

A Distributed Range Assignment Protocol

Steffen Wolf, Tom Ansay, and Peter Merz

Distributed Algorithms Group
University of Kaiserslautern
Kaiserslautern, Germany

Abstract. We present a new distributed algorithm for creating and maintaining power-efficient topologies in a wireless network. The wireless nodes establish links to neighbouring nodes in a self-organizing fashion. The protocol is designed to first create a connected topology and then iteratively search for better links to reduce the overall power consumption. First results from simulated experiments with various network sizes show that the resulting topologies are close to optimal in respect to the total energy consumption.

1 Introduction

Self-organization is an important aspect in wireless ad-hoc networks and wireless sensor networks. In these networks, which work without a wired backbone infrastructure, each node must bring its own energy source. In many scenarios, the wireless nodes are equipped with batteries, which provide them with a limited amount of energy. Since energy is needed for communication, the lifespan of the network is limited as well. In order to increase the lifespan of the wireless network, one could either provide the nodes with better energy sources, or try to use the available energy as efficiently as possible.

In this paper, we demonstrate how the wireless network itself can be used as a self-optimizing system to reduce the energy needed for transmitting data. We will adopt a high level of abstraction from Santi [1], using a graph representation. We assume that the wireless nodes can adjust their transmission power and that a higher transmission power corresponds to a larger coverage area for this node.

Section 2 gives a detailed description of the problem, and Section 3 gives an overview of related work. In Section 4 we present our distributed algorithm, and in Section 5 we provide first results of our experiments. Section 6 concludes the work and gives an outline of future research.

2 Problem Formulation

The set of problems of assigning transmission ranges to wireless nodes such that the resulting topology has a pre-defined property (e.g. connectivity) and the overall energy consumption is minimized are called Range Assignment Problems (RAPs) [1]. There are many interesting RAPs. For example in the Minimum

Energy Broadcast Problem (MEB), one node has to transmit a message to all other nodes of the network. These nodes can relay the message, so the topology is a directed tree rooted at the first node spanning the whole network.

In this paper, we want to concentrate on topologies that are symmetrically connected, i.e. a link is established only if both nodes have a transmission range high enough to reach the other node. This RAP is called the Weakly Symmetric Range Assignment Problem (WSRAP). It is NP-hard [2, 3], and can be formulated as follows:

Given a network $G = (V, V \times V)$ of wireless nodes V and a cost function $c : V \times V \rightarrow \mathbb{R}$, we search for the spanning tree $T \subseteq V \times V$ that minimizes the overall power consumption

$$C(T) = \sum_{v \in V} \underbrace{\max_{u \in V: \{u, v\} \in T} c(u, v)}_{\text{transmission power of node } v} \quad (1)$$

Usually, the cost function c is modelled as a polynomial function of the Euclidean distance d using a distance-power gradient α : $c(u, v) = d(u, v)^\alpha$. Measurements have shown that this distance-power gradient α can take values between 1 and 6. In an ideal obstacle-free environment, it takes the value of $\alpha = 2$.

In the same way, the transmission power assigned to a node translates to a transmission range. In the WSRAP, the transmission power of a wireless node is set such that the node can reach its farthest neighbour. Note that the formulation can be changed to a formulation for the Minimum Spanning Tree (MST) problem just by exchanging the max with a sum in (1). However, while the MST can be solved in polynomial time, the WSRAP is NP-hard. The MST can still be used as an approximation of the WSRAP. It has been shown that the MST already gives a 2-approximation [2, 3].

A valuable side effect of the overall power minimization is a reduction of interference in the wireless network, as the lower transmission ranges translate to smaller interference areas. However, actually minimizing interference is a different optimization problem, and is not part of our analysis.

3 Related Work

The WSRAP was first introduced by Călinescu *et al.* in [2] as a variant of another Range Assignment Problem searching for strong connectivity only. They show that the Minimum Spanning Tree (MST) is a 2-approximation for the WSRAP. This approximation ratio is also shown to be tight.

Cheng *et al.* rediscover the problem in [3]. They emphasize that the problem is different than the strong connectivity range assignment problem and provide a new proof of NP-hardness. They also rediscover the MST as a 2-approximation, and present a new greedy heuristic. This heuristic builds the tree in a way that resembles Prim's algorithm for building MSTs, but it needs global knowledge and can only be transformed to a distributed algorithm with a high level of cooperation between the nodes.

Improving upon these global heuristics, Althaus *et al.* [4] present two local search heuristics. In the edge-switching (ES) heuristic, edges from the tree are replaced by non-tree edges re-establishing the connectivity. In the edge-and-fork-switching (EFS) heuristic, pairs of edges, so-called forks, are inserted in the tree, and the resulting cycles are cut again by removing other edges. In both heuristics, the process is repeated until a local optimum is reached. The authors also try to limit the number of hops for the ES heuristic, as this translates to the level of cooperation in a distributed algorithm. However, they do not proceed in this direction and do not model the wireless network as a self-organizing system.

