A Topology Control Algorithm for Wireless Sensor Networks that considers Overhearing

Flávio M. Assis Silva and Ulisses Telemaco Neto
PPGM - Graduate Program on Mechatronics
LaSiD - Distributed Systems Laboratory
DCC - Department of Computer Science
UFBA - Federal University of Bahia
Salvador, Bahia, Brazil
Email: fassis@ufba.br, ulisses.telemaco@gmail.com

***Abstract*—Topology control is one of the main techniques that can be used to decrease energy spenditure in wireless sensor networks. Although it has been the subject of much research, less attention has been devoted to study the effects of *overhearing* on topology control, i.e., the effects of the cost implied by nodes hearing transmissions even if these transmissions were not intended to them. In this paper we describe a distributed (localized) algorithm for topology control in wireless sensor networks. Our approach differs from previous work mainly in the sense that it takes the effects of overhearing into consideration and that it might eliminate more communication links from a given connectivity graph, and thus possibly assign lower transmission power to some nodes. This is done by eliminating so-called k-redundant edges, instead of eliminating only two-redundant edges. We present the sketch of the proof of properties of the algorithm and simulation results.**

I. Introduction

One of the main aspects of the design of Wireless Sensor Networks (WSN) is the need for conserving the energy of nodes. Since nodes are typically battery-powered and WSNs are deployed in environments where replacing node batteries or even having physical access to the nodes are not possible or viable, the adoption of techniques that make a node spend as less energy as possible is of paramount importance.

One of the main techniques that have been proposed for conserving node energy is *topology control*. Topology control represents a set of techniques to assign an appropriate transmission power to each node of a network with the purpose of maintaining some property (e.g., connectivity) over the graph that represents the communication links between nodes, while minimizing energy consumption (and/or interference) that is (are) strictly related to the nodes' transmitting range [1]. Topology control is possible since radio devices used in nodes typically support transmissions with different power levels. As higher the transmission power, as higher the amount of energy spent during transmission.

Different approaches have been proposed to topology control in WSN (e.g., [1]–[4], to name a few). Of particular importance is the development of distributed *localized* algorithms for topology control (e.g., [2], [3], [5]–[8]). A topology control algorithm is localized if each node decides which transmission power to use based on local information (i.e., information about itself and its neighbours). Localized algorithms contribute to network scalability, a typical requirement of WSNs.

Although localized algorithms for topology control in WSNs have been the subject of much research, only few of them consider the effect of *overhearing* (e.g., [6], [9], [10]). According to frequently adopted energy models (e.g., [11]), each node spends an amount of energy during reception which is a constant value multiplied by the time the node remains at this state. Since wireless transmissions are performed by broadcasting, all nodes that are in the radio range of a transmitter will hear the transmission, even if they are not intended receivers of it (except in cases such as when nodes have their radio devices turned off). Although neglected in many previous works, overhearing has an impact on topology control [10].

In this paper, we describe an approach for topology control which takes overhearing into consideration. Our approach differs from previous work in the sense that it is a distributed localized algorithm that eliminates so-called *k-redundant edges* (as defined in [3] and explained in section III) that are detectable at the vicinity of each node. This means that more communication links between nodes might be detected as redundant, possibly allowing nodes to use a lower transmission power. We describe the proposed algorithm, present sketches of proofs of properties that are valid over the communication graph induced by the algorithm and we present a quantitative result based on simulation.

This paper is structured as follows. In Section II we describe the adopted system model. In Section III we formulate the topology control problem. In Section IV we describe our approach for topology control. In Section V we present the result of applying the algorithm to a simulated scenario. In Section VI we discuss related work. Finally, in Section VII we conclude the paper.

II. System Model

In this paper, a wireless sensor network consists of a set of n static nodes. We model the behaviour of each node by a *process* associated with the node. So we have n processes, $p_1, p_2, ..., p_n$, one for each node. Since there is a one-to-one correspondence between processes and nodes, we will use the terms *process* and *node* interchangeably.

We assume that each node can adjust its transmission power to any value between 0 and a certain maximum, referred to as M_{power}. This maximum power level is the same to all nodes. Varying the transmission power of a node might change its set of neighbours. We assume that there is a path between any pair of nodes (processes) in the network, if all nodes transmit at M_{power} (this property is formally presented in the next section).

