algorithms are built to be energy efficient. Using different types of batteries with different capacities has an impact on the lifetime of the wireless sensor network presented in [6], [7] and [10]. In these papers authors analyze the problems of different battery levels from the wireless sensor network lifetime extension point of view. However these papers do not consider dynamic reconfiguration of the network in cases when some of the nodes reach certain battery levels, rather just the problem when one of the nodes disconnects due to the loss of power.

In this paper, we propose to apply a battery level based RNG in conjunction with RSSI value in order to take the advantage of works from the literature. We consider dynamic reconfiguration of the graph when battery levels of nodes evolve in time. The usage of specific hardware allowed us to have information about the critical battery levels of the sensor nodes. These two informations, along with the unique ID of each sensor node (also hardware specific property) forms the metric which is then used for topology control using relative neighborhood graph.

4 Battery-level based RNG

In this section, we introduce our topology control algorithm. It is a Relative Neighborhood Graph in which the metric to be considered takes into consideration the battery level of nodes. The main idea is that nodes that are running out of energy should be given less importance and appear as leaves in the resulting graph. In this way, when their battery is exhausted, the network will not be disconnected and we avoid costly graph re-computing.

Nevertheless, the battery level can not be used directly by the algorithm since battery level is related to a node and RNG graph uses metrics related to links. Value assigned to link uv between nodes u and v should be the same from the point of view of u and from the point of view of v to avoid links to be removed improperly and network disconnections. Therefore, we first assign a value to edges based on node battery levels.

4.1 Making Connection from Voltage Level to RNG Weight Function

In order to make a connection between battery level of two sensor nodes on each link, we introduce a value called *power factor*. Power factor of link kl, noted $PF(k,l)$ is determined using the voltage level of the two nodes on the link and is such that $PF \in \{0,1,2\}$. We consider a battery level threshold τ. If the battery level of a sensor node is lower than this threshold, the node is considered as a critical node. Power Factor value is assigned to every edge by using Algorithm 1 that takes as an input value battery level threshold τ and battery level for all nodes. According to these values, Algorithm 1

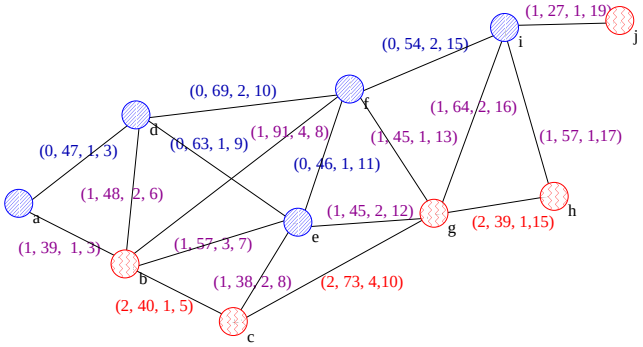


Fig. 1. Example for PF computation. Blue nodes have a high battery level. Red nodes have a low battery level. Every link is assigned with a tuple $(PF, distance(RSSI), |Id1 - Id2|, |Id1 + Id2|)$

divides sensor nodes into two sets: *(i)* normal battery state nodes, with battery level higher than τ, and *(ii)* critical battery state nodes, with battery level lower than τ. Then depending on the battery state of each sensor node on the link, the power factor is assigned to each link as follows:

- power factor 0 – if both nodes are in normal battery state,
- power factor 1 – if exactly one of the nodes is in critical battery state,
- power factor 2 – if both nodes are in critical battery state.

Algorithm 1 is run on node u. It assumes that node u is aware of the battery level of each of its neighbors and knows whether there exist a link between the two of them. This information can be achieved through the use of Hello messages in which every node piggybacks its battery level and its position if available otherwise its neighborhood table. Based on this, node u is able to identify any triangle within its neighborhood and then to assign power factor value on every of these links.

