Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 28 (2005) 1498–1504
computer communications
www.elsevier.com/locate/comcom

Algorithms for ad hoc and sensor networks☆

Roger Wattenhofer

ETH Zurich, Zurich, Switzerland

Received 31 August 2004; accepted 15 December 2004
Available online 26 April 2005

Abstract

Wireless and mobiles networks are excellent playground for researchers with an algorithm background. Many research problem turn out to be variants of classic graph theory problems. In particular the rapidly growing areas for ad hoc and sensor networks demand new solutions for timeless graph theory problems, because: (i) wireless devices have lower bandwidth and (ii) wireless devices are mobile and therefore the topology of the network changes rather frequently. As a consequences, algorithms for wireless and mobile networks should have: (i) as little communication as possible and should (ii) run as fast as possible. Both goals can only be achieved by developing algorithms requiring a small number of communication rounds only (so-called *local* algorithm). In the work we present a few algorithmic applications in wireless networking, such as clustering, topology control and geo-routing. Each section is supplemented with an open problem.

Keywords: Topology control; Interference; Clustering; Dominating sets; Geo-routing

1. Introduction

An ad hoc or sensor network consists of mobile nodes featuring, among other components, a processor, some memory, a wireless radio, and a power source; physical constraints often require the power source to be feeble—a weak battery or a small solar cell.

Ad hoc and sensor networks are emerging areas of research that have been studied intensively for a few years only. Roughly, the researchers investigating ad hoc and sensor networks can be classified into two categories. On the one side there are the systems researchers who build real ad hoc or sensor networks; the Berkeley Motes project [16] is a popular hardware platform marketed by Crossbow (www.xbow.com) that is used in many deployments, but alternative hardware platforms are available as well [5,34]. On the other hand there are the theoreticians who try to understand the fundamentals of ad hoc and sensor networks, by abstracting away a few 'technicalities' that arise in real systems.

Not surprisingly—as in other areas of computer science and engineering—there is no consensus what the technicalities are. Most theoreticians model the networks as nodes (points) in a Euclidean plane; two nodes can communicate if they are within their mutual transmission range, which in an unobstructed and homogeneous environment translates into whether their Euclidean distance is at most the maximum transmission range R. This model is widely known as unit disk graph and—though not quite practical—respected as a first step by practitioners.

More surprisingly, however, most theoreticians make much stronger assumptions. It seems that a majority of papers assumes that the nodes are distributed uniformly at random. At a high node density, such a postulation renders many problems trivial. Also it is not clear that a uniform node density distribution makes sense from a practical point of view. Recently deployed large-scale sensor networks report highly heterogeneous node densities—in 'interesting' areas there are several nodes per square meter, whereas in other ('routing-only') areas nodes are hundreds of meters apart. For mobile ad hoc networks (MANET's), it is often assumed that the nodes move Brownian, a behavior that is not often seen in our macroscopic world.

In this paper we advocate using more realistic *graph theoretical* models. We feel that theoretical research should drop *average-case* assumptions such as uniformly at random distributed nodes and/or Brownian motion, and

☆ A preliminary version of this paper was presented as an invited talk at the 30th Workshop on Graph-Theoretic Concepts in Computer Science [36].

E-mail address: wattenhofer@tik.ee.ethz.ch.

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.12.037

instead study *worst-case* distributions and motion models. In this paper we outline a selection of the algorithms that were developed to work also in the non-uniform worst-case.

The paper is organized as follows. In Sections 2, 3, and 4, we sketch a number of algorithmic results in three key areas of ad hoc and sensor networking. In Section 2 we discuss topology control, in Section 3 clustering, and in Section 4 geo-routing, a special but well-studied form of routing. In Section 5 we conclude the paper.

2. Topology control

Since energy is the limiting factor for lifetime and operability of an ad hoc network, researchers have developed a variety of mechanisms and algorithms to conserve energy. These mechanisms and algorithms are often dubbed 'topology control'.