We use the energy model described in [11]. According to this model, energy is spent by nodes during transmission, reception and during processing states. However, the energy spent during processing is neglected, since it is typically very low compared with the energy spent during transmission and reception. The energy spent to transmit is the energy spent to run the radio electronics and the power amplifier. Both are dependent on hardware characteristics, such as digital coding and modulation used. The energy spent to run the power amplifier is also dependent on the distance between the transmitter and receiver and is computed according to a specific power loss model.

Thus, the energy spent to transmit an l-bit message from process p to process q, denoted $E_{Tx}(l, p, q)$, is given by:

$$E_{Tx}(l, p, q) = l.E_{elec} + l.\epsilon.d^{\alpha}, \tag{1}$$

where: E_{elec} is the energy spent by the transmitter electronics; d is the distance (in meter) between p and q; α is the power loss exponent ($2 \leq \alpha < 4$); and ϵ is a parameter characteristic of the transceiver and the channel [12].

The energy spent to receive an l-bit message, denoted $E_{Rx}(l)$, is given by:

$$E_{Rx}(l) = l.E_{elec}, \tag{2}$$

where we assume that the energy spent to run the receiver electronics equals the energy used to run the transmitter electronics.

We assume additionally that each process knows its current geographic location (the nodes obtain this information from a positioning system, such as GPS, or by other means, such as using triangulation using some reference points in the network).

Communication channels are assumed to be *reliable*, i.e., a message sent by a process to another eventually arrives at its destination, uncorrupted and exactly once. Processes (and therefore nodes) do not fail. We assume additionally that the system is synchronous, i.e., there is a known upper bound on the time a message takes to arrive at its destination and to a process to execute each of its single steps.

We consider additionally that the nodes are distributed over a plane (i.e., the location of each node is given by a pair of x, y-coordinates) and that the radio range of nodes is circular (with the node at the center of the circle).

III. Topology Control

Sensor nodes' radio devices usually support adjustable transmission power. It is not energy-efficient for each node to transmit at its maximum power if they can transmit at a lower power level. *Topology Control* refers to the set of techniques that can be used to define a communication topology for a network (by choosing a specific transmission power level to each node) in such a way that a certain property on the resulting communication graph is valid (e.g., connectivity) while reducing energy consumption (and/or interference) [1].

Seeking an optimal power assignment for nodes of a sensor network such that the topology is connected (in the case of undirected graphs) or strongly connected (in the case of directed graphs) is NP-Complete [4]. Thus, many algorithms have been proposed to find power assignments which minimize total energy spenditure. In [3] the authors first described a distributed (localized) protocol to manage the transmission power of nodes in such a way that nodes can transmit with lower power and the induced communication graph has the *minimum-energy property*, as defined below. Our approach described in this paper follows similar principles.

Based on the notation introduced in [3] and [2], we use the directed graph $G_{max} = (V, E_{max})$ to represent the resulting communication network when all nodes transmit at maximum power. The set V represents the set of nodes (processes) and E_{max} represents communication links between the nodes. An edge $(p, q) \in E_{max}$ iff node q is in p's radio range when p transmits at maximum power. We assume that G_{max} is strongly connected. We want to find a subgraph $G_{min} = (V, E_{min})$ of G_{max} such that: (a) G_{min} and G_{max} contain the same set of nodes, V; (b) G_{min} has less edges than G_{max} (if possible); (c) G_{min} is strongly connected; and (d) G_{min} has the *minimum-energy property* [2], i.e., for each pair of nodes $u, v \in V$, there is a path from u to v in G_{min} that has the same cost as a minimum cost path between them in G_{max} (considering the cost of edges as defined in (3), in section IV-A). Thus, G_{min} is a subgraph of G_{max} that is still strongly connected, might have less edges than G_{max}, and preserves the cost of minimum cost paths. G_{min} can be used instead of G_{max}, for example, for finding minimum cost paths between all pair of nodes of the network with a potential decrease in the total amount of energy used (e.g. [3]).

In particular, the topology control problem addressed in this paper is the problem of finding a G_{min} subgraph of a given G_{max} graph, with the properties above, by removing so-called *redundant edges* that are locally detectable (i.e., at the vicinity of each node), considering the cost of overhearing. An edge (p, q) is *k-redundant* ($k \geq 2$) iff there is a path with length k (i.e., with k edges), $\langle p, n_1, n_2, ..., n_{k-1}, q\rangle$, such that sending a message from p to q along this path has a lower cost (i.e., it results in less energy being spent) than sending the message directly from p to q.

