

Heuristics for Designing Energy-efficient Wireless Sensor Network Topologies

Andréa Cynthia Santos, Fatiha Bendali, Jean Mailfert, Christophe Duhamel,
Kean-Mean Hou

► To cite this version:

Andréa Cynthia Santos, Fatiha Bendali, Jean Mailfert, Christophe Duhamel, Kean-Mean Hou. Heuristics for Designing Energy-efficient Wireless Sensor Network Topologies. Journal of Networks, Academy Publisher, 2009. hal-02082851

HAL Id: hal-02082851

<https://hal.uca.fr/hal-02082851>

Submitted on 11 Apr 2019

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.

Heuristics for Designing Energy-efficient Wireless Sensor Network Topologies

Andréa C. Santos, Fatiha Bendali, Jean Mailfert, Christophe Duhamel, Kean-Mean Hou
 LIMOS, Université Clermont-Ferrand II, Campus des Cézeaux, 63170, Aubière, France
 Email: andrea@isima.fr, bendali@math.univ-bpclermont.fr, {mailfert, duhamel, kun-mean.hou}@isima.fr.

Abstract—Wireless Sensor Networks (WSN) have been studied in several contexts. There are many challenges involving WSN design such as the energy resources optimization, the robustness and the network coverage. We address here the problem of energy-efficient topology design. A well-designed dynamic topology and efficient routing algorithms may allow a large reduction on the energy consumption, which is one of the main concerns of WSN nodes. In this work, we propose to model the problem of clustering a WSN topology as a variation of the independent dominating set optimization problem. Then, we describe two heuristics to generate a WSN topology and two ways to evaluate the energy consumption. Computational results are presented for instances with up to 500 nodes.

Index Terms—Wireless sensor networks, clustering, optimization, independent dominating set, heuristics.

I. INTRODUCTION

A wireless Sensor Network (WSN) consists of a set of wireless sensor nodes distributed in an area. A wireless sensor node (sensor) is usually composed of limited resources such as battery, CPU and memory. For example, the LiveNode has an ARM7TDMI core running at 48 MHz and 64 Kb of on-chip SRAM [1]. The sensors called *sink nodes* are more powerful: they have a stronger battery, a larger memory and more computational power. They are typically used to coordinate the sensor network and to collect all the data.

WSN have a large number of practical applications such as environmental monitoring [2], biological detection, automatic manufacturing, smart care, smart home (security) [3], etc. However, in order to provide a good service, it is needed to deal with difficult technical problems such as maximizing the network lifetime. Some works in the literature consider this problem as an optimization problem. For instance, the coverage problem with minimal energy consumption is modeled as an integer linear program in [4] and it is solved by a distributed algorithm in [5]. Minimizing the energy consumption for unclustered WSN over the time is also modeled as an integer linear program in [6].

The topology of a sensor network changes along its lifetime since nodes may fail due to battery shortage. This evolution characterizes ad-hoc networks. A WSN differs from mobile ad-hoc network in several points such as: hardware has a lower performance, mobility is reduced, and sensors are more densely distributed. In terms of WSN communication routing protocol, it usually does not

change along the network lifetime. Communications can be done, for example, by sending messages periodically or by continuously sending them from all sensors, or by sending messages only when an event occurs. In fact, the network topology and the routing protocol are not independent from each other and their interaction has a strong impact on the performance and on the lifetime for such networks. If the network topology is poorly designed, communication can fail. Moreover, if the routing protocols are not well defined, they may waste energy: for example, by sending more messages than necessary due to packet loss or due to loops in the routing topology. Usually, a wireless sensor node consumes about 75% of its energy in using wireless access medium.

Let a *cluster* be a subset of sensors nodes which are usually organized in a hierarchical way: masters nodes collect data, slaves nodes perform sensing activities, and bridge nodes perform inter-cluster communication. Many works use clusters as a base structure for the network topology [7]–[12], while others do not adopt clusters [13]–[15]. A comparison between these two approaches is done in [16]. Results show that cluster-based topology is more efficient when the sensors within each cluster usually measure highly-correlated data. In such situations, the cluster head can aggregate the sample data from all the sensors into a single message, thus reducing the amount of sending messages. In this work, we consider the problem of designing a WSN cluster-based topology to maximize the network lifetime.

A cluster-based architecture contains some structural properties on its communication graph which are used to design the WSN topology. We model the topology design problem as a variation of the independent dominating set problem. Heuristics are proposed to compute the topology having those special properties. Moreover, two different ways of evaluating the topology in terms of energy consumption are suggested. In the first evaluation, the energy consumption is measured by computing a maximum flow. In the second one, the topology is evaluated by using two criteria: the number of clusters and the hop average. The communication rules are not explicitly defined. Yet, the evaluation result provides valuable informations about paths that can be used by each sensors to send messages. Computational results are presented for instances with up to 500 nodes.

This work is organized as follows: the problem is defined in Section II. It is then modelled using graph

theoretical elements in Section III. Algorithms are detailed in Section IV. Computational results are presented in Section V. Some concluding remarks and perspectives are given in Section VI.

