Construction of a Minimum Energy Broadcast Backbone with Bounded Delay in Heterogeneous Wireless Sensor Networks

Karla Breschi
Univ. Bourgogne Franche-Comté
FEMTO-ST Institute/CNRS
Email: kjimenez@femto-st.fr

Julien Bernard
Univ. Bourgogne Franche-Comté
FEMTO-ST Institute/CNRS
Email: julien.bernard@femto-st.fr

***Abstract*—Efficient broadcast in wireless sensor networks can exploit backbone structures, in relation to connected dominating sets in graphs. This paper proposes the construction of a backbone for broadcast in heterogeneous wireless sensor networks having minimum energy consumption and a delay bounded by a predefined constant $h_{\max}$. For the purpose of the paper, we assume the sensors composing the network have an energy belonging to $[W_{\min}; W_{\max}]$ such that $W_{\max} = c * W_{\min}$ where c is a predefined constant. To address this problem we propose two different approaches. For small instances, we propose an integer linear program that computes an optimal solution for the problem. Since solving an integer linear program is NP-hard, obtaining solutions for large network instances may be impossible in a reasonable time, thus we propose an approximation algorithm that computes a solution for the problem in polynomial time and whose approximation ratio is $c * h_{\max}$.**

I. Introduction

A wireless sensor network is a set of sensors deployed in a more or less random manner forming a multi-hop network. The objective of such networks is to collect information from the environment, such as temperature, and to route them to a particular node named sink. A sensor is an embedded system characterized by an energy constraint due to the fact that they are battery powered. Often the sensors are placed in hostile areas difficult to access for a human, therefore saving energy is a primary problem. Wireless sensor networks are used in many fields such as aeronautics, automotive and medical surveillance.

Broadcast in wireless sensor networks is a fundamental problem that is linked to many applications. As soon as the base station needs to transmit an information to the whole network, broadcast is necessary.

The basic strategy called flooding, which consists in transmitting every received messages, consumes a large amount of energy, a critical resource for a wireless sensor network, directly influencing the network lifetime. Furthermore, flooding consumes bandwidth, increasing collisions and duplicating messages. The idea for broadcast protocols is to determine a subset of nodes, called *backbone*, that will transmit the messages, in order to reach all network nodes. A backbone brings many advantages: it removes unnecessary transmission links, reduces the communication overhead, reduces the redundancy and, by decreasing the energy consumption, increases network lifetime.

The backbone construction depends on the issue to be solved. In our case, the objective is to optimize two metrics at the same time: energy and delay. Energy consumption is strictly related to network lifetime whereas delay is associated with the network performance. Bounding the delay improves power consumption and reliability because fewer hops are needed to reach the targets. This also results in the reduction of transmission failures because the longer the path between the source and the destination, the higher the probability of failure.

In this paper, we propose an Integer Linear Program (ILP) to construct a minimum energy broadcast backbone with bounded delay. Since solving an ILP is NP-hard, we propose an approximation algorithm in order to solve the problem in polynomial time.

This paper is organized as follows: Section II presents the related works, Section III gives the definition of the problem treated in this paper, Section IV provides the definition of an Integer Linear Program giving an optimal solution for an energy efficient backbone with bounded delay, Section V presents an approximation algorithm solving the same problem, Section VI gives a theoretical study in order to prove the approximation ratio of the proposed algorithm and section VII summarizes the conclusions of this work.

II. Related Work

The problem of determining an optimal backbone can be translated in the graph theory as determining an optimal Connected Dominating Set (CDS). The construction of a CDS has been widely studied with different approaches: linear programming, genetic algorithms or greedy algorithms. Different algorithmic models were used to represent a sensor network: for instance unit disk graphs, disk graphs or more restrictive graph classes like chordal graphs (definition will be given in the next section). According to the parameters to be optimized, several variations of CDS were proposed [1].

We will focus on works that aim at optimizing energy and delay.