IV. A Topology Control Algorithm

A. Algorithm Overview

Our approach follows general steps similar to algorithms described in previous work [5], [6]. We, however, take into consideration the cost of overhearing and eliminate (locally detectable) k-redundant edges. The algorithm is fully distributed and localized, in the sense that the actions taken by each node depends solely on information about itself and its neighbours.

The algorithm finds the transmission power of a node (process) by finding this node's (process') *reduced set of neighbours*. A node q belongs to a node p's reduced set of neighbours iff q is one of p' s neighbours and the edge (p, q) is not k-redundant, considering p's local information. In other words, a node q belongs to a node p's reduced set of neighbours iff sending a message directly from p to q results in less energy being spent than sending the message through any other path composed by p's neighbours.

Each process executes the same algorithm. The algorithm has three phases. In the first phase, each process broadcasts its id and its x, y-coordinates at full transmission power. After this phase, each process will know the id and position of all of its neighbours.

In the second phase, each process broadcasts a message with the id and x, y-coordinates of itself and of all of its neighbours. After having received these messages from all its neighbours, each process will build a directed graph. The (directed) graph built by process p will be referred to as $G_p^c = (V_p^c, E_p^c)$. The set V_p^c will contain p, p's neighbours and the neighbours of p's neighbours. The set of edges E_p^c will reflect the neighbourhood relationship, i.e., an edge (r, s) will be in E_p^c iff s is a neighbour of r.

In the third phase, each process calculates the shortest cost paths from itself to each of its neighbours. The cost of each edge, however, depends on the distance between the nodes (the power used to transmit the message) and the number of nodes that hear the transmission. The cost of an edge (p, q), i.e., the cost implied when process p sends a message to q (independently of the size of the message), denoted by $Cost(p, q)$, is given by:

$$Cost(p, q) = E_{elec} + \epsilon.d^{\alpha} + E_{elec}.h, \tag{3}$$

where E_{elec}, ϵ and α have the same meanings as described in section II, d is the distance between p and q, and h is the number of p's neighbours whose distances to p are less or equal to d (i.e., h represents the number of p's neighbours which would hear a transmission from p to q). Observe that the cost associated with a (p, q) edge is the cost of transmission ($E_{elec} + \epsilon.d^{\alpha}$) added to the cost of reception ($E_{elec}.h$).

For a given process p, its reduced set of neighbours will be the set of its neighbours q for which the path $\langle p, q\rangle$ (i.e., the direct path from p to q) is a minimum cost path from p to q (according to p's local knowledge).

```
FindReducedSetOfNeighbours

01   $RNbrs_p \leftarrow \emptyset \quad Pos_p \leftarrow \langle p, \langle x_p, y_p \rangle\rangle$
02   broadcast $\langle Pos_p \rangle$                              {Phase 1}
03   wait until $timeout(\Delta)$                            {Phase 2}
04   broadcast $\langle RNbrs_p, Pos_p \rangle$
05   wait until $timeout(\Delta)$                            {Phase 3}
06   $G_p^c \leftarrow$ buildConnectivityGraph($RNbrs_p$)
07   $G_p^{min} \leftarrow$ findMinimumCostPaths($p, G_p^c, Pos_p$)
08   $Nbrs_p \leftarrow \{q : q \in V$ and $(p, q) \in E_p^{min}\}$
09   return $Nbrs_p$

10   on receive $\langle Pos_q \rangle$ from $q$ do
11       $Pos_p \leftarrow Pos_p \cup Pos_q$
12       $RNbrs_p \leftarrow RNbrs_p \cup (p, q)$

13   on receive $\langle RNbrs_q, Pos_q \rangle$ from $q$ do
14       $Pos_p \leftarrow Pos_p \cup Pos_q$
15       $RNbrs_p \leftarrow RNbrs_p \cup RNbrs_q$
```

Fig. 1. Algorithm *FindReducedSetOfNeighbours* executed by process p to find its optimized set of neighbours

This approach eliminates k-redundant edges that are locally detectable, i.e., it eliminates an edge (p, q) if there is a path $\langle p, r_1, r_2, ..., q\rangle$ of length k ($k \geq 2$) from p to q, whose nodes are p and its neighbours, and whose cost is less than the cost of (p, q). This is true, since each process p calculates the shortest cost paths in G_p^c and p will only consider a process q as a member of its reduced set of neighbours if (p, q) is a minimum cost path between p and q. If the cost of the path $\langle p, r_1, r_2, ..., q\rangle$ is lower than the cost of (p, q), then p will have node r_1 as a candidate member of its reduced set of neighbours, and not q.