II. PROBLEM DEFINITION

We consider a set of identical sensors which differ only by their identification name and a unique sink node. We assume that each sensor is able to know its location and that the sink node does not act as a sensor itself. Moreover, sensors are organized in clusters as in [16]: in a cluster, sensors have a logical classification in masters, slaves and bridges. Slaves and bridges can be used to sensing activities, and masters only collect the data from the slaves and monitor the cluster. The slaves only transmit data to their master and the bridges also allow inter-cluster communication as they have to belong to at least two clusters. Figure 1-(a) represents a set of sensors along with the sink node (black square). Figure 1-(b) illustrates a cluster-based network topology with the sink node (black square), the master nodes (black circles), the bridge nodes (circles with a cross) and the slave nodes (white circles). The dotted circles define the cluster associated to each master node. As can be seen, several bridges may belong to the intersection of two clusters and a bridge can also belong to more than two clusters.

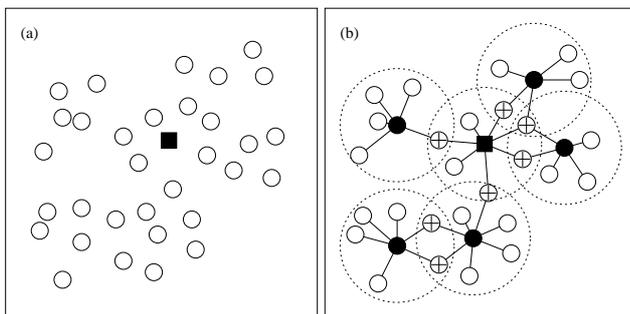


Figure 1. a set of sensors and an associated network topology.

The topology design problem consists in defining a communication network over the sensors and in defining routes to send the data from each sensor to the sink. The sink is known in advance, thus the core of the problem is to partition the remaining nodes into masters, bridges and slaves. The objective is to maximize the network lifetime induced by the resulting topology. The network topology will change the first time a cluster is unable to send its data to the sink. Three situations are responsible for such a cluster failure: the master runs out of battery, a set of bridges in the way to the sink runs out of battery or all slaves in the cluster run out of battery. Under this definition, maximizing the network lifetime is equivalent to the maximal number k of messages all the sensors can send to the sink before the occurrence of the first failure: sending k messages from each sensor is possible while sending $k+1$ is not possible without altering the topology.

Thus, the optimization problem is to define the topology of the WSN in order to maximize the network

lifetime. When a cluster fails, the topology can be re-evaluated in order to re-establish connection with the sensors still active in the cluster. This can have a local impact on the topology.

III. MODELLING THE WSN TOPOLOGY AS A GRAPH

The WSN topology problem is modelled as a finite, loopless, connected and undirected graph $G(V, E)$, where V is the set of nodes which corresponds to the sensor nodes and E is the set of edges. Let $[u, v] \in E$ be an edge with endnodes $u, v \in V$, it means that sensors u and v can communicate. The sink node $s \in V$ plays an important role in the network since any data which circulates in the communication network must reach it.

During the initialization step of the WSN, each sensor sends its identification in such a way the neighbourhood structure is progressively built, as in [8]. Thus, we assume the sink node has the full information about the neighbourhood structure after running the initialization algorithm. This structure corresponds to the communication graph G mentioned above. Figure 2 illustrates the communication graph obtained from Figure 1-(a), assuming a small radius of wireless communication.

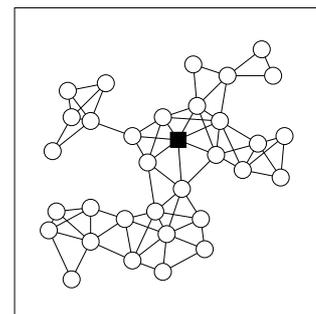


Figure 2. A communication graph G .

Let $M \subset V$ and $S \subset V$ be the set of sensor nodes that will respectively act as masters and slaves. Bridge nodes are slaves which also ensure connections between pairs of master nodes. In terms of graph, an *independent* set $T \subseteq V$ is a set of nodes such that no edge of E have both endnodes in T . Moreover, a subset of nodes $D \subseteq V$ is a *dominating* set of G if every node of $V \setminus D$ is adjacent to at least one node of D . Thus, the problem of clustering consists in defining a partition Π of V into two subsets M and S such that the set M satisfies the following properties:

- (P1) $s \in M$,
- (P2) $\forall v \in S, \exists m \in M \mid [v, m] \in E$,
- (P3) $\nexists (m, m') \in M \times M \mid [m, m'] \in E$.

Property (P1) states that the sink node s acts as a master node. Property (P2) determines that any slave node is connected to at least a master node. In fact, this property means that M is a *dominating* set of G . Finally, property (P3) specifies that there is no direct connection between two master nodes. It means that M is also an *independent* set of G . Thus, M is an independent dominating set

of G . In other words, M is a maximal inclusionwise independent set of G .

Properties (P1) to (P3) do not guaranty the final graph representing the WSN topology solution is connected. In fact, the WSN topology problem requires that there is a path connecting any master node to the sink node s . Thus, another property must be defined in G . Let $d(u, v)$ be the minimum number of hops (amount of edges in a path) between $u, v \in V$. A node $v \in V$ is said to be a neighbour of $u \in V$ whenever $d(u, v) = 1$. Whenever M is an independent set of G , let $H = (M, E(M))$ be an induced graph of M such that the following conditions are satisfied:

- (a) $[u, v] \in E(M)$ if $u, v \in M$, and
- (b) $d(u, v) = 2$.

$N(u)$ represents the set of all neighbours of node u , where $[u, v] \in E(M)$ if and only if $N(u) \cap N(v) \neq \emptyset$. It means that there are nodes which are neighbours of both nodes u and v (bridges nodes). Thus, the partition $\Pi = (M, S)$ must also satisfy property (P4).

