Energy-Aware Constrained Relay Node Deployment for Sustainable Wireless Sensor Networks

Djamel Djenouri[1], Miloud Bagaa[1,2]
1: CERIST Research Center, Algiers, Algeria
2: Aalto University, Finland
Email: ddjenouri@acm.org, miloud.bagaa@aalto.fi

***Abstract*—This paper considers the problem of communication coverage for sustainable data forwarding in wireless sensor networks, where an energy-aware deployment model of relay nodes (RNs) is proposed. The model used in this paper considers constrained placement and is different from the existing one-tiered and two-tiered models. It supposes two different types of sensor nodes to be deployed, i) energy rich nodes (ERNs), and ii) energy limited nodes (ELNs). The aim is thus to use only the ERNs for relaying packets, while ELNs use will be limited to sensing and transmitting their own readings. A minimum number of RNs is added if necessary to help ELNs. This intuitively ensures sustainable coverage and prolongs the network lifetime. The problem is reduced to the traditional problem of minimum weighted connected dominating set (MWCDS) in a vertex weighted graph. It is then solved by taking advantage of the simple form of the weight function, both when deriving exact and approximate solutions. Optimal solution is derived using integer linear programming (ILP), and a heuristic is given for the approximate solution. Upper bounds for the approximation of the heuristic (vs. the optimal solution) and for its runtime are formally derived. The proposed model and solutions are also evaluated by simulation. The proposed model is compared with the one-tiered and two-tiered models when using similar solution to determine RNs positions, i.e., minimum connected dominating set (MCDS) calculation. Results demonstrate the proposed model considerably improves the network life time compared to the one-tiered model, and this by adding a lower number of RNs compared to the two-tiered model. Further, both the heuristic and the ILP for the MWCDS are evaluated and compared with a state-of-the-art algorithm. The results show the proposed heuristic has runtime close to the ILP while clearly reducing the runtime compared to both ILP and existing heuristics. The results also demonstrate scalability of the proposed solution.**

I. Introduction

Energy awareness is the key element that will enable to design sustainable computer systems and networks in the future. We consider in this paper wireless sensor networks (WSNs) as a particular category of computer networks where energy represents the main challenge that should be overcome to achieve sustainability. A wireless sensor network is a set of wireless sensor nodes (SNs) that are capable to sensing the environment, processing the acquired data, and communicate through wireless radios. In this paper, we are interested in the particular problem of relay node (RN) addition for efficient communication coverage in WSN. In many applications, SNs are first deployed in specific areas for sensing coverage. RNs are then deployed to connect the SNs to the base stations (BSs) and/or assure some criteria in the final topology. A typical example is infrastructure monitoring [1], where SNs located in predetermined sensing locations to collect information about the monitored infrastructure and then send the gathered physical information to BSs via multi-hop routes. The precise positions and the number of SNs and BS are known prior to the deployment, and the problem after SNs deployment would be to optimaly place RNs [1].

Two models for RNs placement are used in the literature, one-tiered vs. two-tiered. In the former, any SN can be used as a relay to forward data, whereas in the latter, an SN only sends its own data but cannot be utilised as a relay. Existing solutions may also be categorized into constrained vs. unconstrained models. Unconstrained placement permits to deploy RNs anywhere in the network's area, which is not realistic in many scenarios where placement faces physical constraints that make it possible only in some regions. This is considered in the solutions belonging to constrained placement category. We consider constrained node placement but with a different and more general scenario, where two types of SNs may co-exist, i) energy rich nodes (ERN), and ii) energy limited nodes (ELN). Only ERN can forward data traffic for other nodes, while the ELN use is limited to sensing and sending their own data. A variant of constraint one-tiered model is obtained when the set representing ELNs is empty, while a variant of constrained two-tiered model is reflected when the one representing ERNs is empty. ERNs may be those nodes equipped with– by abstraction– energy unconstrained resources such as large energy storage capacity or more importantly, energy harvesting capabilities. Our contribution is to define this new model for minimum RNs addition in a deployed network and accordingly propose a solution that completely eliminates the use of what we call ELNs for data forwarding, which will thus be ensured by ERNs and the added RNs. Excluding ELNs from data forwarding intuitively improves network efficiency and prolongs the network lifetime compared to the existing one-tiered model, and allowing the use of SNs that are capable of forwarding data will reduce the cost in terms of the number of added RNs, compared to the two-tiered model. While energy capacity is the main motivation for heterogeneity in the proposed model, the latter is general and applies to any other scenarios where nodes' heterogeneity leverages their capacity of forwarding data, e.g., processing power, memory, etc.

The problem is modeled as a variant of the traditional minimum connected dominating set (MCDS), proved to be

NP-hard, reduced to finding the minimum weighted connected dominating set (MWCDS) in a vertex weighted graph. Optimal solution is derived using integer linear programming (ILP), then a heuristic is given for polynomial time resolution with bounded approximation. To reduce time complexity, the simple form of the weight function is exploited both in the ILP and the heuristic. An upper bound of the approximation for the heuristic is derived, as well as for its runtime. Further, the proposed solutions are evaluated by a thorough simulation study where both the heuristic and the ILP are evaluated. To demonstrate the advantage of the proposed model over the one-tiered and the two-tiered models, we compare the proposed solutions with, i) the use of traditional MCDS calculation and replacement of all the obtained set by RNs, which is equivalent to use a MCDS calculation based solution in the two-tiered model, and ii) the use of the same method then using the SNs in the set as relays without replacement, which is equivalent to use an MCDS calculation based solution in the one-tiered model. A trivial energy-aware solution is also used in the comparison, which minimizes the use of ELNs by calculating the shortest paths on the node weighted graph without any RN addition. This gives a clear view on the gain that might be obtained from the proposed model and solutions, as well as the related cost. But to be fair, we do not claim direct comparison with any particular solution from the literature given the difference in the model. The proposed heuristic for MWCDS calculation has also been evaluated and compared with a state-of-the-art algorithm [2].

The remainder of the paper is organized as follow: The related work is presented in Sec. II, then the network model in Sec. III. The problem is formulated and solved in Sec. IV using ILP and a heuristic. Upper bounds are derived for the approximation and the runtime of the proposed heuristic in Sec. V. A thorough numerical analysis by a simulation study is presented in Sec. VI, and finally Sec. VII concludes the paper.

II. Related Work

Additional RNs placement in deployed WSN has largely been treated in the literature. Existing solutions may be divided into two models, one-tiered vs. two-tiered [3]. Solutions of [4], [5], and [6] are examples of those using the one-tiered model. The connectivity problem in WSN was addressed in [5], where the authors targeted the deployment of a minimum number of RNs such that each SN is connected to a BS. The authors aimed to find a tradeoff between the network lifetime and cost. The problem was modelled by a Steiner tree with a minimum number of Steiner points and bounded edge-length. The authors of [4] considered the RN problem such that a survivability requirement is achieved. The aim was to determine the location of the minimum number of RNs such that each SN is connected to a BS through several node-disjoint paths, which provides fault tolerance in case of node failure. Most works on one-tiered model, including [5], [4] considered the unconstrained deployment where RNs can be deployed anywhere, which is unrealistic in most cases. In [6], the authors considered the problem of constrained RN placement where the RNs can only be placed in a set of candidate positions, which is more realistic. They addressed the connectivity and the survivability problems and propose approximation algorithms with a proved polynomial time complexity.

The limitation of the one-tiered model is the possible fast exhaustion of sensors' batteries due to the use of any SN for data forwarding, regardless of its energy potantiels/capacity. This may structurally damage the network and cause a network partition. To address this problem, many solutions have been proposed in the two-tiered model, such as [7], [8], [3], [9], [10], [11], [12], [13], [14]. The authors of [9] considered both the one-tiered and the two-tiered models and provided an approximation algorithm for the one-tier model and a polynomial time approximation scheme for the two-tier one. The survivability problem was investigated in [10], where the aim was to ensure that each SN is connected to the BS through two disjoint paths. The connectivity problem was addressed in [11], where the RNs were assumed to have a wider communication range than the SNs. In [3], both the connectivity and the survivability requirements have been considered. In their experimental results, the authors showed that the number of added RNs does not exceed twice the number in the optimal solution. Chelli et al. [14] considered constrained node placement in two-tiered model and proposed a one-step algorithm to construct a connected Steiner-tree topology.