B. Detailed Description

The algorithm executed by a process p is shown in Fig.1 (algorithm *FindReducedSetOfNeighbours*). It uses the following variables:

1) $RNbrs_p$: a set of (r, s) edges. If $(r, s) \in RNbrs_p$, then p knows that process r can communicate with process s in a single hop when full transmission power is used;
2) Pos_p: a set of 2-tuples $\langle q, \langle x_q, y_q\rangle\rangle$, where x_q and y_q are the Euclidian x, y-coordinates of process q. Pos_p represents the set of coordinates of processes that are known by process p;
3) Δ: a period of time. It is assumed that Δ is long enough for a process to receive the messages broadcast by all its neighbours in phases 1 and 2 (lines 2 and 4 in Fig.1);
4) $G_p^c = (V_p^c, E_p^c)$: the connectivity graph, as known by process p. The V_p^c set includes process p, its neighbours (when transmitting at full power), and the neighbours of its neighbours (when transmitting at full power). I.e.:

$$V_p^c = \{p\} \cup \{q : q \in V \wedge (((p, q) \in E_{max}) \vee (\exists\ r : r \in V \wedge (p, r) \in E_{max} \wedge (r, q) \in E_{max}))\}$$

and

$$E_p^c = \{(p,q) : q \in V_p^c \wedge (p,q) \in E_{max}\} \cup \{(r,s) : r \in V_p^c \wedge s \in V_p^c \wedge (p,r) \in E_{max} \wedge (r,s) \in E_{max}\}$$

5) $G_p^{min} = (V_p^{min}, E_p^{min})$: the directed graph which contains minimum cost paths between node p and its direct neighbours. I.e.,

$$V_p^{min} = \{p\} \cup \{q : q \in V \wedge (p,q) \in E_{max}\}$$

and

$$E_p^{min} = \{(q,r) : q \in V_p^{min} \wedge r \in V_p^{min} \wedge (q,r) \in E_p^c \wedge (\exists s : s \in V_p^{min} \wedge (q,r) \text{ is in a minimum cost path from } p \text{ to } s)\}$$

6) $Nbrs_p$: p's reduced set of neighbours. This is the set returned by *FindReducedSetOfNeighbours*

When the algorithm begins, the set $RNbrs_p$ is empty and the set Pos_p contains only information about p's x,y-coordinates (line 1, Fig.1).

In the first phase of the algorithm, process p broadcasts a message containing its id and its x,y-coordinates (line 2, Fig.1).

In the second phase (lines 3 and 4, Fig.1), process p waits until all its neighbours send their respective ids and x,y-coordinates (line 3, Fig.1). As described above, we assume a period Δ which is enough for all processes to receive this message from all its neighbours. When process p receives a $\langle q, (x_q, y_q)\rangle$ message from process q, it adds the coordinates of process q to the Pos_p set and it adds (p,q) to the $RNbrs_p$ set (lines 10-12, Fig.1). The pair (p,q) represents the fact that q is one of p's neighbours. After having received the messages from all its neighbours, process p broadcasts its set of neighbours with their respective ids and coordinates (line 4, Fig.1).

In phase 3 (lines 5 to 9, Fig.1), process p first waits until it receives the messages with neighbours data (id and position) from all of its own neighbours (line 5, Fig.1). As described previously, we assume the time period Δ to be enough for that. When process p receives a $\langle RNbrs_q, Pos_q\rangle$ message from process q, it updates its $RNbrs_p$ and Pos_p sets (lines 13-15, Fig.1). I.e., it includes the information about neighbourhood of processes in $RNbrs_p$ and updates its knowledge about the coordinates of processes in its surrounding.

After having received the location information from all of its neighbours, process p calculates minimum cost paths from itself to each of its neighbours, but taking the cost of overhearing into consideration (i.e., each edge will have the cost as specified in (3) - see Section IV-A). G_p^{min} is the resulting graph (i.e., the graph with such paths) (line 7, Fig.1). The set $Nbrs_p$ will be the set of processes which are direct neighbours of p in G_p^{min} (line 8, Fig.1). This set is returned by the algorithm (line 9, Fig.1).