A. Backbone and bounded diameter

A number of works studying the construction of connected dominating sets with bounded diameter are available. Their goal is to minimize the diameter of the graph, i.e. the transmission delay in the context of wireless sensor networks.

Buchanan et al [2] introduce the notion of s-club to bound the distance traveled by the information and consequently reduce the latency. An s-club is a connected dominating set whose induced subgraph has a diameter at most s. In their paper, the authors prove that finding an s-club is a NP-complete problem.

Schaudt [3] proves that for all $k \geq 1$, the problem of deciding if a graph has a dominating set whose induced subgraph has a diameter at most k is NP-complete. Therefore the author decided to concentrate his efforts on a restricted class of graphs admitting a set with these particular features: chordal graphs. A graph is said to be chordal if each of its four or more cycle vertices has a chord, i.e. an edge connecting two non-adjacent vertices of the cycle. Using that specific class of graphs, he showed the problem can be solved in $O(mn)$ time in a graph with n vertices and m edges.

Li et al [4] deal with the construction of a bounded diameter connected dominating set whose size is minimized. The authors propose an algorithm with constant approximation factor in the case of unit disk graphs and a time complexity of $O(n^2)$ in a graph with n vertices.

Kim et al [5] build connected dominating sets in unit disk graphs considering the following metrics: size, diameter and average hop distance. The authors propose two centralized algorithms and a distributed version for the second one. The main idea of those algorithms is to build a tree using the depth first search while seeking a maximal independent set. Finally, they connect the retained nodes in the maximal independent set.

Akbari Torkestani [6] underlines the fact that when the degree constraint decreases, the backbone delay increases. As a result, he treats the problem of the construction of a connected dominating set with minimum weight and an optimal degree in order to obtain an energy efficient backbone whose delay is bounded. For this purpose, the author develops a heuristic based on learning automata.

Krumme et al [7] present a work dealing with the construction of multicast trees with minimum eccentricity. The authors propose a polynomial time algorithm that exploits geometry properties to construct a multicast tree that allows connecting a set of sources nodes to a set of sink nodes while minimizing the hop distance between this two groups.

B. Backbone and energy

There are some works treating the construction of a backbone for routing like [8] and [9]. In both contributions, the network is represented by an undirected graph where all the nodes have the same transmission range. The authors try to construct a load-balanced backbone by minimizing the maximum degree of the dominating nodes.

The paper [10] proposes a self-stabilizing algorithm allowing to build a connected dominating set with minimal energy for the routing. To achieve this objective, the authors propose a solution based on topology control. Their algorithm reduces the transmission range of the nodes while ensuring the connectivity of the network which allows them to reduce the energy consumption.

Li et al [11] treat the construction of a broadcast tree that minimizes the total energy cost. They prove that this problem is NP-hard and propose two heuristics and one approximation algorithm. The authors model the network by a directed graph where each node has its own transmission power.

C. Comparison with our work

The above-mentioned works deal with the construction of a backbone whose energy is minimized and/or delay is limited. Most of them treat either one or the other metric and often build a backbone for routing. Only [6] focuses on both the energetic cost and delay.

Our goal is to optimize both energy consumption and delay in the case of a broadcast. We choose to model a heterogeneous wireless sensor network as a directed disk graph as this model is more generic than unit disk graphs. In particular, this model allows taking into account the asymmetrical links. In addition, to solve the problem, we will use a different method: linear programming for small program instances and an approximation algorithm in the other cases.

To build an efficient energy backbone whose delay is limited, we will use the same metric employed in [7]: the eccentricity. The difference with what we propose is that we concentrate our efforts in a more general case using directed graphs and we seek to optimize energy at the same time.