Compared to the one-tiered model, the two-tiered one has the problem of higher cost in terms of the number of RNs to add. It does not consider and take advantage of possible homogeneity of SNs, particularly in terms of energy capacity. For example, it is possible to have some SNs endowed with energy harvesting capacity that allows them to forward traffic for others [15] [16]. Neither of the previous models allow to mimic this feature. While energy harvesting has been recently considered by the research community and several problems in energy harvesting WSN have been revisited such as stochastic harvesting modeling [17], [18], [19], and [20], duty cycle management [21], [22], and clustering [23], few has been done for RN placement. To our knowledge, the only works in this context are [24], [25]. In the latter, the authors considered the scenario where possible potential locations for RNs are limited and known a priori, and every location was supposed to have a constant energy harvesting potential. The aim of the authors was to deploy a minimum number of harvesting enabled RNs in the network while increasing the harvesting capability. The authors used the one-tiered model, where all SNs were supposed not to be harvesting-enabled but they have been used to forward packets similarly to the added RNs. So the consideration of energy harvesting capability does not eliminate the problem of one-tiered model used by the solution. We define a new model in this paper that allows to reflect nodes' heterogeneity and thus capacity to forward data. The proposed model is general and the context of energy harvesting WSN is just an example of application [24].

III. Network Model and Preliminaries

A. Network Model and Problem Statement

The network is represented by an undirected unit disk graph, $G = (V, H, E)$, where the set of vertices, V, represents the nodes, and E the set of edges, $(u, v) \in E$ iff both u and v are in the communication range of one another. The graph G is supposed to be initially connected. The set V is composed of two subsets, the set of ERNs, denoted by H, and the one of ELNs, $V \setminus H$. V may include one or several BSs that may or may not forward packets. This depends on the application scenario, although the BSs are generally energy unconstrained nodes. The proposed solution works in both cases. For abstraction, if the BSs can be used to forward packets, they are to be included in H. The proposed solution is general and applies to any energy model. The only requirement is that a node in H should always be able to forward traffic when requested. This can be ensured when having high energy sources at those nodes, or when they are endowed with energy harvesting capability coupled with the use of some energy storage unites (capacitors, rechargeable batteries, etc.), and/or by shaping the network traffic model according to the availability of energy, etc. For instance, in time varying energy harvesting models (e.g., using solar energy), the heavy load may be performed when the energy source is available (during day light), and in the absence of energy the SNs activity may be reduced to a certain degree depending on their energy storage capacity. Dealing with issues such as SN scheduling and traffic control is out of the scope of this paper. Nodes in H are then regarded as energy unconstrained nodes. With the previous assumption, it is trivially power efficient to use only SNs from H for packet relaying to the BS(s). Therefore, the problem is to ensure data forwarding only through nodes from H, with addition of a limited number of RNs to ensure communication coverage (connectivity). Sensing is performed by all the SNs (both ERNs and ELNs), and its coverage is supposed to be assured a priori in the initial deployment. Dealing with sensing coverage is another problem and it is out of the scope of this work. Constrained RNs placement is considered in this paper, and potential positions of RNs are limited to areas near SNs position in the initial deployment. This is more realistic than unconstrained deployment in many situations, such as the existence of obstacles, unaccessible areas within the deployment regions, etc. It is also reasonable to assume feasibility of deploying RNs where SNs have been deployed. Accuracy of all the formulations and analysis presented hereafter relies on this model and its assumptions.

B. Definitions and Notations

Concepts used in the problem formulation and resolution are defined bellow.

- A connected dominating set (CDS) for a graph, $G = (V, E)$, is a sequence of vertices, $S \in V$, that fulfils: i) $\forall u \in V \setminus S, \exists v \in S, (u, v) \in E$, ii) the subgraph induced by S is connected. The set of all sequences that satisfy the previous conditions is denoted by $\mathcal{CDS}(G)$. A set that fulfils condition (i) but not necessarily condition (ii) is called dominating set (DS).
- A minimum connected dominating set (MCDS) for a graph, $G = (V, E)$, is a connected dominating set with a minimum number of vertices, i.e. $S = \underset{\xi \in \mathcal{CDS}(G)}{\arg\min} |\xi|$. The set of all sequences that satisfy this condition is denoted by $\mathcal{MCDS}(G)$. A dominating set with a minimum number of vertices but which is not necessarily connected, i.e., $S = \underset{\xi \in \mathcal{DS}(G)}{\arg\min} |\xi|$, is called minimum dominating set (MDS).
- A minimum weighted connected dominating set (MWCDS) for a vertex weighted graph $G = (V, E, W)$– where W is a function that assigns a weight to every vertex in V– is a connected dominating set with a minimum cumulative weight; $S = \underset{\zeta \in \mathcal{CDS}(G)}{\arg\min} \sum_{u \in \zeta} W(u)$. The set of all sequences that satisfy this condition is denoted by $\mathcal{MWCDS}(G)$. A dominating set with a minimum cumulative weight but which is not necessarily connected, i.e., $S = \underset{\zeta \in \mathcal{DS}(G)}{\arg\min} \sum_{u \in \zeta} W(u)$, is called minimum weighted dominating set (MWDS).
- The k-neighborhood of a vertex v, denoted by $\mathcal{N}_k(v)$, is the set of vertices that are at most $k - hop$ from v, i.e., the shortest path separating them to v is no more than k hops. In particular, $\mathcal{N}_0(v) = \{v\}$, and $\mathcal{N}_1(v)$ stands for direct neighbors.
- The maximum independent set of a graph G, denoted $\mathcal{MIS}(G)$, is the maximum set of vertices that are unconnected. Likewise, we define the MIS in the $k-hope$ from a cental vertex, v, say $MIS_k(v)$, as the maximum set of vertices that are unconnected in the adjacent subgraph defined by the k-neighborhood of a vertex v.
- The dominating weight of vertex, v, say $\tilde{\omega}(v)$, is the minimum weight of vertex v's neighbors, i.e. $\tilde{\omega}(v) = min\{\omega(u), u \in \mathcal{N}_1(v)\}$.
- A multi-graph (or pseudograph) is a graph that may include multiple edges (parallel edges) between every pair of vertices, i.e., two vertices may be connected by more than one edge.
- Clustering in a graph: The process of clustering in a graph is to replace every subset of vertices (following some criterion) by a single vertex, called a cluster, and replacing accordingly the appropriate edges as well. The process is iterative until all the vertices are replaced and a reduced graph is obtained (called clustered graph). The term cluster used in the following of the paper is related to this concept.

IV. Problem Formulation and Resolution

A. Problem Formulation

The problem described in Sec. III can be solved in the model represented by, $G = (V, H, E)$, by finding a CDS in the graph G ($S \in \mathcal{CDS}(G)$), with a minimum number of vertices from $V \setminus H$. Let us denote this problem by $P1$. Once $P1$ is solved, RNs will be added at the positions co-located with ELNs of S (if any). This is to substitute them in the task of packet forwarding.

Theorem 1: The problem $P1$ is NP-hard.

We refer to the Appendix for the proof of this theorem. To our knowledge, the problem $P1$ has not been treated in the operational research and graph theory literature. However, since it is very similar to the traditional MCDSP that has been largely treated, we propose a transformation to some existing variant of MCDSP. This is for the purpose of taking advantage of the existing heuristics from the literature, such as [2], [26]. These heuristics are particularly appropriate to unit disk graphs , for which they achieve better approximation compared to solutions for general graphs.

The graph, $G = (V, H, E)$, of $P1$ is transformed into a vertex weighted graph, G_w, using the following weight function:

$$W : V \to \{0, 1\}$$
$$\forall u \in H, W(u) = 0; \forall v \in \{V \setminus H\}, W(v) = 1. \qquad (1)$$

The problem $P1$ then reduces to the search for a *minimum-weight* CDS ($S \in MWCDS(G_w)$). In the following, the new problem is denoted by $P2$.

Theorem 2: If S is a MWCDS for $P2$ (i.e, S $\in$ $MWCDS(G_w)$), then it is an optimal solution for $P1$.

The Appendix is referred for the proof of this theorem. From this theorem, we realize that solving $P1$ is equivalent to solving $P2$.

Theorem 3: Let the problem P_H be defined for $G(V, H, E)$ as follows: Find S such that,

$$S = \underset{\xi \cup H \in \mathcal{CDS}(G)}{\arg\min} |\xi|.$$

If S is a solution to P_H, then $S \cup H$ is a solution to $P1$.

We refer to the Appendix for the proof of this theorem. It Results from Theorem 3 that solving $P1$ (or $P2$) is equivalent to adding a minimal set, say S, to H, to construct a solution $S \cup H \in \mathcal{MCDS}(G)$ (respectively $S \cup H \in MWCDS(G_w)$). This feature will be useful in the following to reduce the search space, both in the optimal solution and the heuristic.