(P4) H is connected.

To conclude, a cluster-based WSN topology solution consists in defining a partition $\Pi = (M, S)$ which satisfies Properties (P1) to (P4), referred here as the Independent Dominating Set Problem with connecting requirements (IDSC problem). From now on, a feasible solution for the IDSC problem is represented by a graph $G'(V', E')$, with $V' = M \cup S$ and E' is the set of edges connecting the independent dominating nodes to the slaves nodes. Moreover, in G' , a slave node connecting two master nodes is called bridge. Figure 3-(a) illustrates a feasible graph G' for a WSN topology. The black nodes, the white circles, and circles with a cross represent respectively the masters, the slaves, and the bridge nodes. Moreover, the graph H is represented in Figure 3-(b). Observe that black nodes are separated by two edges (two hops).

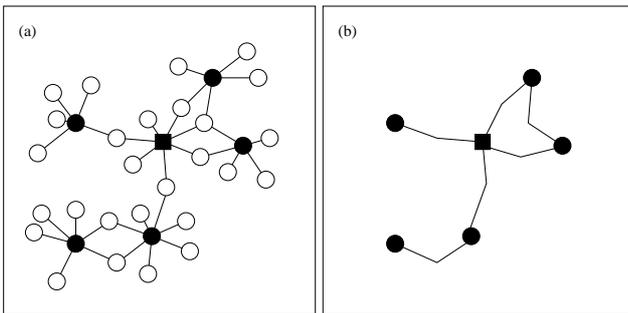


Figure 3. Example of a feasible solution G' and of a graph H .

A natural theoretical question is to know whether from a graph G as defined above, we can always produce a graph G' solution to the IDSC problem or not. In section IV, the Theorem 1 answers this question.

We give below three examples of graph G for which a solution to the IDSC problem is known. The sink node and the master nodes are respectively represented by the black square and the black circles.

- Whenever $G = (A \cup B, E)$ is a connected bipartite graph, then $M = A$ or B , that is $M = \{u \in A \cup B \mid d(s, u) \text{ is even}\}$, see Figure 4-(a).
- When $G = C_p$ is a cycle with p vertices, and without loss of generality, one sink node $s = 1$. In this case, a solution is the set $\{1, 3, \dots, 2\lfloor \frac{p}{2} \rfloor - 1\}$, see Figure 4-(b).
- Finally, when G is a graph with $V = \{s, u_0, \dots, u_n, v_0, \dots, v_n\}$ and $E = \bigcup_{i=1}^n \{[u_0, v_i], [v_0, u_i], [u_i, v_i]\} \cup \{[s, u_0], [s, v_0], [u_0, v_0]\}$, that is, G is not a bipartite graph, but $G \setminus \{s\}$ is, as shown in Figure 4-(c). In this case, the set of masters M is given as follows: $M = \{s\} \cup \{u_i \mid i \in I\} \cup \{v_j \mid j \in \{1, \dots, n\} \setminus I\}$ $\forall I \subseteq \{1, \dots, n\}$ and M and G' satisfy properties (P1) to (P4). For this case, there is an exponential number of subsets that could be masters.

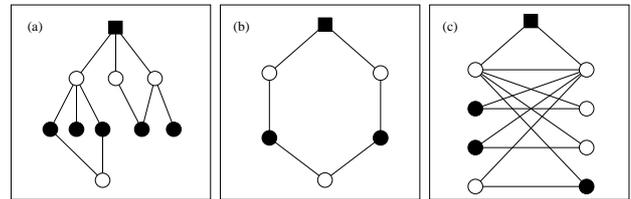


Figure 4. Graphs where the solution for the IDSC problem is known.

IV. ALGORITHMS FOR DESIGNING SOLUTIONS FOR THE IDSC PROBLEM

We propose a three-steps algorithm for designing a WSN topology and to maximize its network lifetime. The first step computes a feasible solution for the IDSC problem. Two ways for generating feasible solutions for the IDSC problem are suggested: a greedy procedure and a random one. The feasible solutions are evaluated through the second step. Two evaluations are proposed: the first uses the maximum flow and the second uses the number of clusters and the hop average. Weights are used to specify the priority each node has to become a master node. They are updated in the last step of our algorithm.

Figure 5 represents an overview of the algorithm. The communication graph $G(V, E)$ and the sink node s are given as input. The three steps, which are detailed below, are carried out until a stopping criterion is met. Then, the output data is the resulting topology graph $G'(V', E')$.

A. A greedy and a random procedures for the IDSC problem

The general idea of the greedy algorithm is to select a node as master at each iteration. The first node to be included to M is the sink node. The graph G' is also expanded at each iteration. A pseudo-code for this procedure is given in Algorithm 1. Moreover, the structure *candidates* corresponds to a list of nodes candidates to become a master node. A pseudo-code for updating the candidate list is presented in Algorithm 2.

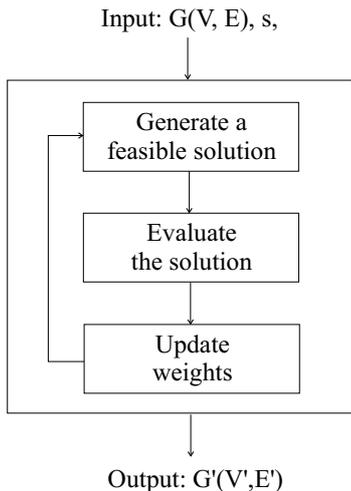


Figure 5. Algorithm overview.