III. Problem definition

A heterogeneous wireless sensor network is represented by a graph $G = (V, E)$, where $V = \{0, \ldots, n-1\}$ is the set of n vertices that corresponds to wireless nodes and $E \subset V * V$ is the set of edges that represents the wireless connections between nodes, i.e. a directed edge (i, j) models the fact that j is in the communication range of i. We define the neighborhood of i to be $N^+(i) = \{j, (i, j) \in E\}$. We denote the node 0 as the base station of the wireless sensor network. V^* is equal to V without the base station. The eccentricity of node 0, noted $\epsilon(0)$, is the longest of the shortest paths to every other node in terms of hop distance.

To define connectivity, we use the disk graph model (DG) [12]. More precisely, if r_i is the transmission range of node i, and d_{ij} is the euclidian distance between node i and node j then j is a neighbour of i if $d_{ij} \leq r_i$. This model takes into account the heterogeneity of a wireless sensor network by assigning different transmission ranges to the various nodes.

We note by W_i the energy spent by node i to send data to all the nodes in his neighborhood. There is no constraint on how to calculate the energy consumption (e.g.: First order radio model [13]).

A dominating set of graph G is a subset D of V such that every vertex not in D is in the neighborhood of a vertex in D. The nodes belonging to D are named dominators or dominating nodes and the nodes belonging to $V \setminus D$ are the dominated nodes or dominatees. A dominating set D of graph G is a connected dominating set (CDS) if D induces a connected subgraph of G. So computing a connected dominating set in a graph is equivalent to computing a backbone in a wireless sensor network. The aim is to reduce the energy consumption so we assign a weight to each node i which is the energy W_i necessary to send a message to its neighborhood. The objective is to choose a subset of V respecting the dominance and connectivity properties and such that the sum of the selected nodes' weights is minimized. This problem is known as Minimum Weighted Connected Dominating Set (MWCDS).

In our case, we want, in addition, to bound the delay during the broadcast. As the delay is proportional to the number of hops, we compute a CDS with minimum weight and where the hop distance from the base station is bounded by a constant $h_{\max}$. Note that $h_{\max}$ must be at least $\epsilon(0)$ in order to reach the node with the longest path from the base station. We call this problem Minimum Weighted Connected Dominating Set with Bounded Hop Distance (MWCDS-BHD).

IV. Integer Linear Programming Formulation

In this section, we introduce a mixed integer linear program modeling the MWCDS-BHD problem. The program builds a backbone tree with minimum weight whose height is bounded by $h_{\max} - 1$.

The formulation requires the following decision variables. First, $x_i, i \in V$ is a binary variable which indicates if the node i belongs to the backbone ($x_i = 1$) or not ($x_i = 0$). Then, in order to ensure that the set is connected, we construct a directed tree that is rooted in node 0, the base station. The nodes, belonging to the directed tree obtained by the ILP and that are not a leaf, represent the backbone nodes. Decision variable $y_{i,j}$, $(i,j) \in E$ tells whether edge (i,j) is in the directed tree formed by the backbone nodes. In other words, if $y_{i,j} = 1$, then nodes i and j are in the backbone and node i is the father of node j in the directed tree. Finally, the decision variables $u_i, i \in V$ guarantee that there is no cycle in the backbone.

The objective function of the integer linear program aims to minimize the total energy cost W of the constructed backbone. Mathematically, this is written as follows: $W = \sum_{i \in V} x_i W_i$. The constraints are defined as follows:

$$\forall i \in V^*, \sum_{\{j | i \in N^+(j)\}} x_j \geq 1 \qquad (1)$$

$$\forall i \in V^*, \sum_{\{j | i \in N^+(j)\}} y_{j,i} = x_i \qquad (2)$$

$$\forall (i,j) \in E, x_i + x_j - 2y_{i,j} \geq 0 \qquad (3)$$

$$\forall (i,j) \in E, (n+1)y_{i,j} + u_i - u_j + (n-1)y_{j,i} \leq n \qquad (4)$$

$$\forall i \in V, u_i \leq h_{\max} \qquad (5)$$

$$\sum_{i \in V} x_i - \sum_{(i,j) \in E} y_{i,j} = 1 \qquad (6)$$

$$x_0 = 1 \qquad (7)$$

$$u_0 = 1 \qquad (8)$$

Equation 1 ensures dominance: each node must either belong to the backbone or be adjacent to a node belonging to the backbone. Equation 2 specifies that only the backbone nodes have exactly one father in the tree. The other nodes have no father. Equation 3 asserts that when an edge belongs to the backbone, then both its ends are part of the backbone too.