B. Problem Resolution

1) General Framework: The proposed general framework is illustrated by Algorithm 1. This algorithm has as input (line 1), i) the communication graph, $G = (V, H, E)$, which includes the set of vertices (V), the subset of vertices representing the ERNs ($H \subset V$), and the set of edges (E), ii) $\mathcal{F}$, a function that calculates the MWCDS. The output of the algorithm (line 2) is the set of positions where the RNs should be placed to ensure connectivity. The algorithm starts by initiating the sets W, and S_p to empty set (line 3). The vertices' weights are then calculated by applying the formula given in Eq. (1) (line 4) and inserted to the set W. The resulted weighted graph, $G_w = (V, E, W)$, is passed as input to the function, $\mathcal{F}$, in line 5, which produces the MWCDS (or an approximation of it), χ. This MWCDS resolution will be developed later. The ELNs from χ are denoted ξ, whose positions represent the output of the algorithm. The RNs are then to be put in the positions col-located with these ones to replace the appropriate SNs in forwarding packets, while ELNs will be used only to collect and transmit their own data. With the addition of such RNs, the proposed solution ensures that the network can be connected only though ERNs plus the new RNs.

Algorithm 1: General Solution Framework

1 **Input:** $G = (V, H, E)$, Function $\mathcal{F}$
2 **Output:** The set of positions, S_p, where to put the RNs.
3 **Init:** $W = S_P = \emptyset$
4 Assign weights to vertices (construct *W*) using Eq. (1):
5 Run $\mathcal{F}(V, E, H, W)$ to get a MWCDS, say χ.
6 $\xi = \chi \bigcap (V \setminus H)$.
7 $\forall u \in \xi$ add the position of u to S_P.
8 return S_P

2) Optimal Resolution with an Integer Linear Program: To solve $P1$, we use P_H that helps reducing the number of variables and thus the search space compared to $P2$. The problem, P_H, can be modeled by the following mixed integer linear program:

$$\min \sum_{i \in V \setminus H} X_i, \qquad (2)$$

s.t,

$$X_i + \sum_{j \in \mathcal{N}(i)} X_j \geq 1, \forall i \in V \qquad (3)$$

$$\sum_{i \in \mathcal{N}(1)} F_{1,i} = \sum_{i \in V, i \neq 1} X_i \qquad (4)$$

$$\sum_{j \in \mathcal{N}(i)} F_{j,i} - \sum_{j \in \mathcal{N}(i)} F_{i,j} = X_i, \forall i \in V, i \neq 1 \qquad (5)$$

$$0 \leq F_{i,j} \leq n X_j, \forall (i, j) \in E, j \neq 1 \qquad (6)$$

$$\sum_{i \in N(1)} Y_i \leq 1 + X_1(|N(1)| - 1) \qquad (7)$$

$$F_{1,i} \leq n Y_i, \forall i \in N(1) \qquad (8)$$

$$F_{i,j} = 0, \forall (i, j) \notin E, \text{ or } j = 1 \qquad (9)$$

$$X_i = 1, \forall i \in H. \qquad (10)$$

The ILP has as input, i) the graph $G = (V, H, E)$, and, ii) a set $N(i)$ for every vertex, v_i, i.e., the set of adjacent vertices (neighboring nodes). The outputs are: i) a vector of booleans, X, which represents the decision variables, i.e., $X_i = 1$ iff vertex, $v_i \in V$, is selected in the MCDS. Variables in X are only for $v_i \in V \setminus H$, while entries for $v_i \in H$ are fixed a priori to 1 (Eq. (10)). ii) The flow matrix of integers, $F_{i,j}$, $(v_i, v_j) \in E$, as well as iii) the vector Y_i, for $v_i \in N(1)$, which are additive variables used to model the connectivity as it will be explained hereafter.

The objective function, Eq. (2), is to minimize the total weight of vertices in the selected CDS, to achieve MWCDS. The constraint represented by Eq. (3) is to guarantee that either v_i is in the CDS ($X_i = 1$), or it has an edge towards some vertex in the CDS (at least one of the terms X_j should equal 1). Constraints represented by Eq. (4) throughout Eq. (8) are for medeling the connectivity requirements. The principle is to generate a flow, only from an arbitrary vertex v_1. The amount of this flow is the exact amount to cover the CDS (Eq. (4)), i.e., it should be $\sum X_i$ if v_1 is out of the set ($X_1 = 0$), or $\sum X_i - 1$ if it belongs to the CDS (one of the dominating vertices). In the former case, v_1 inevitably would have at least one edge towards a dominating vertex. The generated fellow traverse the dominating vertices and at every one, a single unit of the flow fades (Eq. (5)). Eq. (6) verifies that every flow is bounded by 0 and n, and that no flow goes to the dominated vertices. This is as the term $F_{i,j}$ vanishes when $X_i = 0$. Note here that a more strict upper bound that would reduce the search space is $\sum Xi$ instead of n, but this would make the inequalities non-linear. Also note that the latter condition, combined with Eq. (4) when $X_i = 0$, ensures no flow will be generated from vertices out of the CDS.

Constraints represented by Eq. (7) and Eq. (8) are used to limit the number of neighbors to which node v_1 can transfer its flow. A binary vector, Y, is added to the outputs such that $Y_i = 1$ iff flow is permitted from node, v_1, to node, v_i. Constraint of Eq. (8) ensures that flow can only be transferred from node v_1 to v_j if $Y_j = 1$, while Eq. (7) forces Y_j to be set to 1 for only one neighboring node, v_j, in case, $v_1 \notin CDS$. Otherwise, it is bounded by the number of v_1's neighbors. Finally, conditions expressed by Eq. (9) are to ensure the flow travels only through existing edges, and no flow enters v_1, and Eq. (10) to set X entries to 1 for nodes in H (constants). Note that the latter conditions (Eq. (9) and Eq 10) are just to reduce the number of the ILP variables, and they do not represent constraints to be verified by the ILP solver.

3) Heuristic: The previous ILP represents the optimal resolution for the function $\mathcal{F}$. It can only be used in limited scenarios. Given that the problem is NP-hard, a polynomial-time heuristic is needed as a general and scalable solution. In the following, we propose a heuristic for the function $\mathcal{F}$ based on [2] and [26] while considering the particularity of the problem, as illustrated by Algorithm 2. The principal of [2] is used to find an approximation of the minimum weighted dominating set (MWDS), say χ_1. This starts by initializing χ_1 to $\emptyset$ (line 3), which is progressively augmented by adding local dominating vertices, D_{k+2}, of $k+2-neighborhood$, from a pivotal vertex v (the loop from line 4 to line 20). If there is a vertex with all direct neighbors in $V \setminus H$, then it is selected as v, otherwise v is chosen arbitrary (lines 5 throughout line 9). The inner loop (repeat loop) searches for local dominating set for the $k+2$ vicinity from v ($\mathcal{N}_{k+2}(v)$), and it increases k until the condition in line 17 is fulfilled. The condition states that the weight of the local dominating set for $\mathcal{N}_{k+2}$ is no more than the one for $\mathcal{N}_k$ when multiplied by the factor $(1+\epsilon)$. It is actually this condition that ensures $1+\epsilon$ approximation of the MWDS. The result of Theorem 3 is used to accelerate the search, and all the vertices which have null weight in the appropriate vicinity are initially picked up in every iteration (line 13 for k neighborhood, and line 15 for $k+2$ neighborhood) instead of performing an exhaustive search among all k neighbors and $k+2$ neighbors. Those with non-null weight ($\mathcal{N}_k(v) \cap (V \setminus H)$, and ($\mathcal{N}_{k+2}(v) \cap (V \setminus H)$) are added progressively if necessarily to construct the local MWDS with a minimum addition. The local MWDS is added to χ_1, and the loop continues by selecting another pivotal vertex until covering the set V. At that point (termination of the $while$ loop), χ_1 will represent an approximation of MWDS, but whose elements are not necessary connected.

The second part of the algorithm (from line 21 to line 26) is similar to the solution used in [26] to connect the resulted approximation of the MWDS and construct an approximation of the MWCDS. Although the solution of [26] uses a different approach to calculate a MWDS, the connection algorithm proposed by the authors is general and connects any DS. To calculate the connector set, λ, the connected parts in χ_1 are determined, then every connected part is clustered in c_i. After clustering, an auxiliary multiple graph, $\hat{G} = (\hat{V}, \hat{E}, \hat{W})$, is constructed from, G, as follows: i) The vertices are the clusters (c_i). ii) Between every couple of vertices, $(c_i, c_j) \in \hat{V}^2$, and for every path, $p \in G$, of length not exceeding three hops, an edge between c_i and c_j is added. The resulted graph is a multiple graph with possible multiple edges between a couple of vertices. Note that every path, p, includes only vertices that do not belong to χ_1, and which will be possibly used to connect vertices of χ_1. A minimum spanning tree of $\hat{G}$ is then calculated. λ is then constructed from vertices in G that form every single edge in the calculated spanning tree. χ_1 is augmented with λ to construct the connected χ.