For two communicating ad hoc nodes u and v, the energy consumption of their communication grows at least quadratically with their distance. Having one or more relay nodes between u and v therefore helps to save energy. The primary target of a topology control algorithm is to abandon long-distance communication links and instead route a message over several small (energy-efficient) hops. For this purpose each node in the ad hoc network chooses a 'handful' of 'close-by' neighbors 'in all points of the compass' (we are going to fill in the details later). Having only near neighbors not only helps reducing energy but also interference, since fewer nodes are disturbed by high power transmissions. Clearly nodes cannot abandon links to 'too many' faraway neighbors in order to prevent the ad hoc network from being partitioned or the routing paths from becoming non-competitively long. In general there is a trade-off between network connectivity and sparseness.

Let the graph $G=(V, E)$ denote the ad hoc network before running the topology control algorithm, with V being the set of ad hoc nodes, and E representing the set of communication links. There is a link (u, v) in E if and only if the two nodes u and v can communicate directly. Running the topology control algorithm will yield a sparse subgraph $G_{tc}=(V, E_{tc})$, of G, where E_{tc} is the set of remaining links. The resulting topology G_{tc} should have a variety of properties:

(i) Symmetry. The resulting topology G_{tc} should be symmetric, that is, node u is a neighbor of node v if and only if node v is a neighbor of node u. Asymmetric communication graphs are unpractical, because many communication primitives become unacceptably complicated [32].

(ii) Connectivity/Spanner. Two nodes u and v are connected if there is a path from u to v, potentially through multiple hops. If two nodes are connected in G, then they should still be connected in G_{tc}. Although a minimum spanning tree is a sparse connected subgraph, it is often not considered a good topology, since close-by nodes in the original graph G might end up being far away in G_{tc} (G being a ring, for instance). Therefore the graph G_{tc} is generally not only being asked to be connected, but a spanner. For any two nodes u and v, if the optimal path between u and v in G has cost c, then the optimal path between u and v in G_{tc} has cost $O(c)$.

(iii) Sparseness/Low Degree/Low Interference. The remaining graph G_{tc} should be sparse, that is, the number of links should be in the order of the number of nodes. More ambitiously, one might even ask that *each node* in the remaining graph G_{tc} has a low (constant) degree. Since a low degree alone does not automatically imply low interference (after all nodes might choose few but very far away neighbors!), some researchers have started studying topology control algorithms that concentrate on the interference issue.

(iv) In addition to the properties (i)–(iii) one can often find secondary targets. For instance, it is popular to ask the remaining graph to be planar in order to run a geometric routing algorithm, such as GOAFR [28].

Since connectivity and sparseness run against each other, topology control has been a thriving research area.

The currently best algorithms feature an impressive list of properties. Wang and Li [35] present the currently most promising proposal—a distributed topology control algorithm that computes a planar constant-degree distance-spanner. (As opposed to energy-spanners as considered in earlier work [37,17].) However, the distributed algorithm might be quite slow; in an unlikely (but possible) worst-case instance it will run for a linear number of steps. Also, like many others this algorithm makes strong assumptions: first, all the nodes need to know their exact positions, by means of a global positioning system (GPS) for example. Second, the algorithm assumes that the world is flat and without buildings (a perfect unit disk graph, so to speak). These assumptions make the algorithm unpractical.

In an almost 'retro' approach [38] recently presented the XTC algorithm that works: (i) without GPS and (ii) even in a mountainous and obstructed environment. Surprisingly the XTC algorithm features all the basic properties of topology control (symmetry, connectivity, low degree) while being faster than any previous proposals.

All known topology control algorithms including [35] and XTC [38] do not explicitly address interference, but argue that the sparseness or low degree property will take care of it.[1] In [9] it has recently been shown that

[1] Meyer auf der Heide et al. [29] are a notable exception who study interference explicitly, however, not in the context of topology control, but in relation to traffic models. They show that there are worst-case ad hoc networks and worst-case traffic, where only one of the performance parameters congestion, energy, and dilation can be optimized at a time.

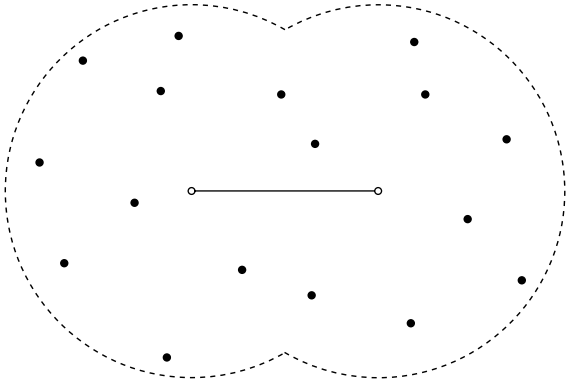

Fig. 1. Nodes covered by a communication link.

the 'low degree⇒low interference' assumption is not correct in a worst case.