Equation 4 is inspired by [14] where it is used to prevent cycles in the traveling salesman problem. This equation is also used in [15] for the same purpose on a spanning tree. It allocates a strictly positive number u_i for each node in the backbone that represents its distance (plus one) from the base station in the tree. If nodes i and j are in the backbone and $y_{i,j} = 1$, then it ensures that $u_j = u_i + 1$. If one of the nodes i or j is not in the backbone, the difference between u_i and u_j is undefined (bounded by the maximum possible difference).

Equation 5 ensures that the distance in terms of hop number is bounded by $h_{\max}$. The variable u_i represents the depth of node i in the backbone tree. So bounding the depth of each node in the backbone ensures that the height of the tree is bounded too. Equation 6 stipulates that the number of backbone nodes is equal to the number of backbone edges plus one (property of trees). The last two equations (7 and 8) give the values for the base station. This integer linear program can be used to find optimal solutions. Unfortunately, the time taken to solve the problem can be quite long, even for moderated size networks due to the NP-hard nature of the problem.

V. Approximation algorithm

In this section, an approximation algorithm calculating an MWCDS-BHD is proposed. The algorithm aims to calculate an energy efficient backbone and ensures that all paths are smaller than $h_{\max}$.

A. Principle

Given a weighted directed graph, the algorithm computes, at first, both arborescences: $\mathcal{A}_1$ and $\mathcal{A}_2$. $\mathcal{A}_1$ is obtained by a shortest path algorithm where the weight is a unit value for each node. Thus, paths in $\mathcal{A}_1$ minimize the number of hops from the base station to any node. $\mathcal{A}_2$ is constructed by a shortest path algorithm where the weight is the energy consumption W_i of each node. To take into account the energy consumed by each node, we have tagged each outcoming edge of a node by its weight W_i. Paths in $\mathcal{A}_2$ minimize the energy consumed to reach the destination node.

The principle of the algorithm is to choose one node k and then to consider $P_1(k)$ and $P_2(k)$, respectively the path between the base station and node k in the arborescence $\mathcal{A}_1$ and $\mathcal{A}_2$. As $h_{\max}$ is at least greater than $\epsilon(0)$, the length of $P_1(k)$ is less than or equal to $h_{\max}$. Since $P_2(k)$ is the path minimizing the energy cost, to connect k to the base station we look at first to path $P_2(k)$ and if its length is less than or equal to $h_{\max}$, we keep it. Otherwise, we choose $P_1(k)$. This ensures that every path is less than or equal to $h_{\max}$. Therefore, the question arises as to what is the energy cost that is added when the chosen path is P_1 instead of P_2. This is discussed in the section V-C.

To describe the algorithm, we introduce the definition of several sets of nodes. The set B is the set of nodes which are currently part of the backbone. The set U is the set of remaining nodes i.e. the nodes that are untreated yet. Initially, U is equal to V. D is the set of dominated nodes. At the beginning of each iteration, the following equation will be true: $V = B \cup D \cup U$.

B. Bounded Backbone Algorithm (BB)

In this section, we will explain in detail the process of the proposed approximation algorithm described in Algorithm 1. As can be seen, at the beginning, the algorithm initializes the sets B, U and D. The main idea is to select iteratively a node k that is not reached by the broadcast yet. BB chooses the most distant node not covered yet in $\mathcal{A}_1$ i.e. the distance is considered in terms of the number of hops. Once the appropriate node has been selected, the next step is to choose a path for reaching it between the paths $P_1(k)$ and $P_2(k)$ as described before. Afterward, the nodes belonging to the chosen path are added to the backbone and removed from the set U. Furthermore, the set of dominated nodes is updated.

