

Local, Distributed Topology Control for Large-Scale Wireless Ad-Hoc Networks

Tim Nieberg, Johann Hurink

Department of Electrical Engineering, Mathematics & Computer Science

University of Twente

Postbus 217

NL-7500 AE Enschede

Email: {T.Nieberg, J.L.Hurink}@utwente.nl

Abstract—In this document, topology control of a large-scale, wireless network by a distributed algorithm that uses only locally available information is presented. Topology control algorithms adjust the transmission power of wireless nodes to create a desired topology. The algorithm, named Local Power Adjustment (LPA), does not rely on information about a node's position, and uses only two local broadcast messages per node to complete. The two objectives of the algorithm are minimizing the maximal and average power used by the nodes for the wireless transmissions. Additionally, the constraint of keeping the entire network connected by bidirectional communication links is imposed.

I. INTRODUCTION

This paper proposes a local algorithm for topology control of large-scale, wireless ad-hoc networks. This distributed algorithm only assumes knowledge about the direct neighborhood of each node. It creates a topology that remains globally connected and reduces the transmission power levels of the nodes in the network. The objectives are to minimize the signal strength of each node, both in terms of average and maximum power of the nodes in the network.

An ad-hoc network is created by wireless communication links between a set of radio transceivers. For a node to reach a neighboring node by direct transmission, a certain signal strength is needed. This setting can be represented by a communication graph whose edges are weighted according to the power needed to establish communication links. Each radio is able to adjust the signal strength which is related to the energy needed for the transmission of messages.

In the algorithm, each node sends exactly two messages, one control message of fixed length and a second message whose length is bounded by the size of the neighborhood. With the control messages, special nodes are recognized in the network that process the locally available information and compute the local topology according to a minimum spanning tree.

With topology control by power adjustment, there are several tradeoffs involved. Reducing the signal strength is one means to reduce the overall energy consumption. There are other benefits, especially in spatially dense networks such as the envisioned setting of wireless sensor networks. Fewer

direct neighbors result in less interference. Also, in a TDMA-based MAC scheme with a fixed number of scheduled slots per frame, the number of neighbors a node can communicate with is bounded, requiring a mechanism for topology control. On the other hand, reduced transmission ranges account for more intermediate hops of messages that have to be routed through the multi-hop topology of an ad-hoc network. This yields an increase in the end-to-end delay.

Theoretically, the power levels for nodes to reach one another are given as a function of the physical distance between these nodes. This assumption holds for ideal settings, e.g. an open field without any obstacles. In less than ideal settings, such as an indoor environment, the distance for a signal to travel depends on many factors including reflection and interference from walls and objects, direction of the antenna etc. In the model and therefore in the algorithm presented in the following, we do not assume a node's location to be known by it. Only information about the neighbors and the signal strength needed to reach these is required. This data can be obtained for example by a transceiver equipped with RSSI (Received Signal Strength Indicator) functionality or other neighborhood detection schemes with varying transmission power settings.

Recently, several distributed algorithms for topology control have been proposed [5], [8], [13]. However, these algorithms assume knowledge about the position of the nodes in the euclidean space, for example coming from a GPS-receiver, or directional antennas, and use techniques from computational geometry. For a survey on topology control for the area of routing in ad-hoc network, we refer the reader to [11]. Furthermore, several papers deal with the tradeoffs involved with the energy consumption due to transmission of packets, e.g. [6], [10], [14].

Adjusting the transmission power to minimize the maximum power used by the nodes in a wireless network has been studied in [12]. The presented algorithm implicitly constructs a minimum spanning tree in the communication graph. However, for the distributed setting, a link-state approach is taken assuming knowledge about global connectivity in each node. A local approach, adjusting the signal strength to reach a certain number of neighbors is presented and evaluated in [7]. It is stated that this approach results in sufficiently connected networks, even though not for all graphs.

This work has been partially supported by the European research project EYES (IST-2001-34734), [3].

In [9] a notion towards a general taxonomy of power level assignment problems is introduced by the class of graphs satisfying certain monotone properties to characterize feasible solutions. The problem of minimizing the average power, even when restricted to weights corresponding to distances in the two-dimensional euclidean plane is \mathcal{NP} -hard, for higher dimension it is even known to be \mathcal{APX} -complete [2].