V. Analysis and Discussions

A. Approximation

An upper bound for the approximation of the heuristic (vs. the optimum) is derived in the following. As the heuristic is composed of two parts, the approximation of each one is first given, i.e., i) finding a MWDS (χ_1), and , ii) connecting it (adding λ). The approximation of the whole algorithm will then be deduced.

Theorem 4: The set, χ_1, computed by the heuristic, $\mathcal{F}$, satisfies: $W(\chi) \leq (1+\epsilon)W(\chi_{op})$, where $W(\chi_{op})$ is the weight of the $\mathcal{MWCDS}(G)$.

Proof: The principle of the first part of the algorithm (line 3 throughout 20) is to build an approximation of the MWDS, χ_1, as the union of the partial MWDS of mutually node-disjoint sets, for the sets $\mathcal{N}_{k_1+2}, \mathcal{N}_{k_2+2}, \cdots, \mathcal{N}_{k_r+2}$, say $\bigcup_{i=1}^{r} D_{ki+2}$, where k_i is the final value of k in the iteration i, and r is the total number iterations. These partial MWDS are obviously node-disjoint. From the stop condition at every iteration of the inner loop (line 17), at the exit of the while loop (line 20) the following is fulfilled:

Algorithm 2: Algorithm describing a Heuristic for $\mathcal{F}$

1: **Input:** $G = (V, H, E, W)$
2: **Output:** A connected dominating set, χ, which is an approximation of the MWCDS.
3: **Init:** $\chi = \chi_1 = \lambda = \emptyset$
Look up for a Dominating Set χ_1
4: **while** $V \neq \emptyset$ **do**
5: **if** $\exists u \in V, \tilde{\omega}(u) = 1$ **then**
6: $v = u$
7: **else**
8: Chose arbitrary vertex v
9: **end if**
10: $k = -1$
11: **repeat**
12: $k = k + 1$
13: $D_k = \mathcal{N}_k(v) \cap (H)$
14: add progressively to D_k a minimum number of non-null weight vertices ($u \in \mathcal{N}_k(v) \cap (V \setminus H)$); until D_k dominates $\mathcal{N}_k(v)$
15: $D_{k+2} = \mathcal{N}_{k+2}(v) \cap (H)$
16: add progressively to D_{k+2} a minimum number of non-null weight vertices ($u \in \mathcal{N}_{k+2}(v) \cap (V \setminus H)$); until D_{k+2} dominates $\mathcal{N}_{k+2}(v)$
17: **until** $\sum_{u \in D_{k+2}} W(u) \leq (1+\epsilon) \sum_{u \in D_k} W(u)$
18: $\chi_1 = \chi_1 \cup D_{k+2}$
19: $V = V \setminus \mathcal{N}_{k+2}(v)$
20: **end while**
Connect the set χ_1
21: Determine the connected components in χ_1, cluster them and denote every cluster c_i
22: Construct an auxiliary graph, $\hat{G} = \{\hat{V}, \hat{E}, \hat{W}\}$ from G as follows:
23: $\hat{V} = \{c_i\}$
24: For every path, $p \in G$, of length 3 or less that connects a vertex from c_i to another one from c_j, add an edge, e, to $\hat{E}$, and set $\hat{W}(e) = W(p)$
25: Compute a minimum spanning tree MST of $\hat{G}$
26: For every edge $e \in MST$, add the vertices that form e to λ
27: $\chi = \chi_1 \cup \lambda$

$$\sum_{u \in D_{k_i+2}} W(u) \leq (1+\epsilon) \sum_{u \in D_{k_i}} W(u), \forall i \in \{1, \cdots, r\}. \quad (11)$$

The summation of all inequalities defined by 11 yields,

$$W(\chi_1) = \sum_{i=1}^{r} \sum_{u \in D_{k_i+2}} W(u) \leq (1+\epsilon) \sum_{i=1}^{r} \sum_{u \in D_{k_i}} W(u). \quad (12)$$

Let $\widetilde{\chi_1}$ be an optimal solution for the minimum weighted dominating set problem ($\widetilde{\chi_1} \in \mathcal{MWDS}(G)$). On the one hand, $\forall_i \in \{1, \cdots, r\}$, $\widetilde{\chi_1} \cap \mathcal{N}_{k_i+1}$ dominates $\mathcal{N}_{k_i}$. On the other hand, $\forall_i \in \{1, \cdots, r\}$, D_{k_i} is the optimal dominating set in $\mathcal{N}_{k_i}$, i.e., its weight does not exceed that of any other dominating set (including $\widetilde{\chi_1} \cap \mathcal{N}_{k_i+1}$). Consequently,

$$\sum_{i=1}^{r} \sum_{u \in D_{k_i}} W(u) \leq \sum_{i=1}^{r} \sum_{u \in (\widetilde{\chi_1} \cap \mathcal{N}_{k_i+1})} W(u). \quad (13)$$

Since $\widetilde{\chi_1} = \bigcup_{i=1}^{r} (\widetilde{\chi_1} \cap \mathcal{N}_{k_i+2})$, $W(\widetilde{\chi_1}) = \sum_{i=1}^{r} \sum_{u \in (\widetilde{\chi_1} \cap \mathcal{N}_{k_i+2})} W(u)$. Therefore,

$$\sum_{i=1}^{r} \sum_{u \in (\widetilde{\chi_1} \cap \mathcal{N}_{k_i+1})} W(u) \leq W(\widetilde{\chi_1}). \quad (14)$$

From Eq. (12), Eq. (13), Eq. (14), it results,

$$W(\chi_1) \leq (1+\epsilon) W(\widetilde{\chi_1}). \quad (15)$$

$\widetilde{\chi_1} \leq W(\chi_{op})$, as χ_{op} should be connected in addition to being a MWDS. Consequently, Eq. (15) yields, $W(\chi_1) \leq (1+\epsilon) W(\chi_{op})$. $\square$

Theorem 5: $W(\lambda) \leq 4W(\chi_{op})$, where λ is the set of vertices added to connect χ_1 in the second part of the algorithm (from line 21).

We refer to [26] for the proof of this theorem, where a similar solution is used to connect the MWDS (but a different function is used for calculating the dominating set). Note that the 4 approximation of that solution has been proven for any dominating set as input.

From Theorem 4 and Theorem 5, the following corollary is obtained:

Corollary 1: $W(\chi) \leq (5+\epsilon) W(\chi_{op})$, i.e., the proposed heuristic provides a MWCDS with $(5+\epsilon)$ approximation in the worst case.

B. Runtime

In the following, upper bounds for the number of steps required to run the different parts of the proposed heuristic are derived. This serves just for the worst case analysis in the most pessimistic scenarios by ignoring all factors that may help to reduce the runtime, e.g., the number of ERNs, etc. Averaged and thorough analysis will be investigated by simulation in the next section.

1) Calculation of χ_1 (while loop): The most time consuming step in this loop is the exhaustive search for the local dominating sets (repeat loop), which trivially increases with k. Now we derive an upper bound for k, say k_{max}, which represents the upper bound of the number of iterations in the repeat loop. Within the loop, the following condition holds,

$$(1+\epsilon) \sum_{u \in D_k} W(u) < \sum_{u \in D_{k+2}} W(u), \forall k \leq k_{max}. \quad (16)$$

This yields,

$$(1+\epsilon)\tilde{\omega}(v) < \sum_{u \in D_2} W(u),$$

$$(1+\epsilon) \sum_{u \in D_2} W(u) < \sum_{u \in D_4} W(u),$$

$$\vdots$$

$$(1+\epsilon) \sum_{u \in D_k} W(u) < \sum_{u \in D_{k+2}} W(u),$$

where v is the vertex selected at the beginning of the while loop.