Algorithm 1 Calculate **Power Factor**

1: **foreach** $k, l \in N(n) \mid k \neq l$ **do**
2: **if** $(BL(k) > \tau) \wedge (BL(l) > \tau)$ **then**
3: $PF(k, l) \leftarrow 0$
4: **else if** $(BL(k) > \tau) \wedge (BL(l) \leqslant \tau) \vee (BL(k) \leqslant \tau) \wedge (BL(l) > \tau)$ **then**
5: $PF(k, l) \leftarrow 1$
6: **else**
7: $PF(k, l) \leftarrow 2$
8: **end if**
9: **end for**

To illustrate the Power Factor computation, let us consider Figure 1. Nodes that appear in blue are nodes which battery level is higher than τ while red nodes are the

ones which battery level is lower than τ. On this figure, link *ad* gets a Power Factor equal to 0 since both nodes *a* and *b* have a high battery level. Link *gh* gets a Power Factor of 2 since it connects two nodes with a low battery level. Link *bc* connects two nodes with different battery levels and thus gets the Power Factor value 1.

4.2 Algorithm for graph planarization using power factor

Once Power Factor is computed on every link, RNG computation can be performed. Nevertheless, since Power Factor only returns 3 different values, in a triangle, several edges could hold the same Power Factor value. Thus we need to use a second metric that allows nodes to choose between the nodes in such a way that every node of the triangle take the same decision regarding edge removal to avoid network disconnections. To do so, we apply the traditional RNG metrics, *i.e.* Euclidean distance between nodes and then, to break any potential additional ties, the node identifiers. To ensure that every node computes the same value for a link, we first compute the difference between the identifiers of the two end nodes. Nevertheless, if node with ID 2 is connected to nodes with ID 1 and 3, the two differences of ID remains the same. So, in such a case, we consider afterwards the addition of IDs. Using both difference and addition of the nodes IDs, we ensure that every node computes the same values for a link and that two links in a triangle can not have the same values since nodes IDs are unique and we can not have at the same result for both addition and difference for two different links in a same triangle. Figure 2 shows the different values considered for every link. Every link holds a set of values $(PF, d, ID-, ID+)$ where PF is the Power Factor, d is the Euclidean length of edge, $ID-$ is the difference between ID of end nodes and $ID+$ is the addition of ID of end nodes. As ID we have considered the rank of the letter in the alphabet. For instance, $ID(a) = 1$, $ID(b) = 2$, etc....

We define $\prec$ as a binary total order such that $uv \prec uw$ if and only if

- $PF(uv) < PF(uw)$ or
- $PF(uv) = PF(uw) \wedge d(u,v) < d(u,w)$ or
- $PF(uv) = PF(uw) \wedge d(u,v) = d(u,w) \wedge |ID(u) - ID(v)| < |ID(u) - ID(w)|$.
- $PF(uv) = PF(uw) \wedge d(u,v) = d(u,w) \wedge |ID(u) - ID(v)| = |ID(u) - ID(w)| \wedge |ID(u) + ID(v)| < |ID(u) + ID(w)|$.

Yet, the edge removal runs as described by Algorithm 2. We assume that node u is aware of the length of every edge within its neighborhood, either because nodes are aware of their positions and broadcasts it in Hello messages to their neighbors, or because they are able to estimate it based on RSSI for instance. Every node considers every triangle within its neighborhood (like triangles fig, igh or efg on Fig. 2) and determines what edge to logically remove in the RNG. To do so, it first compares the PF values of nodes. If one of edges has a lower PF value than other ones, it is removed (*e.g.* link *gh* is removed in triangle *igh* on Figure 2).

If two edges have the same lower PF value or than every edge has the same PF value, the longer one is removed. For instance, in triangle fig on Figure 2, the longer edge is fi and should be removed in a traditional RNG algorithm. But f has a PF value of 0 and thus it is kept. Instead, we remove edge *ig* which has the same PF value than edge fg but is longer.

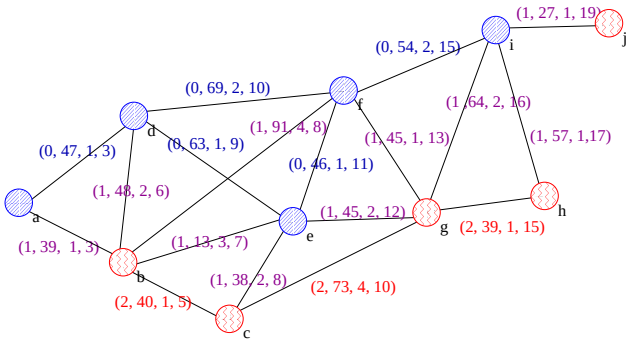


Fig. 2. Example of value computation. Blue nodes have a high battery level. Red nodes have a low battery level.