This paper is organized as follows. Section II and III provide the graph model corresponding to the communication network and the definitions of the two problems discussed in the paper, as well as some properties of optimal solutions and approximations with respect to a minimum spanning tree. Section IV introduces the LPA and shows how a local spanning tree is constructed to reduce the power levels of the nodes. In Section V, we prove the correctness of the local algorithm. We conclude with results of several simulations of the presented algorithm in Section VI, and an outlook on future work.

II. DEFINITIONS AND PRELIMINARIES

A wireless, ad-hoc network is modeled as a directed graph $G = (V, A)$, where $V = \{1, \dots, n\}$ is the set of nodes equipped with processing capabilities and a radio-transceiver. The set of directed arcs $A \subset V \times V$ represents the possible communication between the nodes. There is an arc $(u, v) \in A$ iff node u can transmit to a node v by direct transmission. Each arc $(u, v) \in A$ is weighted with the minimal power $p(u, v) > 0$ that is required by node u to reach v . We assume that the weights are symmetric, i.e. $p(u, v) = p(v, u)$ for all $u, v \in V$.

Each node's transceiver can adjust the signal strength for sending messages to neighboring nodes. Denote by $P := \{P_u \mid u \in V\}$ an assignment of transmission power levels to the nodes in the network. These power levels induce a graph $G_P := (V, A_P)$, where $(u, v) \in A_P$ iff $P_u \geq p(u, v)$. In the following, we refer to $P := \{P_u \mid u \in V\}$, and the resulting directed graph as a solution (with respect to) P .

If two nodes can reach one another in the graph resulting from a set of power levels, we say that there is a bidirectional link between them. Such a bidirectional link between nodes $u, v \in V$ is denoted by $[u, v]$ (i.e. $[u, v] = \{(u, v), (v, u)\}$). Clearly, $[u, v]$ exists whenever $(u, v) \in A_P$ and $(v, u) \in A_P$. This allows us to treat the induced bidirectional links as an undirected communication (sub-)graph. In the following, a solution is called *feasible* if the induced bidirectional subgraph is connected.

We assume that the maximum sending strength of each node is limited by P_{\max} . When setting the power levels of each node to P_{\max} , we obtain the maximal number of arcs in the network realizable by the radio-transceivers. It is referred to as the *maximal graph*. For the remainder, we assume w.l.o.g. that each arc's weight is less than or equal to P_{\max} , i.e. $G = (V, A)$ represents the maximal graph. Furthermore, we assume that the graph G is connected.

For a node $i \in V$, let N_i denote its direct neighborhood, i.e. all nodes that can be reached by a direct transmission sent by node i . For topology control algorithms, the neighborhood N_i depends on the power that node i is transmitting with.

Unless stated otherwise, we assume this to be P_{\max} . In this case, N_i is the set of the maximally reachable nodes surrounding node i . This characterization of a neighborhood is also used to characterize a local algorithm: A *local* algorithm runs decentralized in the nodes and interacts with the direct neighbors only.

Our local algorithm uses only locally available information about the neighborhood of the nodes. In a local algorithm, no node has complete knowledge of the entire network. It is easy to see that a local algorithm scales well with the network size.

In this paper, we consider two closely related problems on the communication graph that allow for reducing the signal strength.

Problem 1 (Maximal Power Adjustment): For an ad-hoc network, given by the maximal graph $G_{\max} := (V, A)$, find a feasible solution so that the maximum signal strength, $\max_{i \in V} P_i$, is minimized.

By looking at the maximum power used by a node in the network, the time to the first node death is minimized.

Problem 2 (Total Power Adjustment): For an ad-hoc network, given by the maximal graph $G_{\max} := (V, A)$, find a feasible solution so that the sum of the power levels, $\sum_{i \in V} P_i$, is minimized.

The latter problem is equivalent to minimizing the average signal power of the nodes in the network. Note that any feasible solution results in a valid assignment for both problems.

III. CENTRALIZED SOLUTIONS BASED ON SPANNING TREES

In this section, we show that the two considered problems are closely related to the minimum spanning tree (MST) problem. A minimum spanning tree is defined as the cheapest subset of edges that keeps the graph connected.

For a given (minimum) spanning tree T for the network, the power levels of all nodes $u \in V$ may be set to $P_u := \max_{\{(u, v) \in T\}} p(u, v)$. Furtheron, let $P^T := \{P_u\}_{u \in V}$ denote the solution obtained according to a tree T as described above. Obviously, P^T is feasible, as the bidirectional edges of the tree T are included. For the objectives, we state the following two theorems.