Let $G_p^R = (V_p^R, E_p^R)$ be the directed graph that represents the relationship between a process p and its reduced set of neighbours. I.e., $V_p^R = \{p\} \cup Nbrs_p$ and $E_p^R = \{(p,q) : q \in Nbrs_p\}$.

The $G_{min} = (V, E_{min})$ graph (see section III) will be the directed graph induced by the G_p^R directed graph of all nodes. I.e., V is the set of all nodes and the set E_{min} is defined as:

$$E_{min} = \bigcup_{\forall p \in V} E_p^R$$

We impose as a requirement to the minimum cost path algorithm that, if the edge (p,q) is a minimum cost path between node p and some neighbour q, this edge is returned as the minimum cost path between these nodes, instead of any other longer path with the same cost that might exist. I.e., the algorithm prefers one-edge paths instead of longer paths with the same cost.

C. Proof of Properties

In this section, we present a sketch of the proof of the following properties of the algorithm: (a) if G_{max} is connected, then G_{min} is connected as well (i.e., the graph induced by the algorithm maintains connectivity); and (b) G_{min} has the *minimum-energy property*. This property expresses the fact that minimum cost paths existing in the original graph are not eliminated by the algorithm (i.e., they continue to exist in G_{min}).

First, we introduce the G_w graph. Let $G_w = (V, E_w)$ be a weighted directed graph, such that: (a) G_w and G_{max} have both the same set of nodes (V); (b) G_w and G_{max} have both the same set of edges (i.e., $E_w = E_{max}$); and (c) each edge (p,q) in E_w has a cost, denoted $Cost_w(p,q)$. For all edges (p,q), $Cost_w(p,q) = Cost(p,q)$, as defined in (3) (see Section IV-A).

Since G_w and G_{max} have the same set of nodes and edges, if G_{max} is strongly connected then G_w is strongly connected as well.

Theorem 1. *If G_{max} is strongly connected, then G_{min} is strongly connected as well.*

Sketch of Proof: Since G_{max} is strongly connected, there is a path between every pair of nodes. Let us consider two nodes, p and q ($p,q \in V$). Let us denote by $\langle r_1, r_2, ..., r_m\rangle$, $m \geq 2$, $r_1 = p$, $r_m = q$ a minimum cost path between these two nodes.

First we prove that $(r_i, r_{i+1}) \in E_{r_i}^R$, $\forall i : 1 \leq i < m$. Observe that the edges belonging to $E_{r_i}^c$ are all in E_w (i.e., $(e \in E_{r_i}^c) \Rightarrow (e \in E_w)$) and $Cost_w(p,q)$ was the cost used at node r_i to find minimum cost paths, since this node has knowledge about all of its neighbours. Since edge (r_i, r_{i+1}) belongs to a minimum cost path in G_{max}, then it is a minimum cost path between nodes r_i and r_{i+1}. Since $G_{r_i}^R$ is generated by an algorithm that finds minimum cost paths, (r_i, r_{i+1}) must belong to $E_{r_i}^R$.

Since all edges belonging to a minimum cost path between nodes p and q belong to E_s^R, for some node s, there will be a path between p and q in G_{min} (since E_{min} is the union of the

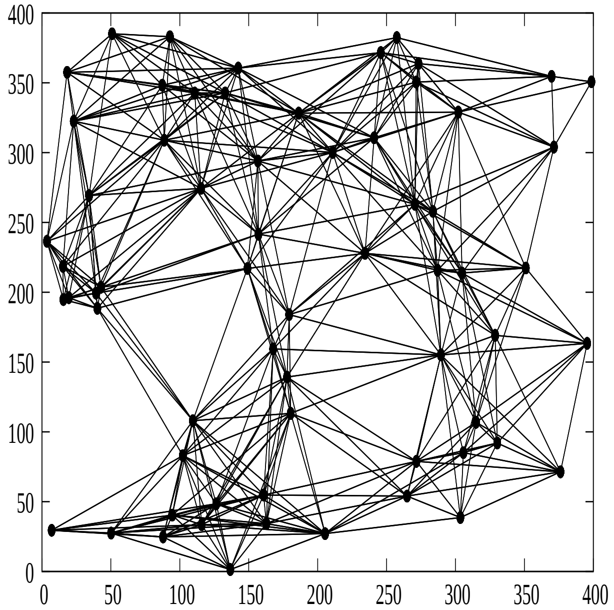

Fig. 2. Example of connectivity graph (all nodes transmit at full power): 64 nodes over a 400x400 region