At the end of each iteration, the energetic cost of nodes added to B is set to zero (setting to zero phase) and, then, since the shortest paths change following this update, the $\mathcal{A}_2$ arborescence is computed again.

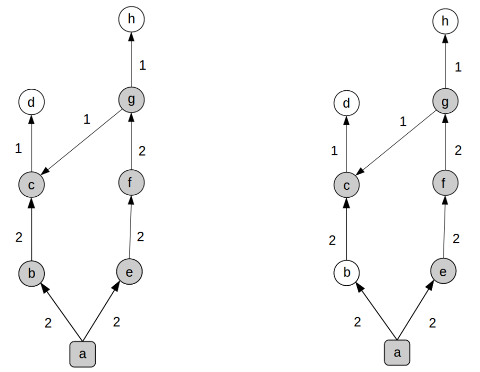
The processing explained above is repeated until the set U is empty. The solution is given by the set B constructed by this process.

Setting to zero the energy of the nodes added to the backbone allows us to choose paths containing nodes belonging to the backbone. Indeed, since they already belong to the backbone, they will not increase the cost of the backbone. To better explain, consider the example of figure 1.

In this figure, we assume $h_{\max} = 5$. The backbone of figure 1a represents the backbone obtained without the setting to zero phase. It is obtained as follows. At the first iteration, BB chooses to treat node h. The chosen path is $P = (a, e, f, g, h)$. Therefore, nodes a, e, f and g are added to the set B, and they are removed from U. Nodes h, c and b are added to D.

At the second iteration, the node to be processed is d. The path selected is $P = (a, b, c, d)$. Nodes b and c are added to the backbone set B and are removed from U. Finally, d is added to D. The algorithm terminates since the set U is empty. At the end, the backbone consists in nodes a, b, c, e, f, and g. Its energetic cost is 10.

The backbone represented in figure 1b is less expensive in terms of energy. Indeed, the backbone consists of nodes a, c, e, f, and g and has an energetic cost of 8. At the second iteration, the zeroing of the energy of the backbone nodes



(a) (b)

Fig. 1. Explanation of the zeroing of energy by BB. Backbone obtained without zeroing (on the left) and backbone obtained by setting to zero the energies (on the right)

allows us to consider the path $P = (a, e, f, g, c, d)$ and to select it because its energy cost is inferior to the energy of $P = (a, b, c, d)$. Indeed, $W_a = W_e = W_f = W_g = 0$. With this method, we obtain the backbone in figure 1b. Therefore, by setting energy costs to zero, BB selects the paths that will minimize the energy cost of the backbone.

Algorithm 1 Bounded Backbone algorithm

Input: Graph G (WSN), Maximum hop distance $h_{\max}$, Energy $W_{\max}$ and $W_{\min}$
Output: Backbone B

```
B ← ∅
U ← V
D ← ∅
while U ≠ ∅ do
    k ← the most distant node in A_1 not covered yet
    if L(P_2(k)) ≤ h_max then
        P ← P_2(k)
    else
        P ← P_1(k)
    end if
    B ← B ∪ {x | x ∈ P ∧ x ≠ k}
    U ← U \ {x | x ∈ P}
    D ← D ∪ ⋃_{(x∈P)∧(x≠k)} N^+(x)
    for x such that x ∈ P ∧ x ≠ k do
        //setting to zero phase
        W_x ← 0
    end for
    Compute A_2
end while
return B
```

The time complexity of the Bounded Backbone algorithm depends on the algorithm chosen for the calculation of $\mathcal{A}_1$ and $\mathcal{A}_2$: in our case, it is Dijkstra's algorithm. The Dijkstra's algorithm is used in the main loop to recompute $\mathcal{A}_2$. Consequently, the time complexity of BB is $O(|V| * (|V| * \log|V| + |E|))$ which is polynomial.

C. Theoretical study

Lemma 1: The set B constructed by BB is a CDS.