Theorem 1 ([12]): For a wireless network $G = (V, A)$, let T be a minimum spanning tree. The corresponding solution P^T is optimal for the Maximal Power Adjustment Problem.

In contrast, the Total Power Adjustment Problem is \mathcal{NP} -hard [2]. Therefore it is unlikely to optimally solve this problem by an algorithm similar to a greedy strategy for obtaining an MST. However, the solution resulting from an MST yields a 2-approximation of the optimal solution to the Total Power Adjustment Problem in arbitrary graphs:

Theorem 2 ([6]): Let T be a minimum spanning tree in the maximal graph, and let

- ϕ_{MST} be the value of T , i.e. the sum of the weights of the edges in T ,
- ϕ_T be the total power assignment of the solution P^T , i.e. $\phi_T = \sum_{u=1}^n \max_{\{(u, v) \in T\}} p(u, v)$, and
- ϕ_{OPT} be the objective value of an optimal solution to the Total Power Assignment Problem.

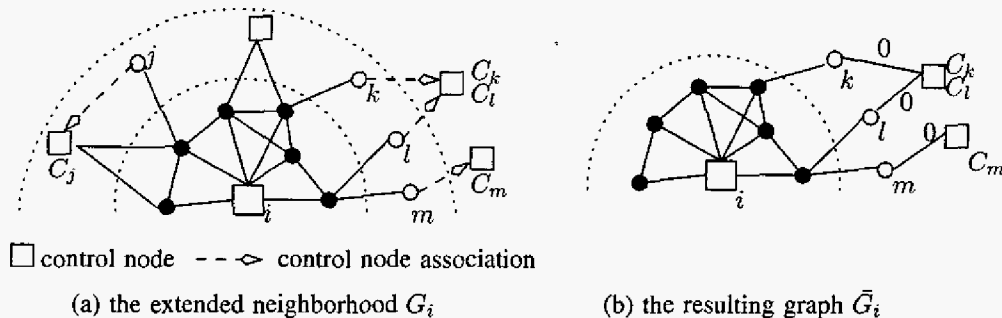


Fig. 1. The extended neighborhood for control node i to construct T

Then $\phi_{MST} < \phi_{OPT} \leq \phi_T < 2 \cdot \phi_{MST}$.

Furtheron, following the above two theorems, we focus on constructing an appropriate minimum spanning tree. A standard algorithm to construct an MST in a weighted graph is known as Prim's Algorithm. The algorithm starts with an arbitrary node of the graph and iteratively adds the closest node to the connected subtree until it spans the entire set of nodes [1].

For the distributed construction of an MST in a communication graph, at least $\Omega(|A| + n \cdot \log(n))$ messages are required and then messages have a length of $O(n)$. The asymptotic time complexity bound of the distributed construction of an MST is $\Omega(n)$ assuming that each message delivery takes one time unit [4].

However, in the setting of a possibly large-scale ad-hoc network, the exchange and storage of large amounts of data is prohibitive. We therefore propose a local algorithm to heuristically create a connected graph that locally resembles an MST.

IV. LOCAL, DISTRIBUTED MST-HEURISTIC

The local, distributed algorithm for building up an MST presented in this section works in two stages. First, some nodes locally elect themselves to become control nodes which are then surrounded by regular nodes, and second, the control nodes perform the necessary calculations to adjust the power levels of their neighbors. Furtheron, we call this algorithm *LPA* (Local Power Adjustment).

Assume that each node has a unique number, and that each node knows the numbers of the nodes within its neighborhood. Initially, each node is considered *undecided*, and during the first stage of the algorithm, it changes its status either to *control* or *regular*. Let s_i denote the status of each node $i \in V$. Each change of a node's status results in a respective broadcast to inform all nodes in the local neighborhood of the change. As each change in the structure is witnessed by a message, the algorithm can be realized event-driven once invoked.

The decision of changing the state of an undecided node is based on the numbers of the nodes $V = \{1, \dots, n\}$. An undecided node $i \in V$ that has the highest node number among all undecided nodes in its neighborhood is able to leave the undecided status:

- if there are no control nodes in the neighborhood N_i of i , the node changes its status to become a control node;

- if there exists a control node in the neighborhood N_i , node i becomes a regular node and associates with the control node that has the highest node number in the neighborhood. For the notation, let the formal association of a regular node $i \in V$ with a control node $u \in N_i$ be given by C_i , i.e. in this case $C_i = u$.