By transitivity we get, $(1+\epsilon)^{\frac{k+2}{2}}\tilde{\omega}(v) < \sum_{u\in D_{k+2}} W(u)$, i.e., $(1+\epsilon)(\sqrt{(1+\epsilon)})^k\tilde{\omega}(v) < \sum_{u\in D_{k+2}} W(u)$. Consequently,

$$(\sqrt{(1+\epsilon)})^k\tilde{\omega}(v) < \sum_{u\in D_{k+2}} W(u). \quad (17)$$

Given the weight function that we used (zero/one weights), the weight of D_{k+2} can be bounded by:

$$\sum_{u\in D_{k+2}} W(u) \leq |MIS_{k+2}(v)|. \quad (18)$$

From Eq. (17) and Eq. (18), and given that $\tilde{\omega}(v)$ [1] is either 1 or 0, the following is obtained,

$$(\sqrt{(1+\epsilon)})^k < |MIS_{k+2}(v)|. \quad (19)$$

According to [27], $|MIS_k|$ can be bounded by, $(2k+1)^2$ (in unit disk graphs). Using this in Eq. (19), it yields,

$$(\sqrt{(1+\epsilon)})^k < (2k+5)^2. \quad (20)$$

As both sides of the pervious inequality are positive, it is equivalent to,

$$(\sqrt[4]{(1+\epsilon)})^k < (2k+5). \quad (21)$$

Let us define $f(k)$ by, $f(k) = (2k+5) - (\sqrt[4]{(1+\epsilon)})^k$. Satisfying the inequality defined by Eq. (21) is equivalent to determine the interval of values of k, for which $f(k) \geq 0$. k_{max} is thus the maximum value that satisfies such a condition, which can be determined by studying the variation of $f(k)$. The derivative of $f(k)$ vanishes at a single point, say k_0, $k_0 = 4\log_{1+\epsilon}(\frac{8}{\ln(1+\epsilon)})$, where $\log_{1+\epsilon}$ denotes the logarithm of base $1+\epsilon$. From k_0, the derivative of $f(k)$ becomes negative. The two limits of $f(k)$ when $k \to +\infty$ (res. when $k \to -\infty$) are $-\infty$. $f(k)$ is then positive in a single and limited interval including k_0, where $f(k_0)$ is the pick value of $f(k)$. Consequently, we conclude that the maximum value of k (in which $f(k)$ is positive) is the one that makes it vanish during its drop at the interval $[k_0, +\infty[$. That is, k_{max} satisfies,

$$(2k+5) = (\sqrt[4]{(1+\epsilon)})^k,$$

s.t.,

$$k > 4\log_{1+\epsilon}(\frac{8}{\ln(1+\epsilon)}). \quad (22)$$

The resolution of the previous equation gives, K_{max}, a very relaxed upper bound that k may reach in any iteration of the loop. Note that it only depends on ϵ, and not the problem size ($|V|$).

Giving the $(0,1)$ weight function, every search stops at the first DS found and there is no need to continue the exploration (contrary to general weight heuristics). Therefore, the worst case is when the dominating set for every $N_k(v)$, say D_k, is $MIS_k(v)$ (its upper bound). Still in the worst case, the latter is expressed by its upper bound given in [27],

$$|MIS_k(v)| = (2k+1)^2. \quad (23)$$

Every iteration of the repeat loop consists in an exhaustive search for the MDS for vertices i) $N_k(v) \setminus N_{k-1}(v)$, and ii) $N_{k+2}(v) \setminus N_{k+1}(v)$ that are not dominated by the current calculated MDS. This is as the MDS for N_{k-1}, and N_{k+1} have already been calculated in the previous step, which will be extended with the obtained partial MDS. The search for the MDS consists in looking up for the DS with one vertex (using all possible combinations). Otherwise, with two vertices (using all possible combinations), etc. The number of steps are then respectively given by $S_1 = \binom{N_1}{1} + \binom{N_1}{2} + \cdots \binom{N_1}{l_1}$, $S_2 = \binom{N_2}{1} + \binom{N_2}{2} + \cdots \binom{N_2}{l_2}$, where $N_1 = |N_k(v) \setminus N_{k-1}(v)|$, $N_2 = |N_{k+2}(v) \setminus N_{k+1}(v)|$, l_1 (resp. l_2), represents the size of the first MDS found for N_1 (resp. N_2). In the worst case, l_1, resp. l_2, is the difference between the maximum independent sets, $|MIS_k(v)| - |MIS_{k-1}(v)|$, resp. $|MIS_{k+2}(v)| - |MIS_{k+1}(v)|$, which represent trivial limits to the partial DS.

Eq. (23) yields $l_1 = 8k$, and $l_2 = 8(k+2)$.

Let us denote the degree of G by Δ. The maximum number of vertices in $N_k(v)$ is no more than $(\Delta+1)|MIS_k(v)|$. Using the bound of the binomial sum, $S_1 \leq l_1\binom{N_1}{l_1}$, as well as the bound of l_1, we obtain, $S_1 \leq 8k\frac{(8k(\Delta+1))!}{(8k)!(8k\Delta)!}$. This yields,

$$S_1 \leq \frac{\prod_{i=1}^{8k}(8k\Delta+i)}{(8k-1)!}. \quad (24)$$

Similarly, the following bound for S_2 may be derived,

$$S_2 \leq \frac{\prod_{i=1}^{8(k+2)}(8(k+2)\Delta+i)}{(8k+15)!}. \quad (25)$$

The total cost of every iteration of the while loop in the worst case is then bounded by,

$\sum_{k=1}^{k_{max}} \frac{\prod_{i=1}^{8k}(8k\Delta+i)}{(8k-1)!} + \frac{\prod_{i=1}^{8(k+2)}(8(k+2)\Delta+i)}{(8k+15)!}$, which is bounded by:

$$k_{max}\left(\frac{\prod_{i=1}^{8k_{max}}(8k_{max}\Delta+i)}{(8k_{max}-1)!} + \frac{\prod_{i=1}^{8(k_{max}+2)}(8(k_{max}+2)\Delta+i)}{(8k_{max}+15)!}\right). \quad (26)$$

The number of iterations is trivially $|V|/|N_{k_{max}+2}|$, which is bounded by $|V|/((\Delta+1)(2k_{max}+5)^2)$. The final bound of the number of steps is then obtained by multiplying this number by the bound in Eq. (26), which is,

[1] remember that $\tilde{\omega}(v)$ denotes the dominating weight of vertex, v, i.e., the minimum weight of vertex v's neighbors

$$\frac{k_{max}|V|}{(\Delta+1)(2k_{max}+5)^2}\times$$

$$\left(\frac{\prod_{i=1}^{8k_{max}}(8k_{max}\Delta+i)}{(8k_{max}-1)!}+\frac{\prod_{i=1}^{8(k_{max}+2)}(8(k_{max}+2)\Delta+i)}{(8k_{max}+15)!}\right). \tag{27}$$

Given that the variable terms in Eq. (27) are V and Δ, and the right side of the product is dominated by the second term (i.e., $\frac{\prod_{i=1}^{8(k_{max}+2)}(8(k_{max}+2)\Delta+i)}{(8k_{max}+15)!}$), the asymptotic order is:

$$\mathcal{O}\left(\frac{|V|\Delta^{8k_{max}+15}}{(8k_{max}+15)!}\right), \tag{28}$$

which represents a very relaxed asymptotic polynomial bound related to the varialbes $|V|$ and Δ (note that k_{max} is a constant).

2) Calculation of λ (Connecting χ): In the worst case, χ is composed of totally unconnected components, which results in a maximum number of clusters (i.e., $|\chi| = |\hat{V}|$). This is the case when the resulted dominating set is an independent set, which is bounded by $(2D+1)^2$, where D denotes the network diameter (the longest path between any two vertices in G). The calculation of the connected components (line 21) can be performed in $O(|E|)$. The construction of an auxiliary graph using paths of length not exceeding three hops (one or two intermediate nodes) can be achieved by exploring all the one-hop and two-hop edges from every c_i. This needs $(\Delta^2+\Delta^3)|\hat{V}|$ steps, i.e., less than $(\Delta^2+\Delta^3)(2D+1)^2$.

The computation of the minimum spanning tree in graphs with integer weights can be achieved in $O(|\hat{E}|)$. More Precisely, in less than $4|\hat{E}|$ [28]. $|\hat{E}|$ can be bounded by, $(\Delta+\Delta^2)|\hat{V}|^2/2$, which is bounded by, $(\Delta+\Delta^2)(2D+1)^4/2$. That is, $4|\hat{E}|$ is bounded by, $2(\Delta+\Delta^2)(2D+1)^4$. Finally, the task represented in line 26 can be performed in $|\hat{V}|$ steps (the number of edges in the spanning tree), i.e., less than $(2D+1)^2$. By summing all the previous bounds, the number of total steps required to calculate λ is then bounded by,

$$|E|+(2(\Delta+\Delta^2)(2D+1)^2+\Delta^2+\Delta^3+1)(2D+1)^2. \tag{29}$$

The asymptotic order of this equation is

$$\mathcal{O}(|E|+\Delta^2D^4+\Delta^3D^2), \tag{30}$$

which represents a general term of a relaxed polynomial upper bound with three variables (Δ, D, and $|E|$). As particular cases, i) in dense networks, the third term of the sum in Eq. (30) dominates the second one, which yields $\mathcal{O}(|E|+\Delta^3D^2)$, whereas ii) in scattered networks, the second one dominates the third and the first ones, which yields $\mathcal{O}(\Delta^2D^4)$.