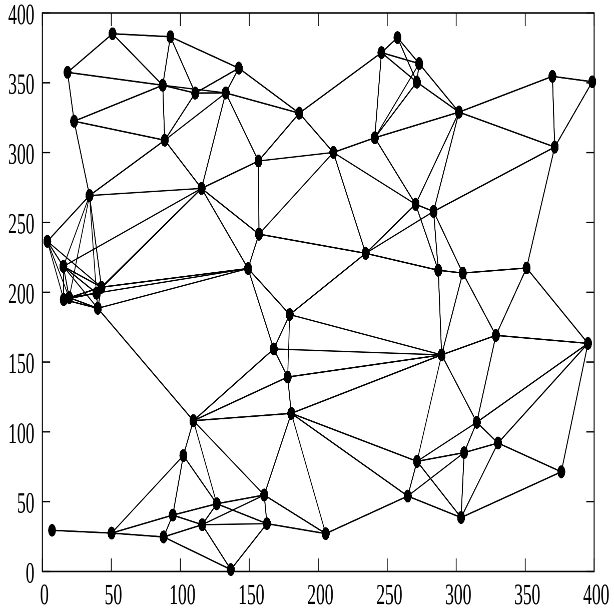

Fig. 3. Graph induced by the algorithm, corresponding to the graph of Fig.2

edges of all E_s^R sets - see previous section). Since it is valid for all pair of nodes p and q, G_{min} is strongly connected. ■

Theorem 2. *Let c be the cost of a minimum cost path between any two nodes, p and q, in G_w ($p, q \in V$). There is a path in G_{min} between p and q with cost c.*

Sketch of Proof: Using the same arguments of the proof of theorem 1, all edges belonging to a minimum cost path between p and q in G_w will be in E_{min}. So, a minimum cost path in G_w will also exist in G_{min}. Since the cost of edges will be the same in G_w and G_r^{min}, for all nodes r, the minimum cost path in G_{min} will have the same cost, c. ■

V. Evaluation

In this section we illustrate the result of applying the algorithm described in the previous section to a simulated scenario. The simulation was done using the ns-2 simulator [13]. The simulation scenario consists of 64 nodes randomly spread over a 400x400 region. The maximum radio range of each node is approximately 100 distance units. We used the radio parameters presented in [11]: the energy dissipated by transmitter and receiver electronics was assumed to be 50nJ/bit (E_{elec} in equations (1), (2) and (3)) and the energy dissipated by the transmit amplifier was assumed to be 100pJ/bit/m^2 (ϵ in equations (1) and (3)). We assumed additionally a path loss exponent equal to 2.

Fig.2 shows the connectivity graph for the generated network, when all nodes transmit at full power. In order to make the presentation clearer, we do not show the orientation of edges. Since we assume that communication is simmetric when all nodes transmit at full power, each edge in the figure represents two directed edges (one for each direction). In the discussion of the previous sections, this graph was represented by G_{max}.

Fig.3 shows the corresponding G_{min} graph, after running the algorithm. For simplicity, we again present the edges as undirected. Each edge between two processes p and q in the figure corresponds either to a (p, q) edge, a (q, p) edge or to both.

When we compare both figures, we see that many edges were eliminated from the graph of Fig.2. As an example of a node which had its set of neighbours reduced, see the node at coordinates $(7, 30)$ (the node that is closest to the origin of the axes). In the graph of Fig.2, this node has six neighbour nodes (nodes in the following coordinates: $(127, 49)$, $(116, 34)$, $(102, 83)$, $(95, 40)$, $(88, 25)$, and $(50, 28)$). After running the algorithm, this node has only the node at coordinates $(50, 28)$ as its neighbour, its closest one (see Fig.3). The edges to the other neighbours are redundant. Thus, this node only has to use enough transmission power to reach its closest neighbour.

If we assign to each node the power needed to reach the farthest node in its reduced set of neighbours, i.e., if each node uses this power to transmit messages, we achieve a power assignment to nodes that: mantains connectivity of the graph; and that results in less energy being spent by nodes during (local) broadcasts than assigning the maximum transmission power to them.