When a node has left the undecided status, it enters the second stage of the algorithm. A regular node $i \in V$ waits until all neighboring nodes have left the undecided state. It then knows from each neighboring node $u \in N_i$ both the status s_u and the associated control node C_u . This information, together with the transmission power needed to reach each neighbor, is passed on to all neighboring control nodes.

Based on this received neighborhood information, a control node knows its local topology, can construct a minimum spanning tree therein, and broadcast this spanning tree to all neighboring nodes. A regular node that receives spanning tree information adjusts its transmission power so that the neighbors of it in the tree are included in the resulting neighborhood. It remains to describe the construction of the local spanning tree in more detail.

The overall goal in the construction of the spanning tree for the one-hop topology of a control node is to ensure that surrounding control nodes can be reached by a bidirectional path after the reduction of the transmission power levels.

Once a control node $i \in V$ has received the neighborhood information from all nodes in N_i , it knows the complete local topology, the two-hop neighbors and their controlling nodes. The local topology is stored in a graph $G_i \subset G_{max}$ which is reduced to an induced graph \bar{G}_i before a minimum spanning tree T is computed.

The set of nodes in \bar{G}_i consists of i and all one-hop neighbors. Additionally, surrounding control nodes that are not in the second order neighborhood need to be accounted for. Therefore, all two-hop neighbors whose control node is *not* in the second order neighborhood are included, as well as an artificial node for the controlling instance at distance 0. The edges of \bar{G}_i are all edges G_i of which the two end nodes belong to \bar{G}_i .

Figure 1 gives an example of the reduction process. In 1(a), the graph G_i with all the information available at i is given. This is then reduced to \bar{G}_i in 1(b). Note that the 2-hop neighbor j is not included in \bar{G}_i as its main controlling node is also a 2-hop neighbor of node i .

Finally, node i calculates a minimum spanning tree in the extended neighborhood \bar{G}_i , and all surrounding nodes are informed about this tree in order to adjust their transmission power accordingly.

V. DISCUSSION AND PROPERTIES

In this section, we show that the LPA terminates, and that the graph $G_P = (V, A_P)$ resulting from the adjustment of the power levels of each node according to the local spanning trees constructed in the control nodes is globally connected.

As all nodes send at full signal strength during the execution of the algorithm, the following properties all refer to the maximal graph $G = (V, A)$. Let a *step* of the algorithm denote the finite time between the change of a node's status and all neighbors being informed about his change.

First, we show that each node reaches the second stage of the algorithm by deciding on its role.

Lemma 3: After at most n steps, each node has decided on its status, being either Control or Regular.

Proof: Let $U_k \subset V$ be the set of nodes which are undecided after step k of the LPA algorithm. Initially, we have $U_0 = V$. Furthermore, in step $k + 1$ at least the node with largest number in U_k changes its status. Since a node that has left the undecided state does not change its status any further, this implies that after step k , at least k nodes have decided on their status. \square

Next, we show two simple properties of the structure evolving from the decision process of the nodes.

Lemma 4: No two control nodes are direct neighbors.

Proof: Let $u, v \in V$ be two neighboring nodes and let $u > v$. Obviously, u decides on its status before v . Thus, v has already a control node in its neighborhood when it has to decide on its status, and therefore cannot become a control node. \square

Lemma 5: For each regular node $u \in V$, the node stored in C_u is the neighboring control node with the highest number.

Proof: When u decides on its status, it has a larger number than all its undecided neighbors, and in its neighborhood already one node is a control node. Furthermore, all decided nodes in the neighborhood have a higher number than u (otherwise they would not have been able to decide on their status). Thus, the chosen control node C_u is the control node with highest number in the neighborhood of u . \square

Note that the above Lemmas also prove that the LPA corresponds to a greedy strategy for calculating a *Maximum Weight Independent Set* formed by the set of control nodes, where the weight is given by the node number, implying that no further node can be added without violating the independence property.

The following Lemma concludes that the LPA algorithm terminates in finite time.

Lemma 6: After a node $u \in V$ has decided on its status, it sends one more message.