Proof: B is a CDS if, and only if, B is a dominating set and the graph induced by B is connected. By construction, B is a connected set. Indeed, the set B is constructed by iterative addition of paths all connected to the sink.

To prove that B is dominant we will proceed by absurd. Suppose B is not dominant. This means that:

$$\exists v \in V \setminus B, \forall u \in B, (u, v) \notin E$$

There are two possible cases: either v is in D or v is in U.

- Case $v \in D$. By construction of set D, we know that:
  $$\exists d \in B \text{ such that } v \in N^+(d)$$
  thus $(d, v) \in E$. This is in contradiction with our hypothesis.
- Case $v \in U$. Since $v \in U$ then $U \neq \emptyset$. This is impossible because in this case the stopping condition of the loop is not respected.

So B is dominant. Therefore B is a CDS. ∎

Lemma 2: In the set B, the distance in terms of hop number between the base station and any other node of the graph is at most $h_{\max}$.

Proof: Let u be a node in V. There are two possibilities:

- u is one of the nodes k selected by BB. To link u to the base station, BB selects either path $P_1(u)$ or $P_2(u)$. If $P_1(u)$ is chosen then we have: $L(P_1(u)) \leq \epsilon(0)$. By assumption, $\epsilon(0) \leq h_{\max}$ so $L(P_1(u)) \leq h_{\max}$. If the chosen path is $P_2(u)$, then we know that $L(P_2(u)) \leq h_{\max}$ otherwise it would not be chosen.
- u is not one of the nodes k selected by BB. Then there are two possibilities :
  - u is an internal node of the path of a node k selected by BB. Suppose that u is a node internal to the path of k_i. We know that $L(P(k_i)) \leq h_{\max}$, consequently $L(P(u)) \leq L(P(k_i)) \leq h_{\max}$.
  - u is a leaf from the resulting tree that has not been selected by the algorithm for processing. Since it was not chosen by the algorithm as a node to be processed, we can deduce that all the nodes necessary to connect it to the base station are already present in the backbone. Furthermore, u is in $N^+(\{x | x \in B\})$. As demonstrated previously, for any node k selected by BB, $L(P(k)) \leq h_{\max}$ so any node $x \in B$ has a hop distance to the base station of at most $h_{\max} - 1$. Suppose that y is the father of u. Then the hop distance of u is equal to the hop distance of y plus one. Finally, y is at $h_{\max}$ hops from the base station.

∎

VI. Theoretical study of BB

In this part, we present a theoretical study to determine the energy loss when the chosen path is P_1 instead of P_2. For Theorem 3, we consider the case when there are two possible energies for nodes: $W_{\min}$ and $W_{\max}$, with $W_{\max} = c * W_{\min}$. Then we prove the approximation ratio of the BB algorithm by considering that the nodes can have an energy belonging to $[W_{\min}; W_{\max} = c * W_{\min}]$.

The first theorem we want to prove is the following:

Theorem 3: The path P chosen by the algorithm has a bounded cost: $w(P) \leq c * w(P_2)$.

For this purpose, we will at first, study the structure of the P_1 path compared to the P_2 path.

1) Structure of the path P_1 with respect to P_2: We assume that P_2 consists of u nodes with energy $W_{\min}$ and v nodes whose energy is $W_{\max}$. And P_1 consists of $u - \Delta u$ nodes whose energy is $W_{\min}$ and $v + \Delta v$ nodes of $W_{\max}$ energy.