Optimal topologies can be found using Mixed Integer Programming formulations (MIP) by Althaus *et al.* [4] or Montemanni *et al.* [5]. In both papers, problem instances with up to 40 nodes were solved. While Althaus *et al.* take more than an hour for each of these instances, Montemanni *et al.* can solve an instance of this size in less than two minutes. However, calculation times increase exponentially with the network size for both algorithms.

Near optimal topologies can be found in shorter time using our Iterated Local Search (ILS) based on the described local search heuristics [6]. Solutions for problem instances with up to 1000 nodes are presented. The largest network size for which optimal topologies are known is 100 (using the MIP from Montemanni it takes a couple of days), and the strongest ILS finds topologies with average costs of only 0.002% above the optimum in less than three minutes.

To the best of our knowledge, no fully developed distributed algorithm for the WSRAP has been proposed so far. There are distributed algorithms for the MST which could be used to construct a 2-approximation to the WSRAP. But the challenge of designing a distributed algorithm for the local search heuristics that includes propagation of routing information and cycle detection, and that works without a high level of cooperation, has not yet been tackled.

4 A Distributed Algorithm

In our distributed algorithm, we assume that each node maintains a list of neighbours, a list of links, and a routing table. Links are established and revoked using a set of control messages. Nodes relay messages for other nodes and react on incoming control messages. The nodes also become active from time to time to try and establish new links or improve the current network topology.

When a node joins a network, it sends out a broadcast ping message to make itself known. Using the signal strength of the response messages, it can calculate the necessary transmission power to reach its neighbouring nodes.

In a first phase, shown in Fig. 1, the node actively sends out **AddLink** messages to the closest not yet reached neighbour. The usual response is a **ConfirmLink** message, containing the routing information of the other node. The link is established and the routing information is updated. Both nodes then propagate the new routing information to their part of the network.

In a second phase, shown in Fig. 2, the node intentionally sends **AddLink** messages to neighbouring nodes that can already be reached. Using the routing

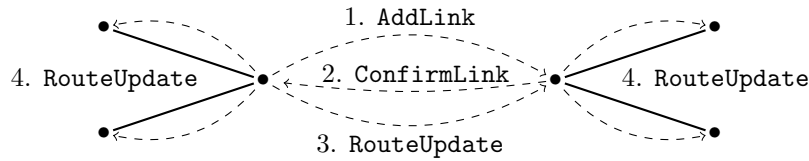


Fig. 1. Adding a new link

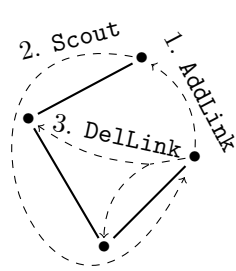


Fig. 2. Replacing a link

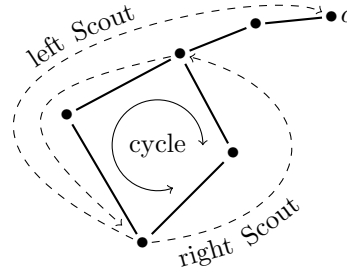


Fig. 3. Cycle detection

information, these nodes send a **Scout** back to the origin node. Each node that relays the **Scout** adds information on how much energy it could save if it were to remove the links the **Scout** is taking. When the **Scout** reaches the origin node, it decides based on this information whether it is favourable to add the new link and which remote link should be deleted. If so, **DelLink** messages are sent to the nodes adjoining the remote link, and an **AddLink** message is sent again to the neighbouring node. Using the new routing information, the neighbouring node should respond with a **ConfirmLink** and the link is established. This phase of the protocol can be seen as a distributed version of the ES heuristic.

Since our aim is to create a self-organizing wireless network, we do not synchronize the nodes. So it can and will happen, that establishing a link creates a cycle in the topology. When the routing information is propagated after adding the link, some remote nodes will recognize the cycle, because they receive contradicting routing information for another node c . As can be seen in Fig. 3, this node c is not necessarily part of the cycle, it can also be several hops away from a node of the cycle. The algorithm has to take care that only links of the actual cycle are considered to be removed. The path from the cycle to node c must not be severed, otherwise the network would be split and the cycle would remain. If a cycle is detected, two **Scouts** are sent to the node c , one for each route. If the **Scout** passes a node that knows a different route back to the origin node, it sends the **Scout** back over this route. If the **Scout** reaches the node c , but this node c does not know a different route back, it ignores the **Scout**. So, only one **Scout** will return to the origin node, this **Scout** has only traversed the links of the cycle, and the origin node can decide which link should be removed and send the necessary **DelLink** messages.

5 Experiments

For our experiments, we used nine sets of well established test instances [6–8]. Each set contains 30 instances, where n nodes are randomly located in a $10\,000 \times 10\,000$ grid, using a uniform distribution. Euclidean distance was used and the distance-power gradient was set to $\alpha = 2$ as in an ideal environment. The sets use $n \in \{20, 50, 100, 200, 500, 1000\}$ nodes, and there are three sets of clustered instances ($n \in \{100, 200, 500\}$, with $c = \lfloor \sqrt{n} \rfloor$ clusters of size 1000×1000 , the cluster centres placed at random locations in the $10\,000 \times 10\,000$ grid).

