Ad-Hoc and Sensor Networks: Worst-Case vs. Average-Case

Roger Wattenhofer
Department of Computer Science
ETH Zurich
CH-8092 Zurich, Switzerland
Email: wattenhofer@inf.ethz.ch

***Abstract*— Ad-hoc and sensor networks are rapidly growing areas of research which study the problems arising when small and feeble devices build a communication infrastructure. A vast majority of researchers in the field make strong average-case assumptions about these networks, for example that the devices are distributed uniformly at random. To system builders on the other hand many of these assumptions appear suspicious. In this paper we advocate an algorithmic (worst-case) approach to ad-hoc and sensor networking. We survey a few also worst-case efficient algorithms for topology control, clustering, and routing.**

I. 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 [14] is a popular hardware platform now marketed by Crossbow (www.xbow.com) that is used in many deployments, but other hardware platforms are available (e.g. [4]). On the other hand there are the theoreticians who try to understand some of 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 theoretical models. We feel that theoretical research should drop *average-case* assumptions such as uniformly at random distributed nodes, and Brownian motion, and instead study *worst-case* distributions and motion models. In this paper we outline some 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 some of the most interesting 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.

II. 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 [27].

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 [24].

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 [29] 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 [30], [15].) 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. And 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 Wattenhofer and Zollinger [31] recently present 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 [29] and XTC [31] do not directly tackle interference, but argue that the sparseness or low degree property will take care of it. In [7] it has recently been shown that the "low degree ⇒ low interference" assumption is not correct in a worst case. It is shown that all currently proposed topology control algorithms – already by having every node connect to its nearest neighbor – commit a substantial mistake: Although certain proposed topologies are guaranteed to have low degree yielding a sparse graph, interference becomes asymptotically incomparable with the interference-minimal topology. In the same paper 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.

Meyer auf der Heide et al. [25] studied interference 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.

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. [9] show that this simplest of all conceivable algorithms works surprisingly well when the nodes are distributed uniformly at random.

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 [17]. These early practical experiences proof that topology control is a technique that is paying off, and deserves more attention.

III. 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 [1].

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 – for a related computational model – it was shown [13] that a distributed MIS construction can take as long as $\Omega(\log n/\log\log n)$ time with n nodes. 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. [12] 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 [10]. In [16] a logarithmic approximation in polylogarithmic time was proposed. In [20] the only distributed algorithm which achieves a nontrivial approximation ratio in a constant number 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. Note that the approximation ratio has recently been improved in [21], which considers general covering and packing problems.

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. In [19], Kuhn et al. 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.

IV. GEO-ROUTING

Routing is of central importance in ad-hoc networks. With the notable exception of a link reversal [11] routing algorithm analysis by Busch et al. [8], not many worst-case results are known.

For a special case of routing known as geo-routing (a.k.a. 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' position. 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 [3], [5] or completely autonomously [28]. 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 [2], [32]. For some scenarios it can also be sufficient to reach any destination currently located in a given area ("geocasting" [26]).

The first correct geo-routing algorithm was Face Routing [18]. 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 combined with greedy routing to give better average-case performance [6].

This is unsatisfactory since also 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 [23]. It basically enhances Face Routing by the concept of a bounding ellipse 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. [24], [22], which are variants of AFR and remain worst-case optimal. 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.

V. 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 on, since the quality of the heuristics depends on the parameters of the simulation. We feel that with the field generally becoming more mature, the "average-case" heuristics will make way for the "worst-case" algorithms.

REFERENCES

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. In *MobiHoc*, pages 157–164, EPFL Lausanne, Switzerland, 2002.

[2] B. Awerbuch and D. Peleg. Concurrent online tracking of mobile users. In *SIGCOMM*, pages 221–233, 1991.

[3] J. Beutel. Geolocation in a Picoradio Environment. *Master Thesis, ETH Zurich and UC Berkeley*, 1999.

[4] J. Beutel, O. Kasten, and M. Ringwald. BTnodes - A Distributed Platform for Sensor Nodes. In *Proc. of ACM SenSys*, 2003.

[5] R. Bischoff and R. Wattenhofer. Analyzing Connectivity-Based Multi-Hop Ad-Hoc Positioning. In *Proc. of the Second Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2004.

[6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in ad hoc Wireless Networks. In *Proc. of the* 3^{rd} *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 48–55, 1999.

[7] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Topology control vs. interference. Technical report, Computer Science Department, ETH Zurich, Switzerland, 2003.

[8] C. Busch, S. Surapaneni, and 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.

[9] D.M.Blough, M.Leoncini, G.Resta, and P.Santi. The k-Neigh Protocol for Symmetric Topology Control in Ad Hoc Networks. In *ACM MobiHoc*, 2003.

[10] U. Feige. A Threshold of ln n for Approximating Set Cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[11] E. M. Gafni and D. P. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Transactions on Communication*, 29, 1981.

[12] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete Mobile Centers. In *Proc.* 17^{th} *Annual Symposium on Computational Geometry (SCG)*, pages 188–196. ACM Press, 2001.

[13] A. Goldberg, S. Plotkin, and G. Shannon. Parallel symmetry-breaking in sparse graphs. In *STOC*, pages 315–324, 1987.

[14] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K.S.J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[15] L. Jia, R. Rajaraman, and C. Scheideler. On Local Algorithms for Topology Control and Routing in Ad Hoc Networks. In *Proc. of the* 15^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2003.

[16] L. Jia, R. Rajaraman, and R. Suel. An Efficient Distributed Algorithm for Constructing Small Dominating Sets. In *Proc. of the 20th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 33–42, 2001.

[17] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Infocom*, 2003.

[18] E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. In *Proc.* 11^{th} *Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.

[19] F. Kuhn, T. Moscibroda, and R. Wattenhofer. How the Hidden Terminal Problem affects Initialization in Ad-Hoc Networks. Technical report, Department of Computer Science, ETH Zurich, 2003.

[20] F. Kuhn and R. Wattenhofer. Constant-Time Distributed Dominating Set Approximation. In *Proc. of the* 22^{nd} *ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[21] F. Kuhn and R. Wattenhofer. Distributed Combinatorial Optimization. Technical Report 426, Department of Computer Science, ETH Zurich, 2003.

[22] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Routing: Of Theory and Practice. In *Proc. of the* 22^{nd} *ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[23] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. Dial-M'02*, Atlanta, Georgia, USA, September 2002.

[24] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *MobiHoc*, 2003.

[25] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grunewald. Energy, congestion and dilation in radio networks. In *Proc. of the* 14^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2002.

[26] J.C. Navas and T. Imielinski. GeoCast - Geographic Addressing and Routing. In *MobiCom*, pages 66–76, 1997.

[27] R. Prakash. Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks. In *Proc. of the* 3^{rd} *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999.

[28] A. Rao, C. Papadimitriou, S. Ratnasamy, S. Shenker, and I. Stoica. Geographic Routing without Location Information. In *MobiCom*, 2003.

[29] Y. Wang and X.-Y. Li. Localized Construction of Bounded Degree Planar Spanner. In *Proc. of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, 2003.

[30] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *INFOCOM*, 2001.

[31] R. Wattenhofer and A. Zollinger. XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks. Technical Report 407, Department of Computer Science, ETH Zurich, 2003.

[32] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. In *IEEE LCN*, 2001.