As P_1 is the shortest hop path, we have:

$$u - \Delta u + v + \Delta v \leq u + v$$

So we can infer:

$$\Delta v \leq \Delta u \tag{9}$$

Knowing that P_2 minimizes the energy, we deduce the following:

$$(u - \Delta u) * W_{\min} + (v + \Delta v) * W_{\max} \geq u * W_{\min} + v * W_{\max}$$

Thus, we obtain:

$$\Delta u \leq \left(\frac{W_{\max}}{W_{\min}} \right) \Delta v \tag{10}$$

Four cases can occur:

1) $\Delta u < 0$ and $\Delta v \geq 0$. Impossible because of equation 9 (Δu would be positive).
2) $\Delta u \geq 0$ and $\Delta v \leq 0$. Impossible because of equation 10 (Δu would be negative).
3) $\Delta u < 0$ and $\Delta v \leq 0$. Then we have $-\Delta v \leq -\left(\frac{W_{\min}}{W_{\max}}\right) \Delta u < -\Delta u$. This is a contradiction with 9.
4) $\Delta u \geq 0$ and $\Delta v \geq 0$. This case is the only possible and so we deduce the following equation:

$$0 \leq \Delta v \leq \Delta u \leq \left(\frac{W_{\max}}{W_{\min}} \right) \Delta v \tag{11}$$

2) Proof of theorem 3: With the results obtained previously, we can complete the proof of theorem 3.

Proof: The energy cost of the P_1 path can be written as follows:

$$\begin{aligned} w(P_1) &= (u - \Delta u) * (W_{\min}) + (v + \Delta v) * (W_{\max}) \\ &= u * W_{\min} - \Delta u * W_{\min} + v * W_{\max} + \Delta v * W_{\max} \end{aligned}$$

Following Equation 11, we know that $\Delta v \leq \Delta u$. So we obtain:

$$w(P_1) \leq u * W_{\min} - \Delta u * W_{\min} + v * W_{\max} + \Delta u * W_{\max}$$

Replacing $W_{\max}$ by $c * W_{\min}$, we obtain:

$$\begin{aligned} w(P_1) &= u * W_{\min} - \Delta u * W_{\min} + v * c * W_{\min} \\ &\quad + \Delta u * c * W_{\min} \\ &= W_{\min} * (u - \Delta u + v * c + \Delta u * c) \\ &= W_{\min} * (u + v * c + \Delta u * (c - 1)) \end{aligned}$$

Since $u - \Delta u \geq 0$, we can upper bound Δu by u:

$$\begin{aligned} w(P_1) &\leq W_{\min} * (u + v * c + u * (c-1)) \\ &= W_{\min} * (u + v * c - u + u * c) \\ &= W_{\min} * (v * c + u * c) \end{aligned}$$

As $W_{\max} = c * W_{\min}$, we get:

$$\begin{aligned} w(P_1) &= v * (W_{\max}) + u * (c * W_{\min}) \\ &\leq c * (v * W_{\max} + u * W_{\min}) \\ &= c * w(P_2) \end{aligned}$$

Thus we can conclude that: $w(P_1) \leq c * w(P_2)$. As the path P is equal to either P_1 or P_2, we can deduce that its energetic cost is equal to either $w(P_1)$ or $w(P_2)$. So $w(P) \leq c * w(P_2)$. ■

Theorem 4: The backbone obtained by BB has an approximation ratio of $c * h_{\max}$.

Proof: Suppose that L^* is the set of leaves in the optimal tree. We assign to each node l^* of L^* the nodes composing the path between l^* and the base station starting with the most distant node each time we are processing a new set. If a node belongs to several paths, we assign it to the first node k in which it appears. This process constructs the set $S^*(l^*)$.

Then, we assign to each node a charge such that the sum of the charges of the set l^* gives his optimal total energy denoted by $w^*(S^*(l^*))$. Thanks to the way of constructing these sets, the sum of the charge of all the nodes of the sets constructed gives the cost of the optimal tree. To calculate the approximation ratio what we need to do is to calculate the charge that each node in a constructed set can take in the tree constructed by BB. Then, as before, by summing all the charges of all the sets, we will get the cost of the tree constructed by BB. We denote $w(S^*(l^*))$ the cost of the set $S^*(l^*)$ obtain by adding the charge that nodes in $S^*(l^*)$ can take in the tree constructed by BB.