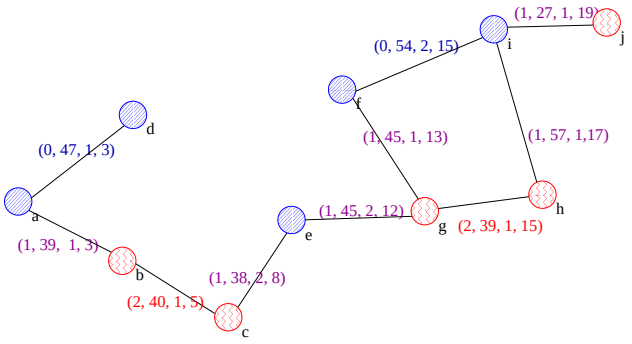
If edges hold the same PF value and are of the same length, ties are broken by considering the difference between identifiers of two nodes on the end of each edge. The edge with the largest ID difference is removed. This is for instance the case in triangle efg on Fig. 2 where edges eg and fg hold the same PF value and have the same length. Their ID is used to differentiate them. Since $|(ID(f) - ID(g)| < |(ID(e) - ID(g)|$, link eg is removed.

Algorithm 2 Calculate *RNG*

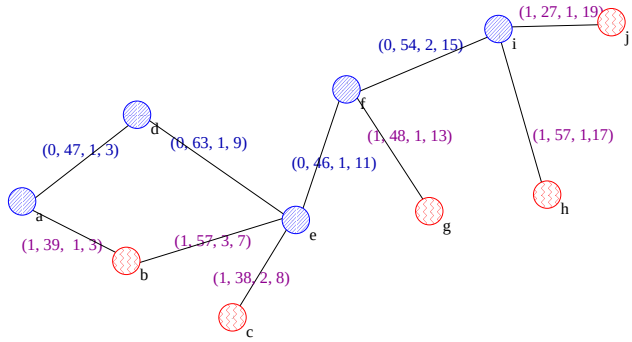
1: $N_{RNG}(u) \leftarrow N(u)$
2: **CalculatePowerFactor**(u)
3: {Node u computes PF factor of every link within its neighborhood.}
4: **foreach** $v, w \in N(u)$ **do**
5: **if** $uv \prec vw \prec uw$ or $vw \prec uv \prec uw$ **then**
6: $N_{RNG} \leftarrow N_{RNG}(u) \setminus \{w\}$ {Link uw is removed from the RNG.}
7: **else**
8: **if** $uw \prec vw \prec uv$ or $vw \prec uw \prec uv$ **then**
9: $N_{RNG} \leftarrow N_{RNG}(u) \setminus \{v\}$
10: {Link uv is removed from the RNG.}
11: **else**
12: {Link vw will be removed from the RNG.}
13: **end if**
14: **end if**
15: **end for**
16: Return $N_{RNG}(u)$

Execution of this algorithm is distributed in the sense that each node is calculating its own set of RNG neighbors, according to given condition and knowing its neighborhood. As a result, we obtain a connected graph in which weaker edges have been removed and where disappearance of weak nodes has a minimal impact of the graph

connectivity. Figure 3 compares the graph obtained after topology control when applying distance-based RNG (Fig. 3(a)) and our battery level based RNG (Fig. 3(b)). As we can see, in our approach, critical nodes appear either as leaves in the graph (nodes c, g, h, j) or in a redundant path (node c). Yet, if one of these nodes fail, the network is not impacted. At contrary, in the traditional RNG, there is no battery level concern and critical nodes belong to principal paths. If node b, c or g fails, the network is disconnected.