Algorithm 1: Greedy procedure.

Input: $G(V, E), s$

Output: $G'(V', E'), M$

- 1 $M \leftarrow \{s\};$
 - 2 $S \leftarrow N(s);$
 - 3 $V' \leftarrow M \cup S;$
 - 4 $E' \leftarrow \{(s, u) \mid u \in N(s)\};$
 - 5 $candidates \leftarrow$ Update candidate list;
 - 6 **while** $(|V'| \neq |V|)$ **do**
 - 7 select $m \in candidates;$
 - 8 $M \leftarrow M \cup \{m\};$
 - 9 $S \leftarrow N(m);$
 - 10 $V' \leftarrow M \cup S;$
 - 11 **for** $(v \in N(m))$ **do**
 - 12 $E' \leftarrow E' \cup \{m, v\};$
 - 13 $candidates \leftarrow$ Update candidate list;
 - 14 **return** $G'(V', E'), M;$
-

Algorithm 2: Update candidate list.

Output: $candidates$

- 1 $candidates \leftarrow \{ \};$
 - 2 **for** $([u, v] \in E)$ **and** $(u \notin V')$ **and** $(v \in V')$ **do**
 - 3 $candidates \leftarrow candidates \cup \{u\};$
 - 4 **return** $candidates;$
-

The procedure in Algorithm 1 receives as input: a connected graph G and the sink node s . The data structures are initialized in lines 1 to 5. The set of master nodes receives the sink node in line 1 and the slave nodes are updated in line 2. V' and E' are respectively initialized in lines 3 and 4. The candidate list are updated in line 5 (see Algorithm 2). The loop in lines 6 to 13 is repeated until all vertices in V are included in V' . A master node is chosen in line 7. The candidates selection depends on the evaluation procedures used (see Section IV-B). In fact, a weight is associated to each candidate. Weights are

computed using a maximum flow algorithm or simply considering the oldest nodes taken as a master node along the algorithm iterations. The node with the highest weight and with the smallest index is selected. In the first iteration, all nodes have the same weight which is equal to zero. The set of master nodes M , the set of slave nodes S , the set of vertices V' , and the set of edges E' are respectively updated in lines 8 to 12. Finally, the list of candidates are set in lines 13. The algorithm returns G' and M in line 14.

Let us assume that graphs G' and G are accessibly by Algorithm 2. This procedure returns the *candidate list*. The structure *candidates* are initialized in line 1. Then, the list of candidates are updated in the loop in line 2 and 3. For all edge $[u, v]$ belonging to E , node u is a candidate to be a master node (line 2), if u is not in V' and if node v is in V' .

The algorithm worst case complexity is $O(|V| \cdot |E|)$. Initializing data structures in lines 1 to 5 consumes in the worst case $O(|E|)$ steps. The loop in lines 6 to 13 performs in the worst case $O(|V| \cdot |E|)$ steps.

The random procedure slightly differs from the greedy one. Basically, instead of choosing the best candidate as a master node in line 7, a candidate is randomly chosen from the list of candidates. In this case, weights are not computed. The algorithm complexity still remains $O(|V| \cdot |E|)$.

1) *The algorithm correctness:* An important question is to know if the proposed procedure provides a solution which satisfies properties (P_1) - (P_4) . This is ensured by the proof of Theorem 1.

Theorem 1: The set M and graph G' found at the end of Algorithm 1 satisfy properties (P_1) - (P_4) .

Proof. We show that M satisfies properties (P_1) - (P_3) , and H is connected. As graph G' is deduced from the partition $\Pi(M, S)$ and from graph H , it also satisfies properties (P_1) - (P_4) for the set V' at each step of the procedure. Observe that in Algorithm 1, at least one node enters V' at each iteration.

In the initialization step (lines 1 to 4), M contains the sink node, V' is initialized with the sink node and all of its neighbours. Moreover, every edge incident to s is included in E' . Thus, the initial set M satisfies properties (P_1) to (P_3) , and the initial subgraph G' satisfies property (P_4) .

Considering that $V' \neq V$. In this case, as G is connected, the set of edges with one extremity in V' is not empty. Thus, there exists an edge $(s) [u, v] \in E$ such that $u \notin V'$ and $v \in V'$. Therefore, if condition $V' \neq V$ is true, there is always at least one candidate to become master node in line 7.

The candidate to master node m , selected in line 7, is not adjacent to any other node in M , otherwise, m would be neighbour to a node in M . Consequently, it has already been included in V' in line 3 or in line 10. Thus, $M \cup \{m\}$ is an independent set. Let $m' \in M$ be

a node adjacent to a node $v \in V'$. Since $[m', m] \notin E$ and $[m', m] \in H$, H is a connected graph. Thus, the new set M in line 8 satisfies properties (P_1) and (P_3) and graph H satisfies property (P_4) . Furthermore, given a node $u \in V' \setminus M$ and the set V' obtained in line 10, either $[u, m] \in E'$ if u is a neighbour of m , or u was already adjacent to a node belonging to $M \setminus \{m\}$. Hence, M satisfies the property (P_2) . \square

B. Evaluation procedures

Once a solution for the IDSC problem has been computed, its effectiveness in terms of network lifetime has to be evaluated. As mentioned in section II, the network lifetime is interpreted here as the amount of time the WSN can work before having to perform an alteration on its topology.