In [9] interference is formally defined as follows: given a communication graph produced by a topology control algorithm, the *coverage* of an (undirected) edge $e=(u, v)$ is the cardinality of the set of nodes covered by the disks induced by u and v, with radius $|uv|$:

$$Cov(e) := |\{w \in V | w \text{ is covered by } D(u, |u, v|)\} \cup \{w \in V | w \text{ is covered by } D(v, |v, u|)\}|.$$

In other words the coverage $Cov(e)$ represents the number of network nodes affected by nodes u and v communicating with their transmission powers chosen such that they exactly reach each other (cf. Fig. 1). Then the interference of a graph $G=(V, E)$ is

$$I(G) := \max_{e \in E} Cov(e).$$

To the best of our knowledge, all currently known topology control algorithms have in common that every node establishes a connection to at least its nearest neighbor. In other words all these topologies contain the Nearest Neighbor Forest constructed on the given network. In the following we show that by including the Nearest Neighbor Forest as a subgraph, the interference of a resulting topology can become incomparably bad with respect to a topology with optimum interference. In particular, interference of any proposed topology is $\Omega(n)$ times larger than the interference of the optimum connected topology, where n is the total number of network nodes.

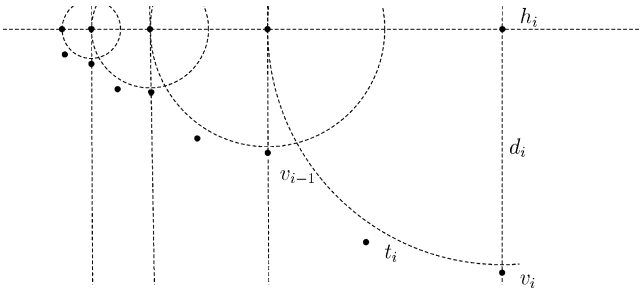

Fig. 2. Two exponential node chains.

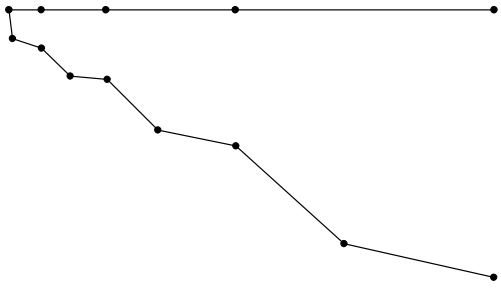

Fig. 3. The Nearest Neighbor Forest yields interference $\Omega(n)$.

Fig. 2 depicts an example graph. In addition to a horizontal exponential node chain, each of these nodes h_i has a corresponding node v_i vertically displaced by a little more than h_i's distance to its left neighbor. Denoting this vertical distance d_i, $d_i > 2^{i-1}$ holds. These additional nodes form a second (diagonal) exponential line. Between two of these diagonal nodes v_{i-1} and v_i, an additional helper node t_i is placed such that $|h_i, t_i| > |h_i, v_i|$.

The Nearest Neighbor Forest for this given network (with the additional assumption that each node's transmission radius can be chosen sufficiently large) is shown in Fig. 3. Roughly one third of all nodes being part of the horizontally connected exponential chain, interference of any topology containing the Nearest Neighbor Forest amounts to at least $\Omega(n)$. An interference-optimal topology, however, would connect the nodes as depicted in Fig. 4 with constant interference.

In other words, already by having each node connect to the nearest neighbor, a topology control algorithm makes an 'irrevocable' error. Moreover, it commits an asymptotically worst possible error, since the interference in any network cannot become larger than n.

Since roughly one third of all nodes are part of the horizontal exponential node chain in Fig. 2, the observation also holds for an average interference measure, averaging interference over all edges.[2]

In [9] three algorithm variants are presented that indeed minimize interference, and at the same time keep the symmetry and the connectivity/spanner property. These algorithms have drawbacks too: currently only one of them is locally computable, but its running time is too slow, which makes a practical implementation impossible.