Proof: First, consider a regular node u . Each time, upon the reception of a message from a neighboring node, u checks if every node in the neighborhood has decided on its status. If and only if this is the case, it broadcasts its

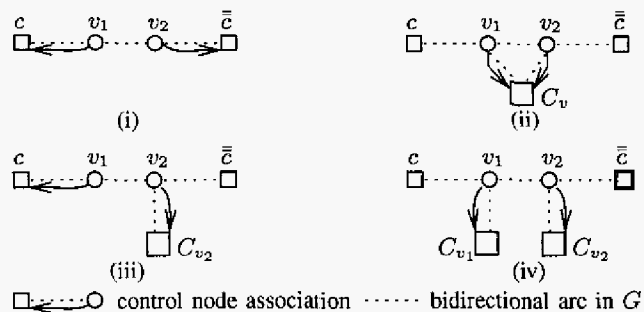


Fig. 2. Cases for proof of Theorem 7

neighborhood information to all neighbors and further on only waits for messages from the surrounding controlling instances containing information on how to adjust its signal strength.

Now, consider a controlling node u (note that u is only surrounded by regular nodes). After receiving neighborhood information from each neighboring node, u computes the extended local spanning tree T and transmits it to its neighbors, which is the last action of a control node in the LPA. \square

Summarizing, each node broadcasts exactly two messages during the execution of the LPA. The first message announcing the status of this node to all neighbors is of fixed size; the second is bounded by the size of the local neighborhood. In case of a regular node, the second message contains information about the direct neighborhood, in case of a control node, a spanning tree on the first order neighborhood is transmitted, extended by some links towards the second order neighbors and their main controlling instance.

The last theorem of this section shows that the solution $P = \{P_u \mid u \in V\}$ created by the LPA induces an overall connected subgraph in the maximal graph G , thus, the LPA computes a feasible solution for both problems discussed in this paper.

Theorem 7: The solution created by the LPA algorithm is feasible, that is, the resulting graph $G_P = (V, A_P)$ is connected by bidirectional links.

Proof: Any regular node is dominated by a control node, the local MST ensures that there exists a bidirectional path connecting this dominated node to the control node. Thus, it suffices to show connectivity by bidirectional paths for the control nodes. As the set of control nodes forms a *maximal independent set*, we need only to show that each control node $c \in V$ can reach *all* control nodes that are at most three hops away from it in the maximal graph G following a bidirectional path.

In G , there are no control nodes in the direct neighborhood of c . If \bar{c} is a control node in the second order neighborhood, then there exists an intermediate node $u \in N_c \cap N_{\bar{c}}$ which is connected to c and \bar{c} in \bar{G}_c and $\bar{G}_{\bar{c}}$ respectively.

Now consider a control node \bar{c} in the third order neighborhood of c . Be $v_1 \in N_c$ and $v_2 \in N_{\bar{c}}$ such that $p(v_1, v_2)$ is minimal. We have to show that there exists a bidirectional path between the two controlling instances of v_1 and v_2 . By looking at these main control nodes C_{v_1} and C_{v_2} , we obtain the following cases, also given in Figure 2:

- $C_{v_1} = c \wedge C_{v_2} = \bar{c}$ (Figure 2.i.)

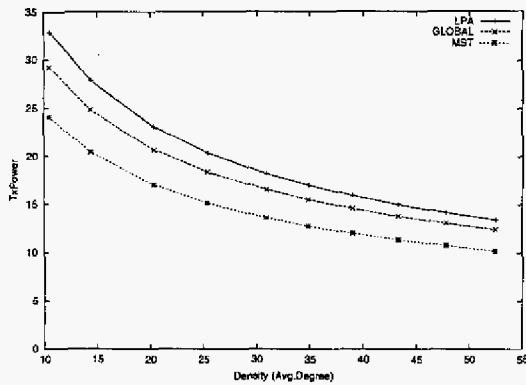


Fig. 3. Performance of LPA (Total Power Adjustment)

As $C_{v_2} = \bar{c}$, the second order neighbor v_2 of c is included in the extended neighborhood of c during the construction of the extended MST. By symmetry, the same holds for the tree computed in \bar{c} . Both control nodes include the edge (v_1, v_2) in their respective extended trees, i.e. the edge with the lowest weight.

- $C_{v_1} = C_{v_2}$ (2.ii.)

Be C_v the control node both v_1 and v_2 consider their controlling instance. As both v_1 and v_2 are in the neighborhood of C_v , they are bidirectionally connected.

- $C_{v_1} = c \wedge C_{v_2} \neq \bar{c}$ (2.iii.)