As a first approach, we propose to model this evaluation as a maximum flow problem. By analyzing the amount of energy needed to a sensing activity, to receive a message and to forward a message, the energy level of each sensor node can be transformed into a maximal amount of messages that the node can handle. This holds for slave, bridge and master nodes.

In the second approach, the number of master nodes and the hop average is used to measure the quality of the solution. The primary criterion is the number of master nodes. The smaller the number of master nodes the better the solution. Between two solutions with the same number of master nodes, the best one has the smallest hop average.

1) *Evaluation using a maximum flow algorithm:* The algorithm of Ford and Fulkerson [17] is used to compute the maximum flow. Applying this algorithm requires some characteristics on the graph such as an unique origin (flow departure), and capacities associated to the arcs. The capacity in a WSN is associated to each sensor node.

The graph $G = (V, E)$ is then converted in a directed graph $D = (N, A)$ to perform the maximum flow algorithm. First, each vertex $i \in V$ is transformed into two nodes i_1 and i_2 in N connected by an arc (i_1, i_2) . The node i_1 corresponds to the entering version of the vertex i , while the node i_2 corresponds to its outgoing version. A capacity representing the maximum number of messages vertex i can afford is associated to each arc $(i_1, i_2) \in A$. We set a theoretical and approximated capacity equal to a thousand messages for slaves, and infinite capacity for master and bridge nodes. It means that each slave can process at most a thousand messages. Moreover, each edge $e = [i, j] \in E$ is transformed into two symmetric arcs (i_2, j_1) and (j_2, i_1) in A . This way, entering nodes receive messages from outgoing nodes. Finally, an artificial source node is added to send flow, and it is connected to each entering node $i_1 \in V$ with infinity capacity.

Using the maximum flow strategy, the paths from masters to the sink nodes can be studied to give more informations about the network data traffic. Arcs $(i_1, i_2) \in$

A whose capacity is saturated correspond to the sensor nodes $i \in V$ that will first fail. Furthermore, the values of the flow on the slave nodes give the number of data they are globally able to send before a failing occurs in the network.

The maximum flow problem in a graph is strongly polynomial using the push-relabel algorithm. Thus, this evaluation is performed in polynomial time.

2) *Evaluation using the number of master nodes and the hop average:* Given a feasible solution using the greedy algorithm suggested in Section IV-A, the number of master nodes is used as a primary evaluation criterion. The second criterion is the arithmetical hop average. It is computed as follows: the minimum number of edges in a path from each master m to the sink s is added and this sum is divided by the total amount of masters. When two solutions have the same number of master nodes, the one with the smallest hop average is considered as the best.

Computing the hop average can be efficiently done by using a breadth-first search on the graph G' . The algorithm starts from the sink node and it stops after having visited all the master nodes. This evaluation takes in the worst case $O(|V| + |E|)$ steps.

C. Update weights

After evaluating a solution, weights associated to each node are updated. Using the first evaluation, see Section IV-B.1, each node $i \in V$ is represented by an arc (i_1, i_2) with an associated capacity. If i is a slave node in G' , its capacity is set to a thousand messages. In this case, the weight associated to i is equal to its capacity minus the number of messages routed by (i_1, i_2) . However, if i is a bridge or a master node (whose capacity is infinite), its weight is simply set to the amount of messages routed by (i_1, i_2) . This way to update weights is biased toward slave nodes which have not routed messages, and toward master and bridge nodes which have routed several messages.

Using the second evaluation, see Section IV-B.2, weights are set to zero for the nodes chosen as a master. For the other nodes, weights are simply incremented in one unit. Using this strategy, nodes with the highest weights have the highest priority to become masters. A lexicographical order is used to select among nodes with the same weight.

The convergence of weights to optimal solutions is not guaranteed, since they are a part of the heuristic. However, as shown in the computational results, the proposed heuristics produce good results. In fact, applying heuristic strategies for designing the WSN topology has some advantages. For instance: the algorithms are polynomial. This means that scalability is ensured, as well as the topology can be efficiently repaired when a cluster fails. Furthermore, the graph communication structure is flexible, then it can be easily modified to use, for example, multiple sinks.

V. COMPUTATIONAL RESULTS

The computational experiments were carried out on an Intel Core Duo with 1.73 GHz clock and 1Gb of RAM memory. The algorithms were developed in C. Four approaches for generating WSN topologies are evaluated. The first and the second approaches use the maximum flow evaluation. The difference relies on the choice of master nodes at each iteration. In the first one, the best candidates are selected as master nodes. In the second one, master nodes are randomly chosen from the candidate list. The evaluation procedure for the third and the fourth strategies is based on the number of clusters and the hop average. They also differ on the choice of master nodes at each iteration. In the third approach, the best candidates are chosen as masters, and in the fourth one, masters are randomly selected from the candidate list.

The instances were generated as follows: $|V|$ nodes are randomly located on an area of $100 \times 100 m^2$. $|V|$ varies from 100 to 500 nodes and the radio range varies from 20 to 60 m.

Several simulations were carried out. For all of them, the algorithms stopping criterion is the number of iterations which is set to 100. In the first simulation, we analyze the evolution of the number of clusters when the amount of nodes varies. The results correspond to the instances with a radio range of 20 m. Figures 6 and 7 illustrate the results using the four approaches. The best number of clusters are reported for each one. Using the maximum flow evaluation (Figure 6), choosing the best candidates as masters leads to four better solutions, while two better solutions are found when masters are randomly selected. For the evaluation using the number of cluster and the hop average (Figure 7), results indicate that the choice of best and of random candidates are quite similar. Applying the best and the random choices for candidates leads respectively to two better solutions and one better solution.