All the previously discussed algorithms work for arbitrary (worst-case) node distributions. For average-case (random) distributions there is an interesting alternative: each node simply chooses its k best neighbors. Blough et al. [11] show that this simplest of all conceivable algorithms works surprisingly well when the nodes are distributed uniformly at random. For general distributions, clearly [11] does not even guarantee connectivity.

[2] Interestingly, the example in Fig. 2 works as well for a number of other definitions of interference.

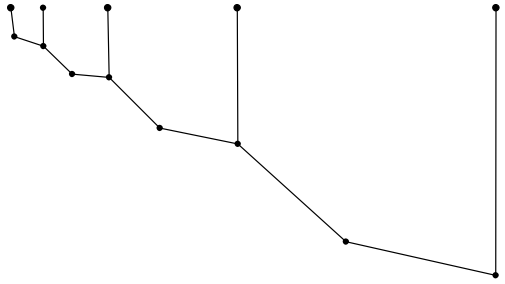

Fig. 4. Optimal tree with constant interference.

Topology control has been (and still is!) a thriving research area for theoreticians. What works well in analysis and simulation has recently also been implemented on the basis of the 802.11 standard [19]. These early practical experiences proof that topology control is a technique that is paying off, and deserves more attention.

Open Problem. For the sake of concreteness, let us specify one of the many open problems. We are given n nodes in the plane. As above we must connect these nodes with a spanning tree. This time, however, we do not charge each edge by how many nodes it will disturb. Instead we charge each node by how many edges it is disturbed. The spanning tree should be chosen such that it minimizes the maximum (or average) disturbed node. Apart from a simple directed sensor-network model [13] nothing is known about the problem.

3. Clustering

Akin to topology control, clustering (a.k.a. backbone building) also aims for computing a subgraph of the original graph. In some sense, however, in clustering this subgraph is not trying to optimize energy by dropping long-range neighbors, but (quite on the opposite) optimizing the number of hops by dropping short-range neighbors.

In mobile ad hoc networks, nodes communicate without stationary server infrastructure. When sending a message from one node to another, intermediate network nodes have to serve as routers. Although a number of interesting suggestions have been made, finding efficient algorithms for the routing process remains the most important problem for ad hoc networks. Since the topology of an ad hoc network is constantly changing, routing protocols for ad hoc networks differ significantly from the standard routing schemes which are used in wired networks. One effective way to improve the performance of routing algorithms is by grouping nodes into clusters. The routing is then done between clusters. A most basic method for clustering is calculating a dominating set. Formally, in a graph G, a dominating set is a subset of nodes such that for every node v either: (i) v is in the dominating set or (ii) a direct neighbor of v is in the dominating set. The minimum dominating set problem asks for a dominating set of minimum size. Only the nodes of the dominating set act as routers, all other nodes communicate via a neighbor in the dominating set.

Between traditional wired networks and mobile ad hoc networks two main distinctions can be made: (i) typically wireless devices have much lower bandwidth than their wired counterparts and (ii) wireless devices are mobile and therefore the topology of the network changes rather frequently. As a consequence, distributed algorithms which run on such devices should have as little communication as possible and they should run as fast as possible. Both goals can only be achieved by developing algorithms requiring a small number of communication rounds only (often called local algorithms).

Most of the algorithms to compute a dominating set use the fact that a maximal independent set (MIS) is by definition already a dominating set. For unit disk graphs it can be shown that any MIS is only a constant factor larger than a minimum dominating set. Often, in a second phase of the algorithm the nodes in the MIS are then connected through two- and three-hop bridges. All these nodes (the MIS and the bridging nodes) then form the backbone. One can route from any backbone node to any other through nodes in the backbone only [2].

Unfortunately, from a worst-case standpoint, it is conjectured that computing a MIS is not as efficient as it seems at first sight. In particular in [23] it was shown that a distributed MIS construction can take as long as $\Omega(\sqrt{\log n/\log\log n})$ time in a graph with n nodes.[3] This is too slow in the setting of a mobile ad hoc network because by the time the MIS is computed, the topology has already changed. In a paper by Gao et al. [15] it was shown that in a unit disk graph one can construct an asymptotically optimal dominating set in time $O(\log\log n)$ only. However, to do so, nodes need to know their coordinates, an assumption that is not always realistic.