(a) Distance based RNG



(b) Battery level based RNG

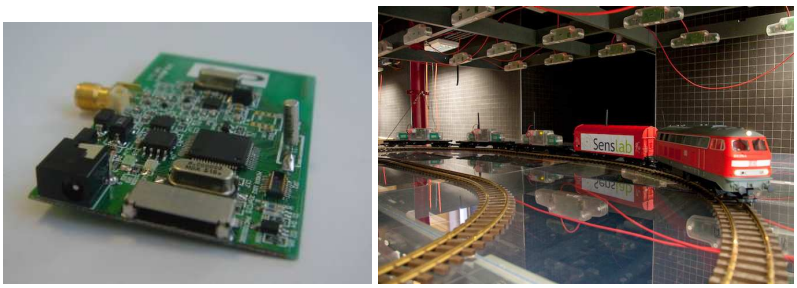
Fig. 3. Example for PF computation. Blue nodes have a high battery level. Red nodes have a low battery level.

5 Implementation and results

In order to validate our proposition and highlight its valuable features, we run experimentations over the SensLAB[1] platform on the Lille site. It is worth noting that performances of our solution strongly depend on the hardware architecture of sensor nodes

[1] http://www.senslab.info/

used for the experiments, more precisely to the microcontroler that is being used on this architecture. This platform is composed of WSN430 sensor nodes (see Fig. 4(a)). On the Lille platform, these sensors are equipped with MSP430F1611, 16 bits RISC microcontroller, ultra low power multichannel RF transceivers CC2420, unique identifier $DS\,2411$, serial flash memory $M25P80$.



(a) WSN430 (b) SensLAB platform

Fig. 4. WSN430 and SensLAB platform. INRIA / Photo N. Fagot

5.1 Experimentation Set Up

We run our experiment on the Lille SensLAB platform. We select a 5×6-node grid via the SensLAB interface spread as depicted by Figure 5. In the SensLAB Lille platform nodes are placed in grid on the distance of $d = 60cm$ between each other. We chose a subset of nodes on the grid leaving out some of the nodes such that we can see what happens with physically longer links and how the algorithm is applied to them.

Microcontroller MSP430F1611, as the whole MSP430x series, is equipped with circuitry called *Supply Voltage Supervisor* – SVS. This part of microcontroller follows voltage level of the power supply of the microcontroller and gives user information when power supply drops under fixed voltage level. SVS can detect 16 discreet voltage levels ranging from 3.7V to 1.9V [13]. We use critical voltage level, $\tau = 3.7V$, which corresponds to value 1110 in the SVS register, *i.e.* we consider that a node has reached a low battery level as soon as the microcontroller sends an information that the critical voltage is reached. Experiments are run 12 hour long, during this time all sensors are loaded with same program which runs RNG algorithm 2, calculating RNG neighbors in distributed way, and recording statistics – neighbor candidates, RNG neighbors and the parameters for each link (power factor, RSSI value, IDs). For the implementation of our algorithm we use CSMA/CA MAC layer implementation provided by the SensTools project [2] and FreeRTOS port for MSP430 microcontrolers.

Our solution is compared to [5] from the literature. As detailed in Section 3, in [5], a RNG is built based on the RSSI on links after a first filter on neighborhood. Our

[2] http://senstools.gforge.inria.fr/

algorithm also uses the RSSI as a metric to estimate at the same time distance and link quality as claimed in Section 2.2 but only as a secondary weight. We use the Power Factor metric as primary one. We do not apply the filter used in [5] since we assume that these bad links will be automatically removed in the RNG computation except if the removal of these links disconnects the network.

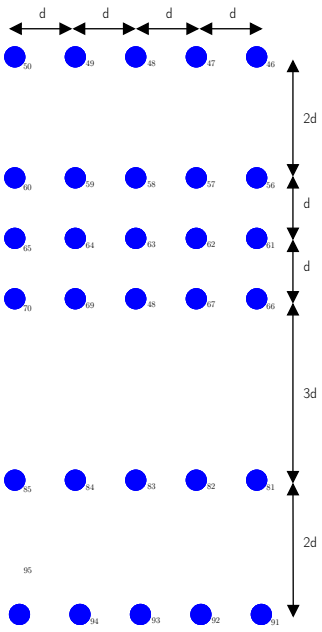


Fig. 5. Placement of the nodes on which the experiment is run. In SensLAB, $d = 60cm$.