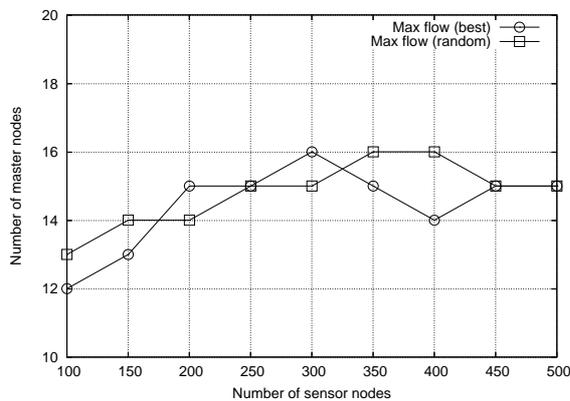


Figure 6. Comparison between the approaches using the maximum flow.

The main difference between the evaluation using maximum flow and the number of clusters and the hop average is the running time. Table I shows the running time for generating a hundred solutions. Each line corresponds to an instance. Columns ($|V|$) and (range) represent

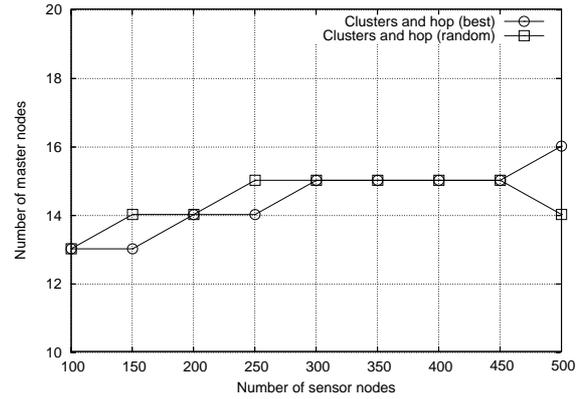


Figure 7. Comparison between the approaches using the number of cluster and the hop average.

respectively the number of sensors and the radio range. For each approach, the running time is given in seconds for the two candidate selections criteria: the best (best) and the random one (random). The evaluation using maximum flow is quadratic, while the other one is linear. Since the time to compute a topology is an important point for a WSN, at a first time, the evaluation using the number of master nodes and the hop average is more suitable. Even though, the maximum flow evaluation remains an interesting strategy that can be useful for some applications and that can be used for giving information about the routing traffic.

TABLE I. RUNNING TIME FOR THE FIRST EXPERIMENT.

| Instances | Maximum flow | Cluster and hop |
|-----------|--------------|-----------------|
| $ V $ | (best) | (best) |
| range | (random) | (random) |
| 100 | 4.91 | 4.94 |
| 150 | 17.33 | 17.33 |
| 200 | 43.48 | 43.53 |
| 250 | 87.31 | 87.47 |
| 300 | 150.84 | 151.33 |
| 350 | 262.66 | 264.58 |
| 400 | 412.86 | 419.36 |
| 450 | 690.14 | 688.11 |
| 500 | 977.67 | 980.81 |

Let us consider the variability of an algorithm as the difference in terms of objective function value between the best and the worst solutions. A robust heuristic produces solutions with a small variability. Table II illustrates the difference in terms of number of clusters, for the best and for the worst solution found using each approach. The variability is quite similar for both strategies. The approach using the number of clusters and the hop average with a random choice of candidates has the highest variability for 5 out of 9 instances. For the other approaches, 4 out of 9 instances have the highest variability.

In the second experiment, only the third and the fourth strategies are used since they are more efficient in terms of CPU time. The goal is to observe the evolution of the number of clusters for the best solution found when the radio range varies. This experiment has been performed

TABLE II.
VARIABILITY BETWEEN THE WORST AND THE BEST SOLUTIONS.

| Instances | | Maximum flow | | Cluster and hop | |
|-----------|-------|--------------|----------|-----------------|----------|
| V | range | (best) | (random) | (best) | (random) |
| 100 | 20 | 5 | 5 | 4 | 5 |
| 150 | | 6 | 5 | 6 | 5 |
| 200 | | 4 | 6 | 5 | 6 |
| 250 | | 5 | 5 | 6 | 5 |
| 300 | | 5 | 6 | 6 | 6 |
| 350 | | 5 | 5 | 6 | 6 |
| 400 | | 7 | 5 | 5 | 6 |
| 450 | | 6 | 7 | 6 | 7 |
| 500 | | 7 | 5 | 5 | 6 |

over all the instances.

Results for the instances with 400 and 500 nodes are displayed in Figures 8 and 9. Results for the other instances are similar. We observe that the solutions have more clusters when the radio range is smaller. This is an expected result. Furthermore, for the set of instances tested, none of the two strategies dominates the other one. They both produce good solutions. It means that, for a practical application, a deeper study is needed using specific instances in order to define which strategy could be integrated into the protocol.