Recently, algorithms to quickly compute a dominating set fast even if there the nodes do not know their coordinates have been proposed. These algorithms in fact even work if the network is not a unit disk but a general graph. In general graphs, the problem of finding a minimum dominating set has been proven to be NP-hard. The best known approximation is already achieved by the greedy algorithm: as long as there are uncovered nodes, the greedy algorithm picks a node which covers the biggest number of uncovered nodes and puts it into the dominating set. It achieves an approximation ratio of $\ln \Delta$ where Δ is the highest degree in the graph. Unless the problems of NP can be solved by deterministic $n^{O(\log\log n)}$ algorithms, this is the best possible up to lower order terms [12]. In [18] a logarithmic approximation in polylogarithmic time was proposed.

In [24] the only distributed algorithm which achieves a nontrivial approximation ratio in a constant number

[3] Another lower bound is $\Omega(\log \Delta/\log\log \Delta)$, where Δ is the maximum degree (number of neighbors) in the graph.

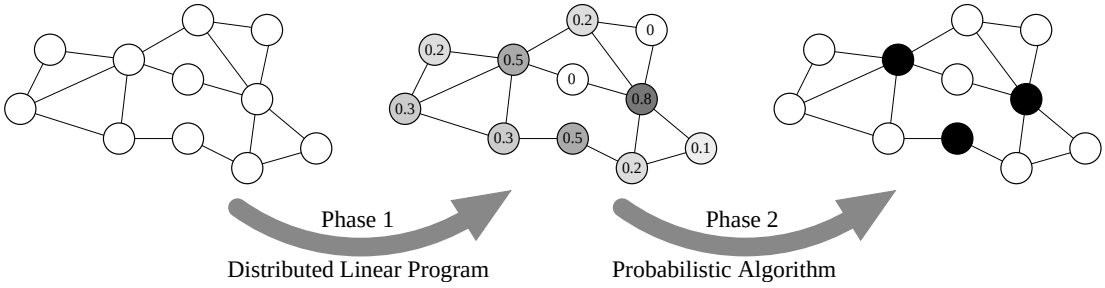

Fig. 5. Distributed dominating set approximation.

of rounds is given. Precisely, for an arbitrary parameter k, in $O(k)$ rounds, an expected approximation ratio of $O(\sqrt{k}\Delta^{2/\sqrt{k}} \log \Delta)$ is presented.

The algorithm consists of two phases (see Fig. 5). First, an approximate solution to the fractional dominating set problem is obtained. In the fractional MDS, weights are assigned to all nodes such that the sum of weights each node sees is greater than or equal to 1. If the MDS problem is formulated as an integer program, the fractional MDS corresponds to the LP relaxation of MDS. The solution to the fractional dominating set can be summarized as follows. Initially all nodes have weight 0. As the algorithm progresses, the nodes gradually increase their weights. This is done in decreasing order of the degrees of the nodes. In order to achieve the locality, the degrees are divided into classes and the assigning of weights is done simultaneously for all nodes of the same class. We obtain a distributed algorithm for the fractional MDS which computes a $k\Delta^{2/k}$-approximation in $O(k^2)$ rounds.

In the second phase of the algorithm, based on their weights, the nodes locally decide whether they become a dominater or not. The second phase only needs two rounds of communication and it merely adds a factor $O(\log \Delta)$ to the overall approximation ratio. This is asymptotically optimal since the integrality gap of the problem is $\ln \Delta$ unless P almost equals NP. In an optional third phase (which is omitted in Fig. 5) nodes can locally approximate a *connected* dominating set by building 'bridges' between dominators.

Recently, with a primal-dual approach it was possible to improve the algorithm such that the first phase of the algorithm essentially constructs a local polynomial time approximation scheme (PTAS), not only for dominating sets but for more general covering and packing problems [25].