C. Application and Possible Extensions

While energy capacity is the main motivation for heterogeneity in the proposed model, the proposed model is general and applies to any scenarios where nodes' heterogeneity leverages their capacity of forwarding data, e.g., processing power, memory, etc. Typical example of environments with heterogeneous energy capacity is when SNs are endowed with energy harvesting capabilities. The solution ideally applies to environments with uniform harvesting potentials, such as solar harvesting in open space areas (e.g., military and border surveillance in a desert), etc., where a node can either be energy harvesting enabled or energy harvesting disabled, with no differentiation between harvesting capabilities when the energy is available (temporal variation is not consider here). Although considering general environments with spacial variant harvesting potentials is out of the scop of the current work, the proposed solution can easily be extended to such environments. First, the weight function should be updated. A normalized weight in the interval $[0,1]$ may be used, where $W(u) = 1$ for a node with a maximum harvesting potential, and, $W(u) = 0$, for a node with no harvesting potential/capability. If the maximum harvesting potential is denoted by, MHP, and the harvesting potential of a node, u, is denoted by, $HP(u)$ ($HP(u) \in [0, MHP]$), then a trivial normalized function would be: $W(u) = (MHP - HP(u))/MHP$. A threshold, say τ, should be defined for the use of SNs in relaying traffic. Only nodes with weights lower than that of the normalized threshold weight, i.e, $W_t = (MHP - \tau)/MHP$, can then be used as relays. The set H is then redefined as the set of nodes with weights lower than W_t, and the same algorithms (both for exact and heuristic solutions) may be used. RNs are simply placed at positions of nodes with a weight higher than W_t in the resulted MWCDS.

VI. Numerical Analysis

The proposed solution and model for minimum RN addition (MRA) are evaluated by simulation. Both the exact solution (ILP) and the heuristic algorithm are evaluated, denoted MRA-ILP and MRA-heuristic, respectively. They are compared to i) the use of MCDS-based solution in the one-tiered model (MCDS-1Tiered), which consists in the use of traditional MCDS calculation (without weights) then using the SNs in the set as relays without replacement, ii) MCDS-based solution in the two-tiered model (MCDS-2Tiered), i.e., MCDS calculation then replacement of all the obtained nodes in the set by dedicated RNs, and finally iii) a trivial energy-aware (TEA). TEA simply calculates the shortest paths on the node weighted graph, where weight 0 is assigned to ERNs and weight 1 to ELNs, similarly to the proposed model but without RNs addition. The comparison metrics include, i) the network lifetime, defined as the time to first battery drain out, ii) the cost in terms of the number of RNs added, and iii) the run-time. We used NetworkX[2] [29] environment to implement the network simulator, and CPLEX[3] to solve the ILP. To measure these performance metrics, the following parameters have been varied: i) number of nodes, ii) the percentage of ERNs, and iii) the average network degree. For every generated topology, a single node is randomly picked up as a BS, and every other

[2]https://networkx.lanl.gov

[3]www.ibm.com/software/products/en/ibmilogcpleoptistud

node periodically generates and transmits a packet to the BS in every cycle. Parameters of $CC2420$ [4] radio have been used to measure energy consumption. Every point in the following plots is the average of extensive repetitions with different random generated graphs, and the error bars are presented with 95% confidence interval.

Before comparison, the parameter, ϵ, of the proposed heuristic is first investigated. Note that this parameter presents a trade-off between the cost (quality of the obtained solutions) and the runtime. High values tend to accelerate the search but for a lower quality (higher cost), while lower values have the opposite impact. This trade-off is analyzed hereafter. Four variants of the heuristic with different values of ϵ $(0.1, 0.5, 1, 1.5)$, as well as the exact solution (ILP), are compared in Fig. 1 and Fig. 2. Fig. 1 shows that the costs smoothly increase as the number of nodes grows, and that the low values of ϵ reduce the cost. The difference between all the variants and the optimum represented by the ILP also rises smoothly. However, Fig 2 shows large difference between the heuristic's variants and the ILP in terms of runtime (note the log scale). The runtime of the latter grows exponentially, as well as its memory footprint. This makes it impossible to simulate scenarios with high number of nodes. The impact of ϵ here is reversed compared to the cost, where lower values of ϵ need higher runtime, but the difference between the variants is still relatively small. In the following, ϵ is set to 0.5, as a balance between optimality (added RNs) and runtime.

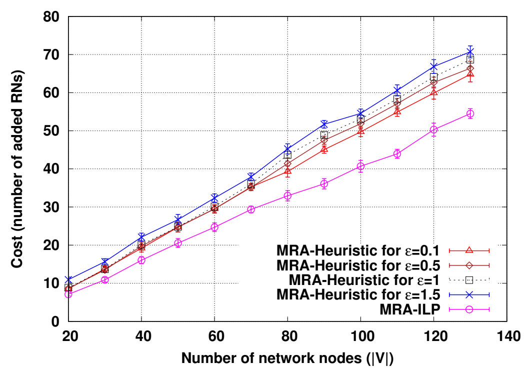

Fig. 1: Cost vs. number of nodes

The network lifetime in terms of number of cycles is presented in Fig. 3, Fig. 4, and Fig. 5. All the figures show large difference between solutions using RNs addition (MRA and MCDS-2Tiered) and those based on the one-tiered model (MCDS-1tiered, TEA). In the former solutions, the energy consumption that affects the network lifetime is that for transmitting the nodes' own readings. This is as only ERNs and dedicated RNs– which are energy unconstrained nodes– are used to forward traffic. Therefore, the lifetime becomes only proportional to the network traffic in the simulated scenarios, which explains its invariance in all the plots for MRA and MCDS-2Tiered. This also explains the same performance of ILP and heuristic versions of MRA (presented with a single plot in every figure). While MRA and MCDS-2Tiered exhibit

[4]https://inst.eecs.berkeley.edu/∼cs150/Documents/CC2420.pdf

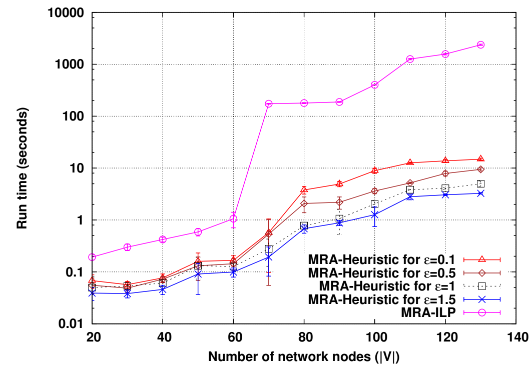

Fig. 2: Runtime vs. number of nodes

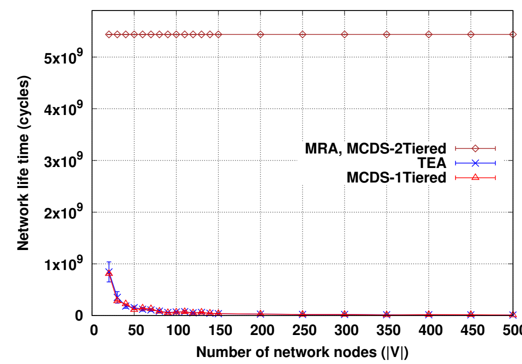

Fig. 3: Network lifetime vs. number of nodes

the same network life time, plots of TEA and MCDS-2Tiered have the same shape in all figures with minor and insignificant fluctuating difference that is due to different selection of relays amongst candidate SNs. The increase in number of nodes raises the relaying load, notably for ELN. This explains the dramatic drop of the network lifetime for TEA and MCDS-2Tiered in Fig. 3, from to $0.8 * 10^9$ to about $0.95 * 10^7$. Both solutions are affected by the increase of the load due to the inevitable use of ELNs for relaying (routing) data, while the use of RNs addition eliminates this problem and keeps the network lifetime at $5.4 * 10^9$. This represents a reduction at a ratio ranging from about 5 times to more 500 times. The increase of ELNs percentage raises the use of such nodes, which justifies the drop of lifetime in Fig. 4. However, the grow of the network degree gives more alternatives when computing the shortest paths in TEA (resp. the MCDS in MCDS-1Tiered) and reduces the ELN use, which explains the grow of the lifetime with this parameter in Fig 5. Note the huge difference between the two categories of solutions for high values of ELN in Fig. 4 (resp. low values of network degree), where the ratio is more than 500 times.