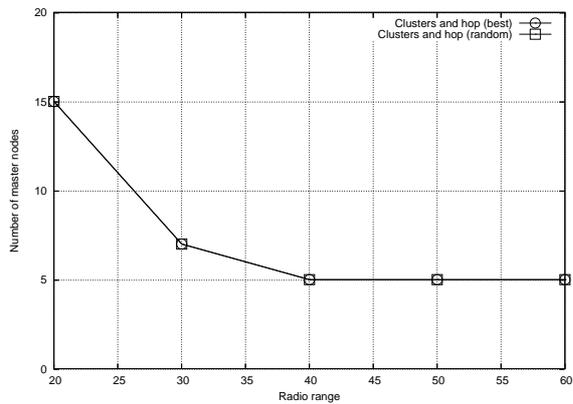


Figure 8. Evolution of the number of clusters varying the radio range for the instance with 400 nodes.

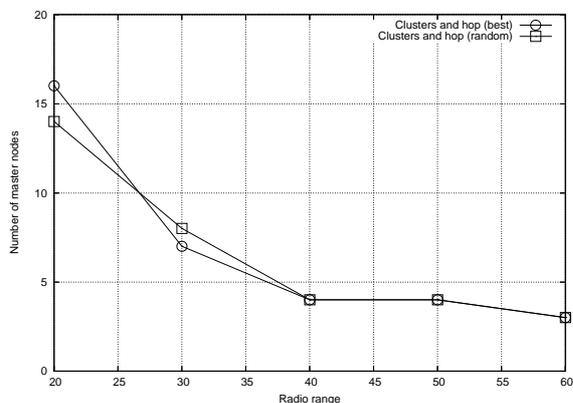


Figure 9. Evolution of the number of clusters varying the radio range for the instance with 500 nodes.

Finally, in the last experiment, a special graph structure is considered. It consists of a regular grid which finds applications, for instance, in agriculture monitoring. In fact, for a regular grid as shown in Figure 10-(a), the greedy heuristic we proposed always finds the optimal solution (see Theorem 1). Figure 10-(a) shows the grid, and the solution computed by the greedy heuristic is given in Figure 10-(b). The sink node, the master nodes and the bridge nodes are respectively represented by the black square, black circles and circles with a cross.

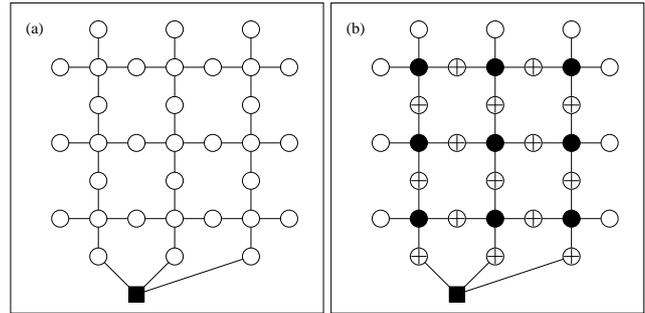


Figure 10. Special regular graph.

VI. CONCLUSIONS

In this work, WSN are formally designed as a graph optimization problem. The problem is associated to a variation of the independent dominating problem with connecting requirements, where some given properties must hold. This is a new optimization approach for clustering a wireless sensor network based on graph theory.

Moreover, four heuristic strategies are proposed in this work for designing WSN topologies. They are compared in terms of quality and efficiency. Those heuristics are promising for several reasons among others: (a) they rely on basic algorithms whose complexity is polynomial. Thus, scalability is ensured for networks having a large number of sensors; (b) the graph communication structure is flexible: for example, the problem is modeled using one sink node, but with a few modifications several sink nodes may be introduced as well; (c) The proposed algorithm contains three independent steps. Thus, new strategies for each step can be easily developed and integrated.

The procedures to measure the topology performance are as follows. The first and the second procedures use a maximum flow algorithm. They differ on the choice of candidates to become master nodes. In the first one, the best candidates are selected. In the second one, candidates are randomly chosen. The third and the fourth strategies use the number of clusters and the hop average as evaluation procedure. The difference also happens on the choice of sensors to become masters: the best candidates (the third strategy) or a random selection (the fourth strategy). Results show that the maximum flow strategies consumes more CPU power than the others. Moreover, between the third and the fourth strategies, results are quite similar. As a consequence, for future work, would be helpful to observe the way those algorithms would perform in

practice. As well as to apply the maximum flow strategy to obtain clues about the routing traffic.

Since heuristics are used, good solutions are generated even if it does not guaranty optimality. In fact, computing optimal solutions may require a lot of computation time. This would be unsuitable for real-life applications for wireless sensor networks with thousands of sensors.

For future work, we intend to test and to simulate realistic scenarios in order to integrate the proposed algorithms into the LiveNode protocol.