All algorithms so far assume that the scheduling of transmissions is handled by the MAC layer. In other words, they assume perfect point-to-point connections between two neighboring nodes. Since a backbone (dominating set) is often used to compute a reasonable MAC layer, many of these papers experience a severe 'chicken-and-egg' problem. Kuhn et al. [21] take a more realistic approach to clustering in ad hoc networks. They consider a multi-hop radio network without collision detection, where nodes wake up asynchronously, and do not have access to a global clock. For this rather harsh model, they show that a $O(1)$-approximative dominating set can be computed within $polylog(\hat{n})$ time, $\hat{n}$ being an a-priori upper bound on the number of nodes in the system.

Open Problem. Though there is some early understanding about the static version of the problem of clustering using dominating sets, the question how to efficiently maintain a clustering when the nodes are mobile, is still wide open.

4. Geo-routing

Routing is of central importance in ad hoc networks. With the notable exception of a link reversal [14] routing algorithm analysis by Busch et al. [10], not many worst-case results are known.

For a special case of routing known as geo-routing (a.k.a. geographic, geometric, location-, or position-based routing), however, there have been quite a few worst-case results. In geo-routing each node is informed about its own as well as its neighbors' positions. Additionally the source of a message knows the position of the destination. The first assumption becomes more and more realistic with the advent of inexpensive and miniaturized positioning systems. It is also conceivable that approximate position information could be attained by local computation and message exchange with stationary devices [4,6] or completely autonomously [33,30]. In order to come up to the second assumption, that is to provide the source of a message with the destination position, a (peer-to-peer) overlay network could be employed [3,39,1].[4] For some scenarios it can also be sufficient to reach any destination currently located in a given area ('geocasting' [31]).

The first correct geo-routing algorithm was Face Routing [20]. Face Routing routes messages along faces of planar graphs and proceeds along the line connecting the source and the destination. Besides guaranteeing to reach the destination, it does so with $O(n)$ messages, where n is the number of network nodes. Face routing was later

[4] Abraham et al. [1] fits well into the context of this paper, since the authors share our worst-case philosophy.

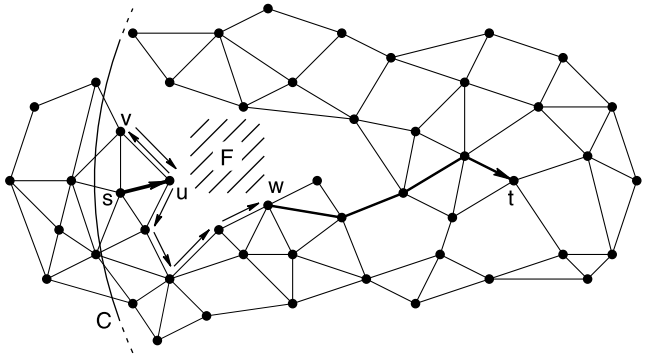

Fig. 6. The GOAFR$^+$ algorithm starts from s in greedy mode. At node u it reaches a local minimum, a node without any neighbors closer to t. GOAFR$^+$ switches to face routing mode and begins to explore the boundary of face F (in clockwise direction). At node v the algorithm hits the bounding circle C (for details, please see [28,26]) and turns back to continue the exploration of F's boundary in the opposite direction. At node w the algorithm decides that it made significant progress [26], falls back to greedy mode, and continues to finally reach destination node t.

combined with greedy routing to give better average-case performance [7].

This is unsatisfactory since already a simple flooding algorithm will reach the destination with $O(n)$ messages. Additionally it would be desirable to see the algorithm cost depend on the distance between the source and the destination. The first algorithm competitive with the shortest path between the source and the destination was AFR [27]. It basically enhances Face Routing by the concept of a bounding region restricting the searchable area. With a lower bound argument AFR was shown to be asymptotically optimal.

Despite its asymptotic optimality AFR is not practicable due to its pure face routing concept. For practical purposes there have been attempts to combine greedy approaches (always send to the message to the neighbor closest to the destination) and face routing; for example the GOAFR and GOAFR+ algorithms by Kuhn et al. [28,26], which are variants of AFR and remain worst-case optimal (see Fig. 6).

On the other side, GOAFR+ is currently also the best geo-routing algorithm in the average-case. In this sense GOAFR+ is a success story for worst-case analysis, where an algorithm derived from a worst-case algorithm is also the best average-case algorithm.