5.2 Experimental results

Figure 6 shows the edges connecting node 46 to its neighbors after the initial exchange of HELLO messages. For the sake of visibility, we have represented only these links.

Figure 7 shows after a topology control performed with our solution (Fig. 7(a)) and with RSSI-based RNG with filters (Fig. 7(b)) like in [5]. At these pictures black edges are RNG edges of the node 46, and yellow edges are RNG edges of the rest of the nodes, ensuring the connectivity between node 46 and the rest of the nodes in its neighborhood. At this step, all nodes have a high battery level and thus both graphs are equivalent. We are running test program on the nodes making them exhaust the battery, this program is just made for the nodes to exchange messages and faster discharge batteries thus to speed up the experiment.

Figure 8 shows the final topology control after some time, when some of the nodes exhausted their batteries and brought them to the critical state. We can see that with our solution (Fig. 8(a)), the network is still connected and that weak nodes appear as

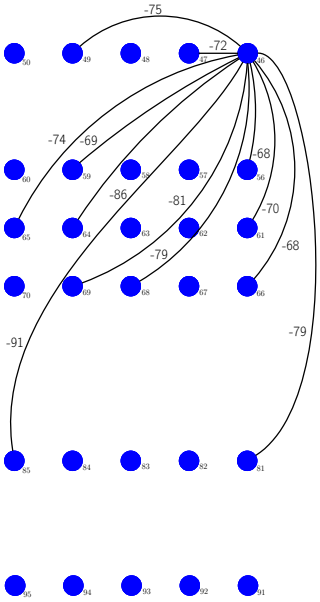
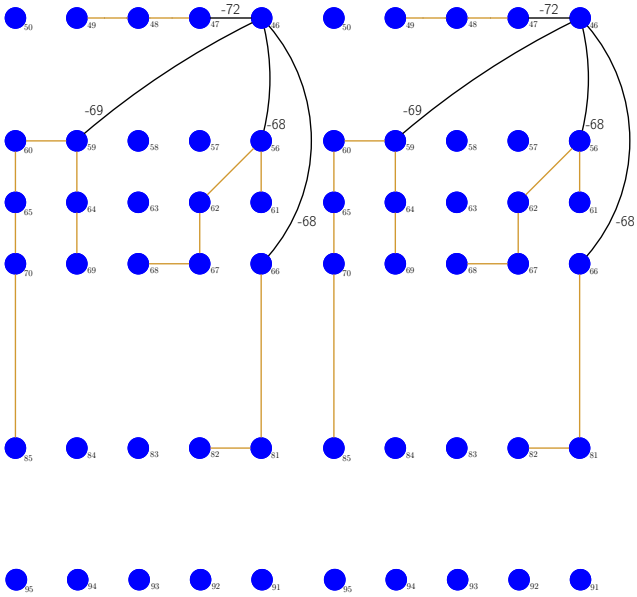
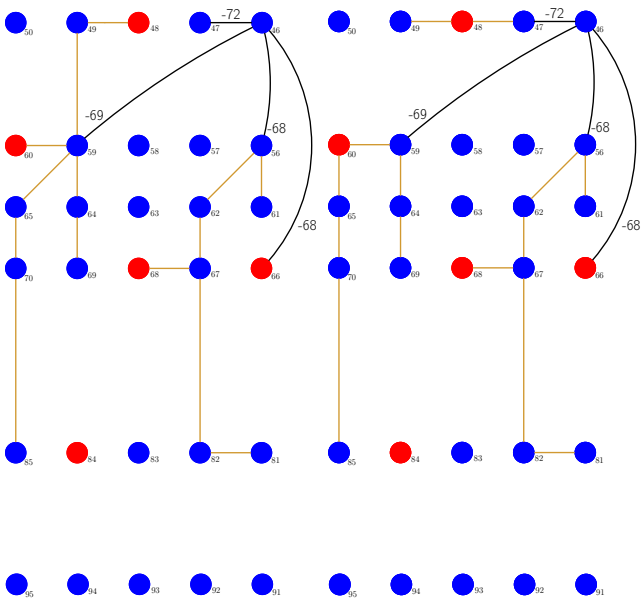


Fig. 6. Initial topology of graph.