VI. Related Work

Topology control has been studied in the context of static networks as well as of networks with mobile nodes (e.g., [1]–[3], [5]–[10], just to name a few). Finding an optimal mapping of transmission power to nodes in such a way that the overall energy consumption on the network is minimized was proven to be an NP-Complete problem [4]. In [3] the authors proposed a distributed algorithm run by each node of a WSN that defines a transmission power to each node in such a way that the induced graph has the *minimum-energy property* (see Section

III). After this work, other researchers have proposed different approaches to this problem [2], [7], [8] (to name a few).

The effect of overhearing, however, was not considered in these works ([2], [3], [7], [8]), although the authors recognize its potential effect on power assignment (e.g., [3]). Additionally, they aim at eliminating two-redundant edges (i.e., k-redundant edges for $k = 2$). In our approach, we consider the effect of overhearing and we eliminate all k-redundant edges (i.e., not only two-redundant edges) that are detectable locally (i.e., in the radio range of a node).

The general structure of the algorithm described in this paper is similar, in some aspects, to the algorithms presented in [5] and [6]. In these works and in our algorithm, each node makes its decision based on a graph constructed with information about its neighbours and the neighbours of its neighbours. However, our approach differs from them in the following aspects: in [6] the objective of the algorithm is to eliminate two-redundant edges; and in [5] the authors do not consider overhearing.

The effect of overhearing is also considered in [9] and [10]. The authors of these papers, however, did not have as objective to propose a distributed algorithm for power assignment.

VII. Conclusion

This paper describes an approach to topology control in wireless sensor networks. We considered networks of static nodes only. Although the general idea of our approach was also used in previous work ([5], [6]), our approach differs from them in certain aspects. The main difference comes from the fact that we take the effect of overhearing into consideration and that our approach is a distributed algorithm which eliminates k-redundant edges which are detectable locally. Previous work either focused on eliminating only two-redundant edges, ignored the effect of overhearing or did not provide a distributed algorithm.

Beyond providing a sketch of the proof of the main properties of the algorithm, we present a result of simulation, using ns-2. The scenario simulated represents an example where the algorithm is effective in eliminating redundant edges from the original G_{max} graph. The algorithm can be used as a means to reduce the amount of energy spent by each node not only by reducing its transmission power, but also by reducing the number of its neighbours (what is useful when executing algorithms such as flooding, for example, over the network).

We assumed a failure-free system model (nodes and communication channels were assumed to be reliable). The occurrence of failures has an impact on some aspects of the algorithm, such as on the detection of the neighbours of a node. Topology control in such a more challenging scenario is the subject of current research.

References

[1] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, jun 2005.

[2] L. Li and J. Y. Halpern, “A minimum-energy path-preserving topology-control algorithm,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, pp. 910–921, may 2004.

[3] V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, aug 1999.

[4] X.Cheng, B.Narahari, R.Simha, M.X.Cheng, and D.Liu, “Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 248–256, 2003.

[5] S. Hong, Y.-J. Choi, and S.-J. Kim, “An energy efficient topology control protocol in wireless sensor networks,” in *Proceedings of the ICACT2007*, feb 2007.

[6] B. H. Liu, Y. Gao, C. T. Chou, and S. Jha, “An energy efficient select optimal neighbor protocol for wireless ad hoc networks,” Network Research Laboratory, School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia, Tech. Rep. UNSW-CSE-TR-0431, oct 2004.

[7] A. Rahman and P. Gburzinski, “On constructing minimum-energy path-preserving graphs for ad-hoc wireless networks,” in *Proceedings of the ICC’ 05*, South Korea, 2005.

[8] X.-Y.Li and P.-J.Wan, “Constructing minimum energy mobile wireless networks,” in *Proceedings of the MobiHOC 2001*, California, USA, 2001, pp. 283–286.

[9] P. Basu and J. Redi, “Effect of overhearing transmissions on energy efficiency in dense sensor networks,” in *Proceedings of IPSN’04*, Berkeley, California, USA, apr 2004.

[10] S. Gobriel, R. Melhem, and D. Mossé, “A unified interference/collision model for optimal mac transmission power in adhoc networks,” *International Journal of Wireless and Mobile Computing*, vol. 1, no. 3/4, pp. 179–190, 2006.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, oct 2002.

[12] T.S.Rappaport, *Wireless Communications: Principles and Practice*. Prentice-Hall, 1996.

[13] K. Fall and K. Varadhan, Eds., *The ns Manual*. The Vint Project, mar 2008.