Open Problem. Recently [33] proposed to use geo-routing algorithms in complete absense of position information. Instead, an algorithm assigns so-called 'virtual coordinates' to the nodes; these virtual coordinates should model the connectivity of the nodes as well as possible. In particular, each node is assigned a coordinate in the plane, such that nodes that are neighbors in the connectivity graph have at most Euclidean distance 1 in the plane, and nodes that are *not* neighbors in the connectivity graph have at least distance 1. In other words, we would like to *embed* a given unit disk graph in the plane. Unfortunately, it was shown by Breu and Kirkpatrick [8] that this is impossible in polynomial time. Recently, there was progress in understanding the problem better by the first non-trivial lower bound [22], and also the first non-trivial approximation algorithm for the problem [30]. However, the gap between the upper and the lower bound is still glaring; we believe that this is a most challenging open problem.

5. Conclusions

In this paper we have discussed several 'worst-case' algorithms for various classic problems in ad hoc and sensor networking. Clearly, the selection of areas in this paper is highly subjective. Besides topology control, clustering, and geo-routing there are a dozen more research areas that are currently in the focus of the community (e.g. positioning, models, data gathering, multicast). Moreover the selection is dreadfully skewed towards our own recent work.

However, there is not as much algorithmic work as one might think. The vast majority of ad hoc and sensor network research follows the heuristics/simulations approach: a heuristic for solving a problem is proposed, and simulated against other heuristics. Unfortunately, this approach does rarely produce solid results, on which one can build, since the quality of the heuristics depends on the parameters of the simulation. We feel that with the field generally becoming more mature, 'average-case' heuristics will make way for 'worst-case' algorithms.

References

[1] Ittai Abraham, Danny Dolev, Dahlia Malkhi, LLS: A Locality Aware Location Service for Mobile Ad hoc Networks. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 2004.
[2] K. Alzoubi, P.-J. Wan, O. Frieder, Message-Optimal Connected Dominating Sets in Mobile Ad hoc Networks, in: Proceedings of ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), EPFL Lausanne, Switzerland, 2002, pp. 157–164.
[3] B. Awerbuch, D. Peleg, Concurrent Online Tracking of Mobile Users, in: SIGCOMM, 1991, pp. 221–233.
[4] J. Beutel, Geolocation in a Picoradio Environment. Master Thesis, ETH Zurich and UC Berkeley, 1999.
[5] J. Beutel, O. Kasten, M. Ringwald, BTnodes—A Distributed Platform for Sensor Nodes, in: Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), 2003.
[6] R. Bischoff, R. Wattenhofer, Analyzing Connectivity-based Multi-hop Ad-hoc Positioning, in: Proceedings of the Second Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), 2004.
[7] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with Guaranteed Delivery in Ad hoc Wireless Networks, in: Proceedings of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 1999, pp. 48–55.
[8] H. Breu, D.G. Kirkpatrick, Unit disk graph recognition is NP-hard, Comput. Geom. Theory Appl. 9 (1–2) (1998) 3–24.
[9] M. Burkhart, P. von Rickenbach, R. Wattenhofer, A. Zollinger, Does topology control reduce interference? in: Proceedings ACM