We know that the biggest charge a node can take is $W_{\max}$:

$$\forall i \in V, e_i \leq W_{\max}$$

Moreover, by construction, the number of nodes in a path is at most $h_{\max}$. Therefore, we can deduce that the cost of $w(S^*(l^*))$ is at most:

$$w(S^*(l^*)) \leq W_{\max} * h_{\max}$$

Hence, the ratio between the energy W of our algorithm and the optimal energy W^* is:

$$\begin{aligned} \frac{W}{W*} &= \frac{\sum_{l^* \in L^*} w(S^*(l^*))}{\sum_{l^* \in L^*} w^*(S^*(l^*))} \\ &\leq \frac{|L^*| * (W_{\max} * h_{\max})}{|L^*| * W_{\min}} \\ &\leq c * h_{\max} \end{aligned}$$

In conclusion, the algorithm has an approximation ratio of $c * h_{\max}$. ■

VII. Conclusion

In this paper, we focused on the problem of building an energy efficient backbone with a bounded delay for a heterogeneous wireless sensor network. To solve this problem, initially, a linear program has been proposed. However, given the complexity of the program and therefore the time needed to get a result on large networks, an approximation algorithm with polynomial time complexity has been proposed. A theoretical study has been done proving the correctness of the algorithm as well as its approximation ratio which is equivalent to $c * h_{\max}$.

References

[1] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks–a comprehensive survey," *Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013.

[2] A. Buchanan, J. S. Sung, V. Boginski, and S. Butenko, "On connected dominating sets of restricted diameter," *European Journal of Operational Research*, vol. 236, no. 2, pp. 410–418, 2014.

[3] O. Schaudt, "On dominating sets whose induced subgraphs have a bounded diameter," *Discrete Applied Mathematics*, vol. 161, no. 16, pp. 2647–2652, 2013.

[4] Y. Li, D. Kim, F. Zou, and D.-Z. Du, "Constructing connected dominating sets with bounded diameters inwireless networks," in *International Conference on Wireless Algorithms, Systems and Applications (WASA)*. IEEE, 2007, pp. 89–94.

[5] D. Kim, Y. Wu, Y. Li, F. Zou, and D.-Z. Du, "Constructing minimum connected dominating sets with bounded diameters in wireless networks," *Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 147–157, 2009.

[6] J. A. Torkestani, "Energy-efficient backbone formation in wireless sensor networks," *Computers & Electrical Engineering*, vol. 39, no. 6, pp. 1800–1811, 2013.

[7] D. W. Krumme and P. Fragopoulou, "Minimum eccentricity multicast trees," *Discrete Mathematics & Theoretical Computer Science*, vol. 4, no. 2, pp. 157–172, 2001.

[8] J. S. He, S. Ji, Y. Pan, and Y. Li, "Greedy construction of load-balanced virtual backbones in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 7, pp. 673–688, 2014.

[9] J. S. He, S. Ji, Y. Pan, and Z. Cai, "Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks," *Theoretical Computer Science*, vol. 507, pp. 2–16, 2013.

[10] J. Ben-Othman, K. Bessaoud, A. Bui, and L. Pilard, "Self-stabilizing algorithm for energy saving in wireless sensor networks," in *Symposium on Computers and Communications (ISCC)*. IEEE, 2011, pp. 68–73.

[11] D. Li, X. Jia, and H. Liu, "Energy efficient broadcast routing in static ad hoc wireless networks," *Transactions on Mobile Computing*, vol. 3, no. 2, pp. 144–151, 2004.

[12] S. Schmid and R. Wattenhofer, "Algorithmic models for sensor networks," in *20th International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2006, pp. 11–pp.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *33rd Annual Hawaii International Conference on System sciences*. IEEE, 2000, pp. 10–pp.

[14] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[15] N. Fan and J.-P. Watson, "Solving the connected dominating set problem and power dominating set problem by integer programming," in *International Conference on Combinatorial Optimization and Applications*. Springer, 2012, pp. 371–383.