By the same argument as the first case, c and C_{v_2} are connected by a bidirectional path. Thus there exist two paths connecting c and C_{v_2} as well as C_{v_2} and \bar{c} . Note that the case $C_{v_1} \neq c \wedge C_{v_2} = \bar{c}$ is symmetric.

- $C_{v_1} \neq c \wedge C_{v_2} \neq \bar{c} \wedge C_{v_2} \neq C_{v_1}$ (2.iv.)

There exists a path connecting v_1 and v_2 since C_{v_1} and C_{v_2} are connected. □

VI. RESULTS

Figure 3 presents results from simulations of the LPA algorithm for networks of different densities. Each data point represents an average value taken over one thousand runs. Each simulation run is set up by placing a thousand wireless nodes randomly in a square area. The maximum transmission range, i.e. P_{max} , is then globally adjusted for all nodes to meet the desired degree of connectivity for the network. For the graph in Figure 3, the value on the horizontal axis gives the average number of neighbors of each node in the resulting ad-hoc network. The values given vertically represent the average ratio (in %) of the assigned transmission power with respect to the maximum value (P_{max}) for a node in the network.

For the Total Power Adjustment problem, the performance of the LPA as computed by the algorithm run on each node is given. The performance of a solution computed by a base station having knowledge of the complete topology (GLOBAL) is also given. For reference, the average weight of a link that is contained in a minimum spanning tree (MST) of the communication network (as computed with global view of the topology) is also given. Note that the values given by the MST represent a lower bound on the best possible assignment of

transmission powers for the total power adjustment problem. With respect to the MST, the global algorithm is on average 21.5 % off, the local power adjustment 33 %.

For the Maximal Power Adjustment problem, the maximal power assigned to a node is 12 % higher than the optimal assignment.

VII. CONCLUSION

This paper presents a local approach to topology control problems. Among the nodes in the wireless network control nodes are chosen that construct a local minimum spanning tree. The algorithm only requires two messages to be locally broadcast per node in the network.

On a global view, a minimum spanning tree solves the Maximal Power Adjustment Problem with optimality, and is at most by a factor of 2 off the optimal solution for the Total Power Adjustment Problem. From our simulations, the LPA is on average about 10 % off a globally computed solution, and the average power assigned to a node is 33 % higher than that given by an MST as lower bound.

The presented algorithm is suited for an ad-hoc network with static or quasi-static nodes, i.e. a network that can be seen as static for a reasonable period of time. Future work may focus on more dynamic network topologies, for example due to mobility or failure of nodes, and on how to maintain the structure in face of these changes.

REFERENCES

- [1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1993.
- [2] A.E.F. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity*, TR-54, 2000, <http://www.eccc.uni-trier.de/eccc/>.
- [3] EYES. website. <http://eyes.eu.org>.
- [4] M. Faloutsos and M. Moile. Optimal distributed algorithm for minimum spanning trees revisited. In *Symposium on Principles of Distributed Computing*, pages 231–237, 1995.
- [5] L. Jia, R. Rajaraman, and C. Scheidele. On local algorithms for topology control and routing in ad hoc networks. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures*, 2003.
- [6] L.M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. In *Theoretical Computer Science, Vol. 243, Issues 1-2*, pages 289–305. Springer Verlag, 2000.
- [7] M. Kubisch, H. Karl, A. Wolisz, L.C. Zhong, and J. Rabaey. Distributed algorithms for transmission power control in wireless sensor networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, March 2003. IEEE.
- [8] L. Li, J. Halpern, V. Bahl, Y.M. Wang, and R. Wattenhofer. Analysis of cone-based distributed topology control algorithms for wireless multihop networks. In *Proceedings of ACM Symposium on Principles of Distributed Computing*, pages 264–273, 2001.
- [9] E.L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, and S.S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. In *Proc. MOBIHOC'02. ACM*, pages 123–134, 2002.
- [10] D. Maniezzo, K. Yao, and G. Mazzini. Energetic trade-off between computing and communication resource in multimedia surveillance sensor network.
- [11] R. Rajaraman. Topology control and routing in ad hoc networks: a survey. *ACM SIGACT News*, 33:60–73, 2002.
- [12] R. Ramanathan and R. Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM (2)*, pages 404–413, 2000.
- [13] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks, 1998.
- [14] S. Ulukus and R. Yates. Stochastic power control for cellular radio systems. In *IEEE Trans. Comm.*, 46(6), pages 784–798, 1996.