Each experiment was repeated 30 times and average values are used for the following discussion. Our simulator was written in C++, and the simulations were conducted on a 2.5 GHz Intel Xeon running 64 bit Linux 2.6. The nodes became active after random waiting intervals between 1 and 10 seconds. To allow for actual concurrency, we simulated a message delay by waiting for 50 ms before processing a message after receiving it. In our experiments we limited the number of links to be added in phase 2 to five. We will show that this already gives quite good results. In normal operation, the nodes could continue trying to improve the topology, possibly piggy-backing the control messages on actual transmissions in the network.

Figure 4 shows the results for the uniformly distributed networks. The excess is plotted over time, averaged over all 30 runs for each of the 30 instances of the same size. The excess defines how much higher the overall energy consumption is compared to the optimal topology (or the best known topology for $n \geq 100$). An excess of 0% means that an optimal topology has been found. In phase 1, which lasts about 20 seconds, the overall transmission energy rises as new links are established to connect the network. Then, the excess is between 17% and 40%. By this time, some nodes have already entered phase 2. The results of the topology changes in phase 2 can be seen in the figures. For larger networks, the transformation is slower, but they also reach the same final excess over the best known topologies: between 1.4% and 3.6%.

For the clustered networks (Fig. 5), both the initial and final excess is higher. The initial excess lies between 39% and 76%, and the final excess between 13% and 19%. In clustered instances, often only one node from each cluster maintains connections to other clusters. If the wrong node is selected to maintain these connections, our distributed algorithm, which is based on the ES heuristic, cannot transfer this task to another node. On the other hand, the amount of energy that actually is saved in these clustered networks is higher.

6 Conclusion

We have presented a self-organizing distributed algorithm to reduce the transmission energy in wireless networks. In experiments with well established test instances, both uniformly distributed and clustered, we have shown that the algorithm can reach close-to-optimal topologies. The energy that is saved using the improved topologies translates to a longer network lifespan without increasing the batteries of the nodes.

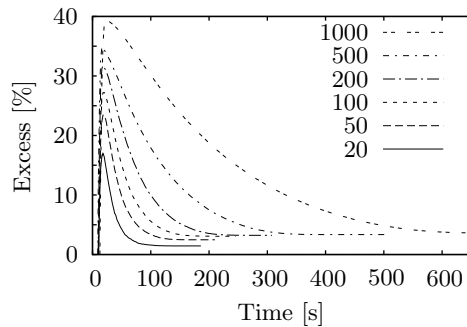


Fig. 4. Results for uniformly distributed random networks

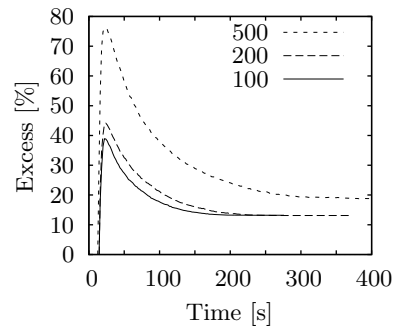


Fig. 5. Results for clustered random networks

In future work, we want to improve the handling of clustered networks, e. g. by allowing the nodes to add more links in phase 2. The increased protocol overhead will have to be weighed against the expected improvements. We also want to expand our algorithm to other RAPs, such as the MEB.

References

1. Santi, P.: *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, Chichester (September 2005)
2. Călinescu, G., Măndoiu, I.I., Zelikovsky, A.: Symmetric connectivity with minimum power consumption in radio networks. In Baeza-Yates, R.A., Montanari, U., Santoro, N., eds.: *Proc. 2nd IFIP International Conference on Theoretical Computer Science*. Volume 223 of *IFIP Conference Proceedings*., Kluwer, Dordrecht (2002) 119–130
3. Cheng, X., Narahari, B., Simha, R., Cheng, M.X., Liu, D.: Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics. *IEEE Transactions on Mobile Computing* **2**(3) (July–September 2003) 248–256
4. Althaus, E., Călinescu, G., Măndoiu, I.I., Prasad, S.K., Tchervenski, N., Zelikovsky, A.: Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks. *Wireless Networks* **12**(3) (May 2006) 287–299
5. Montemanni, R., Gambardella, L.M.: Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Computers & Operations Research* **32**(11) (2005) 2891–2904
6. Wolf, S., Merz, P.: Iterated local search for minimum power symmetric connectivity in wireless networks. In Cotta, C., Cowling, P., eds.: *EvoCOP 2009 – Ninth European Conference on Evolutionary Computation in Combinatorial Optimization*. Volume 5482 of *LNCS*., Springer, Heidelberg (April 2009) 192–203
7. Al-Shihabi, S., Merz, P., Wolf, S.: Nested partitioning for the minimum energy broadcast problem. In Maniezzo, V., Battiti, R., Watson, J.P., eds.: *LION 2007 II. Selected Papers*. Volume 5313 of *LNCS*., Springer, Heidelberg (2008) 1–11
8. Hernández, H., Blum, C., Francès, G.: Ant colony optimization for energy-efficient broadcasting in ad-hoc networks. In Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T., eds.: *ANTS 2008*. Volume 5217 of *LNCS*., Springer, Heidelberg (September 2008) 25–36