(a) Battery-level based RNG (b) RSSI-based RNG + filters

Fig. 7. Topology after topology control.



(a) Battery-level based RNG (b) RSSI-based RNG + filters

Fig. 8. Topology after topology control after nodes have exhausted their battery.

leaves in the reduced graph. We can also see that the network dynamically reorganized itself: link 46 – 47 – 48, which allows node 46 to reach node 48, has been changed for link 46 – 59 – 49 – 48 when the battery of node 48 has dropped under critical level. Furthermore, if we consider complete discharge of the battery of the node 48 in this case if we are using RSSI-based RNG then we will lose of connectivity between nodes 46 and 49 while battery-level based RNG is preserving connectivity between those two nodes.

6 Conclusion

In this paper, we have introduced a new weight for applying RNG and topology control in wireless sensor networks. This weight is primarily based on battery level and is used in such a way that links between and towards weak nodes are taken only if there is no other path. Our algorithm extends the network lifetime. Algorithm has been validated through experimentations. As future works, we intend to evaluate our solution more deeply to measure more precisely the dynamics of the network on different hardware in order to measure the impact of this latter. We also wish to investigate the impact of the number of power factor values. Indeed, nodes are only classified in two categories (high or low battery level). What if we introduce an additional level? At last, we intend to evaluate our algorithm in mobile environments.

7 Acknowledgements

This work was partially supported by CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante and the ANR BinThatThinks project.

References

1. J. Cartigny, F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Localized lmst and rng based minimum-energy broadcast protocols in ad hoc networks. *Ad Hoc Networks*, pages 1–16, 2005.
2. A. Ephremides. Energy concerns in wireless networks. *Wireless Communications, IEEE*, 9(4):48 – 59, august 2002.
3. L. Hu. Topology control for multihop packet radio networks. *Communications, IEEE Transactions on*, 41(10):1474–1481, August 2002.
4. F. Khadar and D. Simplot-Ryl. Connectivity and topology control in wireless ad hoc networks with realistic physical layer. In *Third International Conference on Wireless and Mobile Communications, 2007. (ICWMC '07).*, page 49, march 2007.
5. F. Khadar and D. Simplot-Ryl. From theory to practice: topology control in wireless sensor networks. In *MobiHoc'09*, pages 347–348, 2009.
6. Hengyu Long, Yongpan Liu, Yiqun Wang, Robert P. Dick, and Huazhong Yang. Battery allocation for wireless sensor network lifetime maximization under cost constraints. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, ICCAD '09, pages 705–712, New York, NY, USA, 2009. ACM.
7. Mihail L.Sichitiu and Rudra Dutta. Benefits of multiple battery levels for the lifetime of large wireless sensor networks. In Raouf Boutaba, Kevin Almeroth, Ramon Puigjaner, Sherman Shen, and James P. Black, editors, *NETWORKING 2005*, volume 3462 of *Lecture Notes in Computer Science*, pages 403–408. Springer Berlin / Heidelberg, 2005.
8. Milan Lukic, Bogdan Pavkovic, Nathalie Mitton, and Ivan Stojmenovic. Greedy geographic routing algorithms in real environment. In *MSN*, pages 86–93, 2009.
9. R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 404 –413 vol.2, 2000.
10. Dongfeng YUAN Ruihua ZHANG, Zhiping JIA. Analysis of lifetime of large wireless sensor networks based on multiple battery levels. *Int'l J. of Communications, Network and System Sciences*, Vol.1 No.2:136–143, 2008.
11. Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*, 2006.
12. K. J. Supowit. The relative neighborhood graph, with an application to minimum spanning trees. *J. ACM*, 30:428–448, July 1983.
13. Texas Instruments. *MSP430x1xx Family - User's Guide*, mixed signal products edition, 2006.
14. Godfried T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.
15. R. Wattenhofer and A. Zollinger. XTC: a practical topology control algorithm for ad hoc networks. In *Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, MN, USA, 2004.