REFERENCES

- [1] K.-M. Hou, J. D. Souza, J.-P. Channet, H.-Y. Zhou, M. Kara, A. Amamra, X. Diao, C. de Vaulx, J.-J. Li, and A. Jacquot, "LiveNode: Limos versatile embedded wireless sensor node," in *Proceedings of 1st International Workshop on Wireless Sensor Networks*, Marrakesh, 2007.
- [2] J. D. Lundquist, D. Cayan, and M. D. Dettinger, "Meteorology and hydrology in yosemite national park: a sensor network application," *Lecture Notes in Computer Science*, vol. 2634, pp. 518–528, 2003.
- [3] A. Cerpa, J. Elson, D. Estrin, L. Girad, and M. Hamilton, "Habitat monitoring: application driver for wireless communication technology," in *Proceedings of ACM SIG-COMN Workshop on Data Communications in Latin America and the Caribbean*, vol. 31, no. 2. San Jose, Costa Rica: ACM, New York, USA, 2001, pp. 3–5.
- [4] S. Meguerdichian and M. Potkonjak, "Low power 0/1 coverage and scheduling techniques in sensor networks," University of California, Los Angeles, Tech. Rep. 030001, 2003.
- [5] X. Li, P. Wang, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 753–763, 2002.
- [6] A. Aguiar, P. R. Pinheiro, A. L. V. Coelho, N. Nepomuceno, A. Neto, and R. P. P. Cunha, "Scalability analysis of a novel integer programming model to deal with energy consumption in heterogeneous wireless sensor networks," *Lecture Notes in Computer Science*, vol. 14, pp. 11–20, 2008.
- [7] M. Demirbas, A. Arora, and V. Mittal, "Floc: A fast local clustering service for wireless sensor networks," in *Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*. Florence, Italy: IEEE Computer Society, Washington, USA, 2004.
- [8] X. Diao, E. Lai, K.-M. Hou, and H. Zhou, "An auto-clustering algorithm for wireless sensor network management protocol," in *Proceedings of 1st International Workshop on Wireless Sensor Networks*, Marrakesh, 2007.
- [9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnam, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Maui, Hawaii: IEEE Computer Society, Los Alamitos, USA, 2000, p. 8020.
- [10] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanisms for wireless sensor networks," in *Proceedings of 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. Washington, USA: IEEE Computer Society, Los Alamitos, USA, 2005, pp. 604–613.
- [11] S. Yi, J. Heo, Y. Cho, and J. Hong, "Peach: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks," *Computer communications*, vol. 30, no. 14–15, pp. 2842–2852, 2007.
- [12] O. Younis and S. Fahmy, "Heed: A hybrid energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [13] H. Hanssanein and J. Luo, "Reliable energy aware routing in wireless sensor networks," in *Proceedings of the 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems*. Columbia, USA: IEEE Computer Society, Washington, USA, 2006, pp. 54–64.
- [14] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.
- [15] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multi-hop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545–557, 2003.
- [16] N. Vlahic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?" in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*. Buffalo, New York: IEEE Computer Society, Los Alamitos, USA, 2006, pp. 258–268.
- [17] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.



Andréa C. Santos was born in Natal in 1975, Brazil. She obtained her PhD in Computer Science at the Catholic University of Rio de Janeiro (PUC-Rio) in 2006, Brazil. She did a traineeship during her PhD at the University of Avignon in 2004, France. She graduated (undergraduate and post-graduated studies) in Computer Science at the Federal University of Rio Grande do Norte State in 2002, Brazil.

She is currently doing a post-doctoral training at the Blaise Pascal University in France, as a member of the *Centre National de la Recherche Scientifique* (CNRS). Her research interests are mainly on exact and heuristic algorithms for combinatorial optimization problems, and on designing mathematical models.



Fatiha Bendali was born in Constantine, Algeria. She received her engineer degree in computer science at the university of Constantine in 1985. She held her PhD in Operations Research at the University of Grenoble, France, in 1990 and a HDR degree in 2003 in Computer Science from the University of Clermont Ferrand.

She is currently an assistant professor at the university Blaise Pascal of Clermont-Ferrand. Her research interests are in Operation research, Combinatorial Optimization (Network Flow and polyedral theory) and game theory.



Jean Mailfert was born in Marseille, France. He received his engineering degree in computer science at the national school of mines in Saint-Etienne (1985). He held his PhD in Operations Research at the Grenoble Institute of Technology, France (1990). Then he worked at the EDF Group R&D department from 1990 to 1993. He is currently an assistant professor at the "Université d'Auvergne" of Clermont-Ferrand. His research interests are in Operations research, Combinatorial Optimization (Network Flow and polyhedral theory) and game theory.



Christophe Duhamel was born in Casablanca in 1971, Morocco. He obtained his PhD in Computer Science at the University of Clermont-Ferrand II in 2001, France. He received *diplôme d'ingénieur* in Mathematics and Modelling at the CUST Clermont-Ferrand in 1994. He got his Master's degree in Computer Science at the University of Clermont-Ferrand II in the same year.

He is currently an assistant professor at the ISIMA (Engineering School in Computer Science) at the University of Clermont-Ferrand II. His research interests are on mathematical modeling, exact and heuristic algorithms for optimization problems in telecommunications and in transportation.



Kean-Mean Hou was born in Cambodia in 1956. He held a PhD degree in 1984 and a HDR degree in 1996 in Computer Science from the University of Technology of Compiègne (UTC). He worked as associate professor at UTC from 1984 to 1986. In 1986 he joined IN2 as R&D engineer group leader. From 1989 to 1996, he created a research group which investigated parallel architecture dedicated to real-time image processing at laboratory HEUDIASYC UMR CNRS (UTC). In 1997 he joined the college of engineering school "ISIMA: Institut Supérieur d'Informatique de Modélisation et de leurs Applications" as professor, where he created the SMIR team of the laboratory LIMOS UMR 6158 CNRS (10 researchers) working on the development of basic hardware and software dedicated to WSN. Different sensor nodes (Bluetooth, WiFi and ZigBee), embedded wireless communication and embedded real-time kernel (SDREAM and LIMOS) are implemented and deployed in different applications such as telemedicine, intelligent transportation system and precision agriculture. He holds 3 EU patents, and he evolved in 3 EU projects and 10 technology transfers. He is also evolved in several scientific committees and boards.