This improvement in the network lifetime comes at a cost of adding RNs as presented in Fig. 6, Fig. 7, and Fig. 8. Note that TEA and MCDS-1Tiered are not presented in these figures as they have no cost with this respect (does not add RNs as long as connectivity can be ensured by SNs). Fig. 6 shows that the inevitable increase of the cost vs. the number

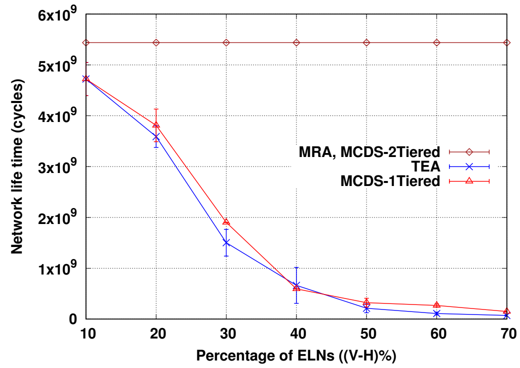

Fig. 4: Network lifetime vs. percentage of ELNs

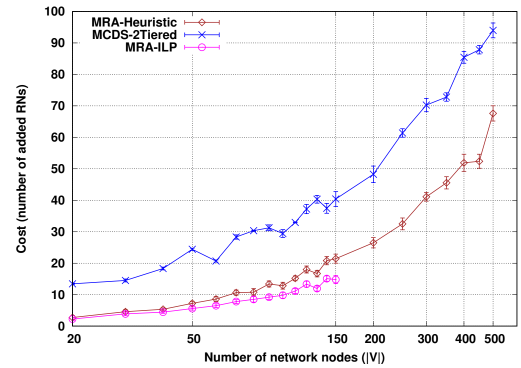

Fig. 6: Cost vs. number of nodes

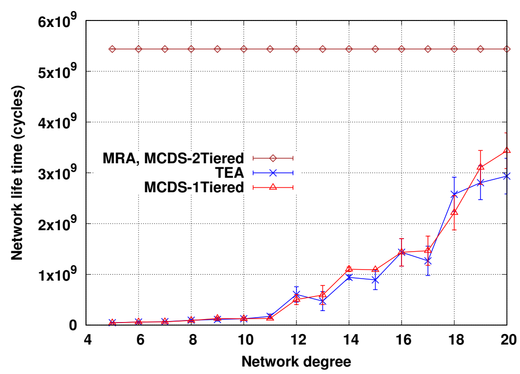

Fig. 5: Network lifetime vs. network degree

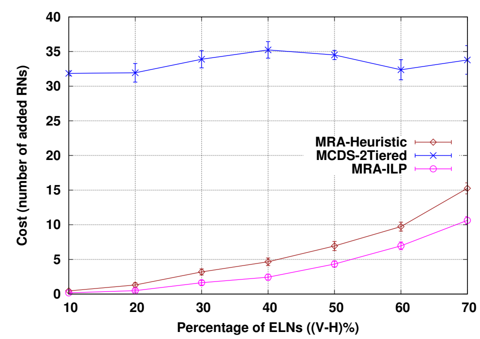

Fig. 7: Cost vs. percentage of ELNs

of nodes is smooth and confirms scalability. The cost of the heuristic version of MRA does not exceed 68 nodes for a network of 500 nodes. The ILP cost is even a bit lower but it was not possible to assess it for scenarios beyond 150 nodes. More importantly, MRA has considerable lower cost than MCDS-2Tiered. The difference grows with the number of nodes, and MRA provides up to 40% reduction for large scales. In Fig. 7, the increase in the percentage of ELNs inevitably affects MRA, but not MCDS-2Tiered that selects all nodes in the CDS without making any distinction between SNs. However, it is important to mention that the cost of MRA remains reasonable even for high percentage of ELN, and the difference with MCDS-2Tiered is always high. For instance, for 70% of ELN, it was below 16 nodes (out of the 150 nodes set in this scenario) for MRA-heuristic, and bellow 12 nodes for the ILP, while it ranges between 32 and 35 for MCDS-2Tiered. This represent a reduction of more than 50% for MRA-heuristic over MCDS-2Tiered and up to 60% for MRA-ILP. The growth of the network degree helps reducing the number of dominators when calculating the MCDS, which explains the drop of all plots in Fig. 8. Large difference between both versions of MRA and MCDS-2Tiered can be remarked from the figure.

Contrary to the MRA-heuristic, the reduction of the cost in ILP comes at a very important increase in the runtime, as shown in Fig. 9, Fig. 10, and Fig. 11. Theses figures plot the runtime of MRA-ILP, MRA-heuristic, and PTAS-heuristic (Polynomial Time Approximation Scheme) that uses a state-of-the-art heuristic for MWCDS calculation [2] (on which the proposed heuristic is based). However, PTAS-heuristic does not take advantage of the particularity of the problem (weights) in the search of the set. Comparison with PTAS-heuristic allows to asses the runtime performance of the proposed heuristic. Note that the cost of PTAS-heuristic (not presented in the previous figures) is the same as MRA-heuristic. Fig. 9 shows exponential increase of ILP for networks with number of nodes beyond 50, while the increase for both heuristics remains smooth and polynomial. However, the figure confirms that the difference between the heuristics increases with the number of nodes, and that the increase in MRA-heuristic is smoother than PTAS-heuristic. For example, MRA-heuristic runtime does not exceed $40sec$ for 500 nodes, while PTAS-heuristic runtime exceeds $90sec$. Fig. 10 shows that the heuristics are unaffected by the increase of the percentage of ELNs and its runtime remains bellow $1sec$, contrary to the ILP. MRA-heuristic still exhibits the lowest runtime. Finally, Fig. 11 also confirms the lowest runtime for MRA-heuristic that was bellow $1sec$, and highest for MRA-ILP. The decrease with the rise of the network degree for PTAS-Heuristic is justified by the fact that the increase in connectivity (network degree) accelerates the finding of dominators as their numbers becomes lower with this increase (Fig. 8). MRA-heuristic compensates the time needed in case of low degrees by taking advantage of the particularity of the problem for accelerating the dominating set construction, which explains its steady

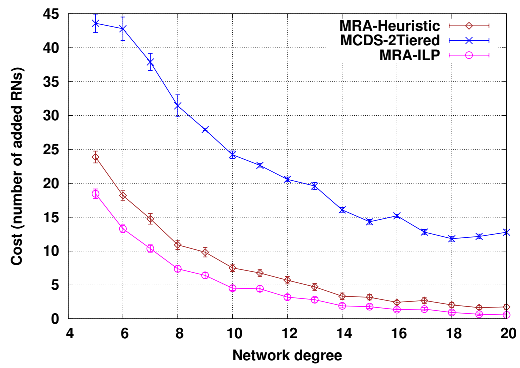

Fig. 8: Cost vs. network degree

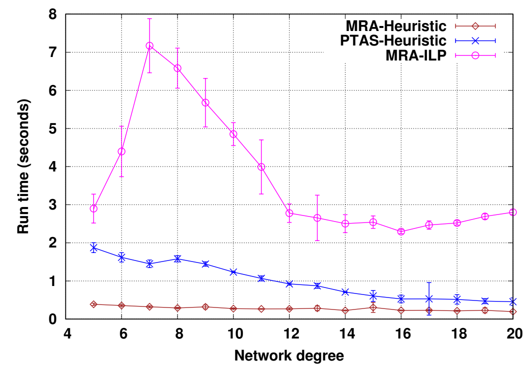

Fig. 11: Runtime vs. network degree

performance that is almost unaffected by the degree. The fluctuation of ILP for network degrees between 5 and 8 is due to the optimization methods used by CPLEX when searching optimal solution. We realized that many branches are eliminated during the exploration of the solutions by CPLEX. But for degrees between 5 and 8, a lot of branches have been explored before convergence (it was not possible to eliminate their exploration), which justifies the increase. This also explains the slight increase after degree 16. Finally, it is important to notice large difference between MRA-heuristic and MRA-ILP.