International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), 2004.
[10] C. Busch, S. Surapaneni, S. Tirthapura, Analysis of Link Reversal Routing Algorithms for Mobile Ad hoc Networks, in: 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2003.
[11] D.M. Blough, M. Leoncini, G. Resta, P. Santi, The *k*-Neigh Protocol for Symmetric Topology Control in Ad hoc Networks, in: Proceedings of ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), 2003.
[12] U. Feige, A threshold of ln *n* for approximating set cover, J. ACM (JACM) 45 (4) (1998) 634–652.
[13] M. Fussen, R. Wattenhofer, A. Zollinger, On Interference Reduction in Sensor Networks. Technical Report 453, Department of Computer Science, ETH Zurich, 2004.
[14] E.M. Gafni, D.P. Bertsekas, Distributed algorithms for generating loop-free routes in networks with frequently changing topology, IEEE Trans. Commun. 29 (1981).
[15] J. Gao, L. Guibas, J. Hershberger, L. Zhang, A. Zhu, Discrete Mobile Centers, In: Proceedings of 17th Annual Symposium on Computational Geometry (SCG), ACM Press, 2001.
[16] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, K.S.J. Pister, System Architecture Directions for Networked Sensors, in: Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2000, pp. 93–104.
[17] L. Jia, R. Rajaraman, C. Scheideler, 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 (SPAA), 2003.
[18] L. Jia, R. Rajaraman, R. Suel, An Efficient Distributed Algorithm for Constructing Small Dominating Sets, in: Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC), 2001, pp. 33–42.
[19] V. Kawadia, P.R. Kumar, Power Control and Clustering in Ad hoc Networks, in: Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2003.
[20] E. Kranakis, H. Singh, J. Urrutia, Compass Routing on Geometric Networks, in: Proceedings of 11th Canadian Conference on Computational Geometry (CCCG), Vancouver, August 1999, pp. 51–54.
[21] F. Kuhn, T. Moscibroda, R. Wattenhofer, Initializing Newly Deployed Ad hoc and Sensor Networks, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom), 2004.
[22] F. Kuhn, T. Moscibroda, R. Wattenhofer, Unit Disk Graph Approximation. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 2004.
[23] F. Kuhn, T. Moscibroda, R. Wattenhofer, What Cannot be Computed Locally! in: Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC), 2004.
[24] F. Kuhn, R. Wattenhofer, Constant-Time Distributed Dominating Set Approximation, in: Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC), 2003.
[25] F. Kuhn, R. Wattenhofer, Distributed Combinatorial Optimization. Technical Report 426, Department of Computer Science, ETH Zurich, 2003.
[26] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, Geometric Routing: Of Theory and Practice, in: Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC), 2003.
[27] F. Kuhn, R. Wattenhofer, A. Zollinger, Asymptotically Optimal Geometric Mobile Ad-hoc Routing, in: Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), Atlanta, Georgia, USA, September 2002.
[28] F. Kuhn, R. Wattenhofer, A. Zollinger, Worst-Case Optimal and Average-Case Efficient Geometric Ad-hoc Routing, in: Proceedings of ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), 2003.
[29] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, M. Grunewald, Energy, Congestion and Dilation in Radio Networks, in: Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2002.
[30] T. Moscibroda, R. O'Dell, M. Wattenhofer, R. Wattenhofer, Virtual Coordinates for Ad hoc and Sensor Networks. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 2004.
[31] J.C. Navas, T. Imielinski, GeoCast—Geographic Addressing and Routing, in: Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom), 1997, pp. 66–76.
[32] R. Prakash, Unidirectional Links Prove Costly in Wireless Ad-hoc Networks, in: Proceedings of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 1999.
[33] A. Rao, C. Papadimitriou, S. Ratnasamy, S. Shenker, I. Stoica, Geographic Routing Without Location Information, in: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom), 2003.
[34] J. Schiller, et al. The scatterweb project: See http://www.scatterweb.net for more details.
[35] Y. Wang, X.-Y. Li, Localized Construction of Bounded Degree Planar Spanner, in: Proceedings of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing, 2003.
[36] R. Wattenhofer, Wireless Networking: Graph Theory Unplugged, in: Proceedings of the 30th Workshop on Graph-Theoretic Concepts in Computer Science (WG), 2004.
[37] R. Wattenhofer, L. Li, P. Bahl, Y.-M. Wang, Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad hoc Networks, in: Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2001.
[38] R. Wattenhofer, A. Zollinger, XTC: A Practical Topology Control Algorithm for Ad-hoc Networks, in: Proceedings of Fourth International Workshop on Algorithms for Wireless, Mobile, Ad hoc and Sensor Networks (WMAN), 2004.
[39] Y. Xue, B. Li, K. Nahrstedt, A Scalable Location Management Scheme in Mobile Ad-hoc Networks, in: IEEE LCN, 2001.

Roger Wattenhofer received his doctorate in computer science in 1998 from ETH Zurich, Switzerland. From 1999 to 2001 he was in the USA, first at Brown University in Providence, RI, then at Microsoft Research in Redmond, WA. He is now professor at the Information Technology and Electrical Engineering Department, ETH Zurich. Roger Wattenhofer's research interests include a variety of algorithmic aspects in networking and distributed computing; in particular, peer-to-peer computing, ad hoc and sensor networks.