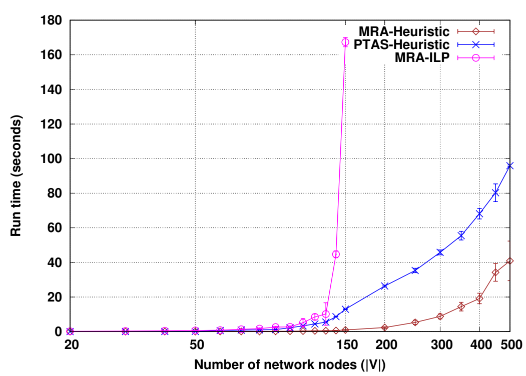

Fig. 9: Runtime vs. number of nodes

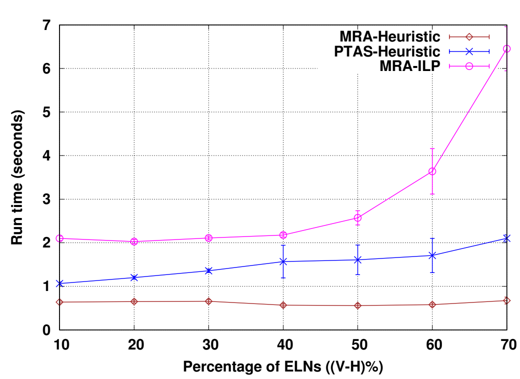

Fig. 10: Runtime vs. percentage of ELNs

VII. Conclusion

Stainability in wireless sensor networks (WSN) has been considered in this paper from the perspective of energy-aware communication coverage. A general environment has been considered, where two types of sensor nodes (SNs), energy rich nodes (ERNs) vs. energy limited nodes (ELNs) may be deployed. Upon deployment, the proposed solution aims at limiting the use of ELN to data reading, and ensuring coverage via ERNs, with addition of a minimum number of relay nodes (RNs) for coverage. This is completely different from the pure one-tiered and two-tiered models used in the literature. The problem has been reduced to finding the minimum weighted connected dominating set (MWCDS) in a vertex weighted graph. An integer linear program (ILP) has been derived as an optimal solution for the problem in terms of the number of RNs to be added. Given the exponential computation complexity of ILP solvers, a heuristic has been proposed. Upper bounds for the approximation of the heuristic to the optimum, as well as for its runtime, have been formally derived.

The proposed model and solutions have been compared by simulation with the use of traditional MCDS calculation and replacement of all the obtained set by RNs, which is equivalent to use an MCDS calculation based solution in the two-tiered model (MCDS-2Tiered), as well as to the MCDS calculation then using the SNs in the set as relays without replacement, which is equivalent to use an MCDS calculation based solution in the one-tiered model (MCDS-1Tiered). A trivial energy-aware solution is also used in the comparison (TEA), which minimizes the use of ELNs by calculating the shortest paths on the node weighted graph without any RN addition. The proposed heuristic for MWCDS calculation has also been evaluated and compared with a state-of-the-art algorithm. The comparison has been performed with respect to energy efficiency (network lifetime), the cost (number of added RNs), and the runtime.

Results of extensive simulations confirm effectiveness of the proposed solution, both when using the ILP and the heuristic. The proposed solution considerably prolongs the network lifetime to the order of more than 500 times compared to TEA and MCDS-2Tiered in large scale networks and scenarios with high number of ELNs. The improvement in the network lifetime has an inevitable cost of adding a reasonable number of RNs that was lower than the number added by MCDS-2Tiered, where the reduction in cost compared to the latter was at the order of 40% for large scale networks. The results confirm considerable reduction in runtime for the proposed heuristic vs. a state-of-the-art polynomial time approximation scheme (PTAS), which exceeded 50% in large scale networks, as well as its scalability, contrary to the ILP. This is at the cost of adding a higher but a reasonable number of RNs compared to ILP.

The proposed solution supposes that the ERNs have enough capacity of energy (e.g. by harvesting or access to high capacity storage) to keep their batteries alive all the time. Although technologies are evolving, it is early to technically fulfill this assumption for high data rate applications, such as those involving video/images transmissions. Relaxing this assumption represents a perspective. This will be tackled first in the particular scenario of energy harvesting environments by considering limited harvesting capacities as well as spatialy/temporaly variable capacities at harvesting nodes.

References

[1] R. Liu, I. J. Wassell, and K. Soga, "Relay node placement for wireless sensor networks deployed in tunnels," in *IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2010*, 2010, pp. 144–150.

[2] Z. Wang, W. Wang, J.-M. Kim, B. Thuraisingham, and W. Wu, "Ptas for the minimum weighted dominating set in growth bounded graphs," *J. of Global Optimization*, vol. 54, no. 3, pp. 641–648, nov 2012.

[3] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Trans. Mob. Comput.*, vol. 11, no. 8, pp. 1399–1411, 2012.

[4] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed fault tolerance," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 216–228, 2010.

[5] X. Cheng, D. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *ACM/Springer Wireless Network*, vol. 14, no. 3, pp. 425–443, 2007.

[6] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: formulation and approximations," *IEEE/ACM Trans. on Networking*, vol. 18, no. 2, pp. 434–447, April 2010.

[7] Z. Zheng, L. X. Cai, R. Zhang, and X. S. Shen, "Rnp-sa: Joint relay placement and sub-carrier allocation in wireless communication networks with sustainable energy," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3818–3828, 2012.

[8] Q. Chen, Z. Hu, Y. Chen, V. Grout, D. Zhang, H. Wang, and H. Xing, "Improved relay node placement algorithm for wireless sensor networks application in wind farm," in *IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, 2013.

[9] A. Efrat, S. P. Fekete, P. Gaddehosur, J. Mitchell, V. Polishchuk, and J. Suomela, "Improved approximation algorithms for relay placement," in *Proc. 16th Ann. European Symp. Algorithms (ESA '08)*. Springer Berlin Heidelberg, 2008, pp. 356–367.

[10] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 1667–1675.

[11] E. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Trans. Computers*, vol. 56, no. 1, p. 134138, 2007.

[12] F. Wang, D. Wang, and J. Liu, "Traffic-aware relay node deployment: Maximizing lifetime for data collection wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1415–1423, 2011.

[13] A. Kashyap, S. Khuller, and M. Shayman, "Relay placement for higher order connectivity in wireless sensor networks," in *IEEE INFOCOM*, 2006, pp. 1–12.

[14] A. Chelli, M. Bagaa, D. Djenouri, I. Balasingham, and T. Taleb, "One-step approach for two-tiered constrained relay node placement in wireless sensor networks," *IEEE Wireless Commun. Letters*, vol. 5, no. 4, pp. 448–451, 2016.

[15] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commu. Surveys and Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.

[16] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *IEEE Commun. Surveys and Tutorials*, vol. 16, no. 1, pp. 195–213, 2014.

[17] Z. Ding and H. V. Poor, "Cooperative energy harvesting networks with spatially random users," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1211–1214, 2013.

[18] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks," in *IEEE MASS*, 2012, pp. 75–83.

[19] N. Michelusi and M. Zorzi, "Optimal random multiaccess in energy harvesting wireless sensor networks," in *IEEE ICC Workshops*, 2013.

[20] P. A. and S. Li, "Light-harvesting wireless sensors for indoor lighting control," *IEEE Sensors*, vol. 13, no. 12, pp. 4599–4606, 2013.

[21] D. Niyato, E. Hossain, M. M. Rashid, and V. K. Bhargava, "Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management," *IEEE Wireless Commun.*, pp. 90–96, 2007.

[22] A. Castagnetti, A. Pegatoquet, T. N. Le, and M. Auguin, "A joint duty-cycle and transmission power management for energy harvesting wsn," *IEEE Trans. Industrial Informatics*, vol. 10, no. 2, pp. 928–936, 2014.

[23] P. Zhang, G. Xiao, and H.-P. Tan, "Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors," *Computer Networks*, vol. 57, no. 14, pp. 2689–2704, 2013.

[24] D. Djenouri and M. Bagaa, "Energy harvesting aware relay node addition for power-efficient coverage in wireless sensor networks," in *IEEE International Conference on Communications, ICC, London, UK*, 2015, pp. 86–91.

[25] S. Misra, N. E. Majd, and H. Huang, "Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks," *IEEE Trans. Computers*, vol. 63, no. 12, 2014.

[26] Y. Huang, X. Gao, Z. Zhang, and W. Wu, "A better constant-factor approximation for weighted dominating set in unit disk graph," *J. Comb. Optim.*, vol. 18, no. 2, pp. 179–194, 2009.

[27] T. Nieberg and J. Hurink, "A PTAS for the minimum dominating set problem in unit disk graphs," ser. Lecture Notes in Computer Science, vol. 3879, 2006, pp. 296–306.

[28] M. L. Fredman and D. E. Willard, "Trans-dichotomous algorithms for minimum spanning trees and shortest paths," *J. Comput. Syst. Sci.*, vol. 48, no. 3, pp. 533–551, Jun. 1994.

[29] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," ser. 7th Python in Science Conference (SciPy2008), 2008.